Learning and Interpreting Deep Representations from Multi-Modal Data

Mandela Patrick

University College

University of Oxford

A thesis submitted for the degree of Doctor of Philosophy

Hilary 2021

Acknowledgements

To my co-supervisors, Andrea Vedaldi and João Henriques, thank you for your constant support, patience and guidance throughout this DPhil. Thank you for always being very generous with your knowledge and time, and giving me the space to tackle research problems that I am excited and passionate about. This thesis would not be possible without you two.

To my amazing collaborators, Yuki Asano, Ruth Fong, and Pa-Yao Huang, you have made each and every one of my research projects enjoyable, and intellectually enriching. I have learnt so much from each of you, and I would not be the researcher I am today if it was not for the countless hours we spent ideating, coding and brainstorming. Collaborators like you make a DPhil worth it.

More broadly, to my VGG family, I am so thankful to you for welcoming me as part of the lab, and always being a constant source of support. Andrew Brown, Lili Momeni, Daffy Afouras, Max Bain, Tendga Han, Christian Rupprecht, and Dylan Campbell, I am gonna miss you all and cannot wait to see how you all impact this field in fundamental ways.

This DPhil would not be possible without the financial support of the Rhodes Trust. Winning the Rhodes Scholarship has been one of my biggest blessings, and being part of the Rhodes community has made my Oxford experience really special.

I also want to thank EPSRC AIMS CDT for funding my research, and in particular, Wendy Poole for taking care of every administrative task related to my DPhil, thus allowing me to focus on my research. Wendy, you are simply the best.

Thank you to Facebook for supporting my DPhil research. I would not have been able to push the limits of large-scale multi-modal self-supervision without access to the data, computing resources and amazing collaborators, Ishan Misra, Geoffrey Zweig, Florian Metze, Christoph Feichtenhofer, Polina Kuznetsova, Rose Kanjirathinkal and Dong Guo.

To my friends in Oxford and London, Stephanie Ifayemi, Madeleine Chang, Samuel Liu, Alexander Thomas, Shaan Desai, Terrens Muradzikwa, Jelani Munroe, and Michael Chen, thank you for always supporting and pushing me throughout this journey.

You do not know how much it means to me to have you all in my life, and I have cherished every moment with you all.

Lastly, and most importantly, thank you to my family (Raymond, Hyacinth, and Nku) for always believing in me. You have been there for me during the highs and lows of this DPhil, and I can always count on you all to pick me up when I am down. This DPhil is for you.

Abstract

Deep learning has resulted in ground-breaking progress in a variety of domains, from core machine learning tasks such as image, language, and video understanding, to real-world industries such as medicine, autonomous driving, and agriculture. Its success has been driven by providing neural networks with manual supervision from large-scale labelled datasets such as ImageNet to automatically learn hierarchical data representations. However, obtaining large-scale labelled data is often a very time-consuming and expensive process. To address this challenge, we push the limits of self-supervision from multi-modal video data. Video data usually contain multiple modalities such as images, audio, transcribed speech and textual captions freely available. These modalities often share redundant semantic information and therefore can serve as pseudo-labels to supervise each other for representation learning without necessitating the use of manual human labels. Without the reliance on labelled data, we are able to train these deep representations on very large-scale video data of millions of video clips collected from the Internet. We show the scalability benefits of multi-modal self supervision by establishing a new state-of-the-art performance in a variety of domains: video action recognition, text-to-video retrieval, text-to-image retrieval and audio classification. We also introduce other technical innovations in terms of data transformations, model architecture and loss functions to further improve learning these deep video representations using multi-modal self-supervision. A secondary contribution of this thesis is new tools to improve the interpretability of deep representations, given that it is notoriously difficult to decipher the key features encoded in these deep representations. For images, we show how perturbation analysis can be used to analyze the intermediate representations of a network. For videos, we propose a novel clustering method using the Sinkhorn-Knopp algorithm to map deep video representations to human interpretable semantic pseudo-labels. The contributions in this thesis are steps to unlocking both the scalability and interpretability of deep video representation learning.

This thesis is submitted to the Department of Engineering Science, University of Oxford, in fulfillment of the requirements for the degree of Doctor of Philosophy. This thesis is entirely my own work, and except where otherwise stated, describes my own research.

Mandela Patrick, June 2021.

Contents

1 Introduction

1.1 Background

Machine learning is the subfield of artificial intelligence that seeks to develop algorithms that can automatically learn from data to improve its predictive and decision-making ability on a given task. The quality of decisions depends heavily on the space of inputs, or the data *representation*. This data representation ideally provides the model with the key attributes (or features) in the data to solve the given task. For example, classifying an animal into a semantic class such as a cat, is much easier given key attributes such as the shape of its head, or its number of legs, compared to given only the numerical values of the colors in an RGB image. These key attributes are usually not easy to automatically extract from the data, and therefore obtaining a powerful representation of the input data is one of the key components of any machine learning system.

Traditionally, these data representations have been hand-crafted. In other words, humans determine which features are salient for the task and develop algorithms to extract these manually defined properties from the input data. For example, SIFT [Lowe, 1999] and HOG [Dalal and Triggs, 2005] represent an image as a histogram of image gradients, in the hope that it will make it easier for the model to detect object edges and

shapes. While human-interpretable, these handcrafted features are domain-specific and require domain expertise to determine which features may be salient for a given task. Furthermore, these hand-crafted features are usually not very performant, with SIFT features only able to attain a top-5 accuracy of 73.8% [Lin et al., 2011] on the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Russakovsky et al., 2015]. These challenges with hand-crafted features, along with the breakthrough success of the AlexNet [Krizhevsky et al., 2012] neural network on the ILSVRC 2012 Challenge, reignited interest in *deep learning* [LeCun et al., 2015], where the goal is to *automatically* learn hierarchical representations of input data for automatic recognition of patterns from data such as images, videos, audio, and text using neural networks. Neural networks and hierarchical features have been a goal since at least the 1960s [Rosenblatt, 1958], but they weren't very effective [Minsky and Papert, 1969] because of a lack of computation, stochastic training, and most importantly, large-scale labelled data.

The recent success of deep learning has been driven by the use of neural networks [LeCun et al., 1998; Hochreiter and Schmidhuber, 1997], and large-scale labelled datasets such as ImageNet [Deng et al., 2009] and Kinetics [Kay et al., 2017]. With an appropriate loss function, such as the cross-entropy loss, which penalizes mis-classifications given the correct labels, these neural networks use the dataset labels to learn hierarchical and transferable representations [Zeiler and Fergus, 2014; Yosinski et al., 2014] of the semantic concepts present in the training data. For example, for an image of a cat paired with the label "cat", unlike hand-crafted SIFT representations, where the features are pre-defined using human intuition, a neural network learns to automatically detect the relevant features to identify the cat in this image.

Learning such hierarchical features is called *representation learning*. Representation learning performance has steadily increased over the years largely due to neural architectural advances. Since the breakthrough of AlexNet [Krizhevsky et al., 2012] on ILSVRC 2012 challenge, bigger and more powerful neural networks have been developed, such as VGG [Simonyan and Zisserman, 2015], GoogLeNet [Szegedy et al., 2015] and ResNet [He et al., 2016], leading to stronger and more transferable representations.

These neural architecture advances have lead to top-5 classification accuracy on the ILSVRC challenge increasing from 84.7% to 95.5%.

While it is possible to scale model sizes with advances in graphical processing units (GPUs), scaling the labelling of large-scale datasets such as ImageNet is not as trivial. Manual data collection is often expensive and time-consuming [Lin et al., 2014], which makes it extremely difficult to label datasets of billions of data points to train these data-hungry networks. Furthermore, labels are very sparse learning signals and can lead to shortcut learning [Geirhos et al., 2020], such as the model using background color instead of object shape to recognise the primary object in image representation learning. This problem is exacerbated for video data [Li et al., 2018], which is even more high-dimensional. Therefore, this motivates our primary research question: can we learn deep representations without explicit manual supervision from human labels?

Furthermore, as a secondary contribution, we are interested in whether we can map these deep representations into a format that is human understandable. Unlike hand-crafted representations, where the key attributes (features) are pre-defined by humans and are therefore interpretable, deep representations are usually not due to the many layers of non-linear data transformations within a neural network. This inspires an additional research question: can we better interpret deep representations?

1.2 Motivation

Learning Representations with Multi-modal Self-Supervision. While deep learning has enabled ground-breaking progress in a variety of domains [Ren et al., 2015; He et al., 2016; Tran et al., 2018], it has traditionally required large-scale labelled datasets [Deng et al., 2009; Kay et al., 2017] to learn representations. However, these labels are usually very expensive and time-consuming to obtain and limits the *scalability* of training these data-hungry models on very large-scale data.

We draw inspiration from image-based *self-supervised learning* that use *pretext tasks* to automatically generate differentiable learning signals from the data itself in order to

learn deep representations. By solving a pretext task such as rotation prediction [Gidaris et al., 2018], where the neural network has to detect the orientation of a rotated image, the hope is that the network learns a meaningful representation of the data. Due to photographs normally being taken upright, the neural network has to learn the difference between "upright objects" *vs.* rotated ones, and can only solve this task if it learns to recognize the objects in an image.

In the last few years, there has been rapid progress in image-based self-supervision, with recent methods [He et al., 2020; Misra and van der Maaten, 2020; Chen et al., 2020; Grill et al., 2020] even surpassing supervised pre-training when representations are transferred to downstream tasks such as object detection [Lin et al., 2014; Everingham et al., 2010], and image classification [Zhou et al., 2017].

Learning deep representations from images is however limiting. Images are a static representation of our world, and image-based self-supervised learning cannot learn representations that encode temporal and causal semantics [Gordon et al., 2020]. Furthermore, image-based self-supervision can only capture a *uni-modal* representation of the world. In applying deep learning in the real world, unimodal data can often be corrupted, noisy, or missing, therefore making it difficult for the model to extract semantic meaning from the input data.

Unlike images, videos offers more dimensions for pre-text tasks such as time [Misra et al., 2016; Wei et al., 2018; Xu et al., 2019] and multiple modalities, such as audio [Arandjelovic and Zisserman, 2017], optical flow [Han et al., 2020] and text [Miech et al., 2020]. Despite recent progress in self-supervised approaches for the video domain [Misra et al., 2016; Wei et al., 2018; Wang et al., 2019a; Kim et al., 2019; Jing and Tian, 2018; Han et al., 2019], their performance are still lagging behind supervised pretraining of representations on large-scale video datasets such as Kinetics [Kay et al., 2017]. We hypothesize that these visual-only approaches are limiting, and that using self-supervision from multiple video modalities, such as text and audio, can result in better deep video representations. First presented in the landmark paper by de Sa

[1994], *multi-modal self-supervision* exploits the co-occurrence of multiple-modalities in natural environments to learn representations.

Why use multiple modalities? With multiple modalities, the sensory inputs are very different but causally linked (e.g. sound and image produced by the same object). Therefore, learning the signal in common enables us to get to the cause of this data, which is an object or event, something with semantic meaning. While within a single modality, the signal in common between, for example, two adjacent patches of an image may be a depicted object, it may also simply be interpreted as a texture in common between the two [Asano et al., 2020a]. Networks can therefore typically solve pretext tasks in single modalities using more low-level features (simple patterns), while multi-modality requires higher-level features (semantic concepts).

This intuition is also grounded in how we as humans learn. Humans learn a great deal from associations between senses: for example, early in development, seeing a face and hearing a voice teaches us about other people's presence and identities [Smith and Gasser, 2005]. The power of learning from multi-modal signals is due to *redundancy*, or the overlapping information between modalities, also referred to as degeneracy [Edelman, 1987] in the psychology literature. This phenomenon explains why we can experience the world even with the loss of sight, because our world experience is present in sound, movement, touch, and even smell.

Given most videos on the Internet have an accompanying sound track or caption, learning video representations from multiple modalities is a scalable and practical solution for self-supervision and we hypothesise that training these neural networks using extremely large scale datasets is key to unlocking the benefits of multi-modal learning.

Interpreting Deep Representations. This thesis also attempts to make a contribution to solve another key bottleneck of deep representations: their lack of interpretability. Unlike hand-crafted features, which are usually human-interpretable, deep representations are often very difficult to interpret and visualize [Samek et al., 2019]. Enabling the interpretability of deep representations is important for a variety of reasons.

1. Introduction

Understanding the key features encoded in a neural network's representation is important for *verification* of the model's decision. Neural networks are black box systems, and therefore it is usually difficult to determine and trust the features used to make its decision. In certain industries, such as the healthcare or legal system, it is absolutely imperative for experts to understand the key features used in a model's decision for these deep learning systems to be widely used and trusted. This will allow experts to discover any flaws in logic or systematic biases [Mehrabi et al., 2019] in the model.

Fundamentally, machine learning systems are being integrated and affecting more aspects of human life, and therefore, it is important to develop a legal framework around which to *regulate* their use. For example, in the near future, when autonomous vehicles are widely used, explainable models are needed for regulators to accurately assign responsibility when these systems make a mistake such as during an accident [Alves et al., 2018] . However, given current deep learning models, it is near impossible for such legal decisions to be made until we have developed the tools to sufficiently interpret the salient features encoded in these deep models. Such concerns have led to the European Union passing new regulations to implement "right to explain" for machine learning models, where users have the right to access the explanation of decisions affecting them [Goodman and Flaxman, 2017].

Improving the interpretability of deep representations will also enable humans to *learn* from machine learning systems. Due to their ability to learn from millions to billions of training examples, machine learning systems have achieved super-human ability in a variety of domains from board-games [Silver et al., 2018] to protein folding [Jumper et al., 2020]. If we can effectively distill the key features encoded in these models, there is an opportunity to use these models to teach humans new skills or discover fundamental science concepts.

With all these use-cases, understanding the key features encoded in deep representations is extremely important. These are the features encoded in the representation of the penultimate layer of a neural network, which is the input to the network's

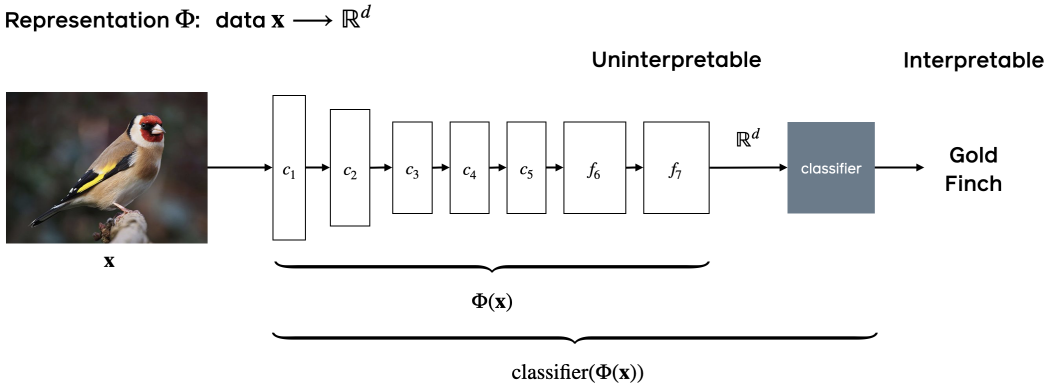


Figure 1.1: Deep Representations. A neural network, usually consists of two parts: a feature extractor (ϕ), which maps input, $\mathbf{x}$ to a data representation $\in R^d$, and a classification layer, which maps the data representation to a label, a human-interpretable output.

classification layer (Fig. 1.1). Understanding the salient features that are encoded in this representation is key to determining what information is used by the neural network for its decision. However, most interpretability research has instead focused on the attribution problem [Simonyan et al., 2014; Fong and Vedaldi, 2017; Selvaraju et al., 2017], which attempts to understand the salient *input* features for a model's decision. In this thesis, we therefore focus on developing tools to visualize and interpret the *hidden* representations of neural networks using perturbation analysis (chap. 7) and clustering (chap. 8).

1.3 Key Ideas

This thesis is primarily focused on (I) Learning Representations with Multi-Modal Self-Supervision; and additionally, (II) Interpreting Deep Representations.

In the sections below, we focus on the key ideas present in the thesis and how they relate to what has been done previously in the literature.

1.3.1 Learning Representations with Multi-Modal Self-Supervision

To learn a deep representation, there are three main components: (I) Data; (II) Model; (III) Learning objective.

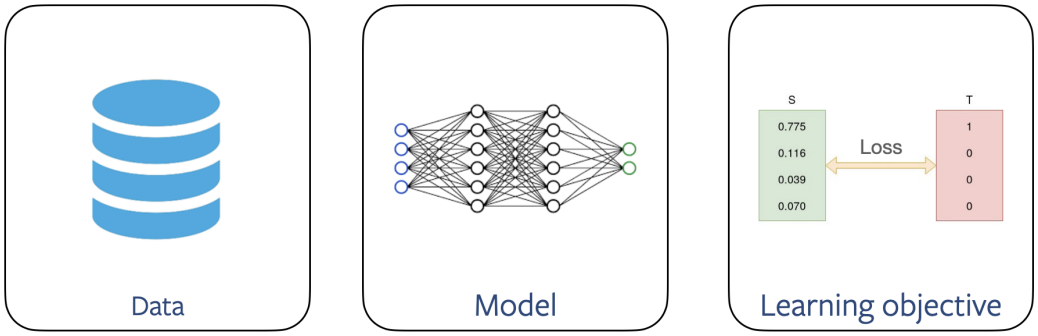


Figure 1.2: Key Ideas for Improvement in Multi-Modal Self-Supervision. Here, we illustrate the key ingredients of learning representations using multi-modal self-supervision.

We systematically explore and present innovations along each axis, significantly pushing the state-of-the-art.

Data. The two core ideas we explore are *data transformations* and *data scale*. Data transformations have been shown to be a key component of image-based self-supervised learning [Chen et al., 2020; Asano et al., 2020a]. Transformations such as random cropping, gaussian blur, rotations, and color jittering, are usually applied to the input data before being passed into the neural network. These data transformations serve as priors on the types of invariances and distinctiveness to encode in the learned representation.

This thesis explores the rich space of data transformations that are possible for video data and uses it for multi-modal self-supervised representation learning. One of the key data transformations we explore is *modality splicing* - which essentially extracts a modality from a video input. Unlike most previous works which splice the audio and visual modality [Arandjelovic and Zisserman, 2017; Korbar et al., 2018; Owens and Efros, 2018] from videos for representation learning, we also explore how other modalities such as automatic speech recognition outputs in English (chap. 4) and other languages (chap. 5) can be used as powerful data transformations for representation learning. We also explore the importance of *composing* data transformations (chap. 2) and leveraging *within-modality data* transformations such as input and feature crops (chap. 3) to improve representation learning performance.

Another important dimension we explore is the importance of dataset scale for improving representation learning performance. Early works use labelled video datasets such as Kinetics-400 [Kay et al., 2017] and Audioset [Gemmeke et al., 2017] to learn video representations. While sufficiently large scale, these datasets are limited by the fact that they were collected for labelling and therefore the size of these datasets are between 200,000 − 1,000,000 videos. In this thesis, we show the importance of leveraging even larger-scale datasets such as IG65M [Ghadiyaram et al., 2019], a collection of 65 million video clips collected from a social media website, and HT100M [Miech et al., 2019], an instructional video-text dataset of 130 million video clips collected from YouTube, for representation learning. With multi-modal self-supervision, we are no longer bottlenecked by having to collect semantic labels for these datasets and instead can use the free supervision present in the audio and text modality. We show that with sufficient data scale, we can push the limits of representation learning, even surpassing supervised pre-training on the standard Kinetics-400 [Kay et al., 2017] dataset (chap. 2).

Models. The model maps the input data into a data representation. In the case of multi-modal self-supervision, there is usually an encoder for each modality. Traditionally, video representations were obtained using 2D convolutional neural networks [LeCun et al., 1998] for a frame-based encoding, and an aggregation module such as average pooling or NetVLAD [Arandjelović et al., 2016] to aggregate these frame-based features over time. More recently, 3D convolutional neural networks [Tran et al., 2015, 2018; Feichtenhofer et al., 2019; Xie et al., 2018] were developed to use 3D convolutional filters to encode short-term temporal dynamics directly in video representations. Other approaches include two-stream networks [Simonyan and Zisserman, 2014; Feichtenhofer et al., 2016] which decompose spatial and temporal processing into two convolutional neural networks. All these architectures try to capture both short and long-term temporal information in video representations.

In this thesis, we explore how the transformer architecture can be beneficial for encoding temporal information in video representations (chap. 3 and 6). Transformers [Vaswani

et al., 2017] are generic neural architectures that use multi-head attention to aggregate and contextualize feature representations from a sequence, and has shown to to be extremely successful in the natural language domain [Jacob Devlin and Toutanova, 2018; Brown et al., 2020; Radford et al., 2019]. We demonstrate that the video encoder in a multi-modal learning setup can benefit from late temporal aggregation using a shallow transformer (chap. 3) rather than naive average temporal pooling as is common in most video architectures [Tran et al., 2018; Xie et al., 2018]. Furthermore, we show that a 3D-convolutional neural network can be fully replaced using a transformer architecture (chap. 6), building on the success of ViT [Dosovitskiy et al., 2021] in the image domain. We show that our proposed video transformer attention model can better model temporal dynamics and motion trajectories outperforming competing 3D convolutional architectures.

Loss Function. To encourage the representation to encode the right features, the choice of loss function is extremely important. The loss function drives the learning process and is a key component of the self-supervised learning pipeline.

Traditionally, the binary cross-entropy or triplet (max-margin) objectives [Chopra et al., 2005] were used in the multi-modal self-supervised learning setup. These losses encouraged the representation to discriminate visual and aural signals from the same video from others.

Recently, noise contrastive training [Hadsell et al., 2006; Gutmann and Hyvärinen, 2010] has gained popularity to learn self-supervised image representations [Wu et al., 2018; He et al., 2020; Chen et al., 2020]. In this thesis, we explore how noise contrastive training can be adapted in the multi-modal setting both for video-text (chap. 4 and 5) and video-audio (chap. 2 and 3) representation learning. We show that this loss is crucial for learning strong video representations across a variety of domains and tasks (chap. 2).

We also explore how other learning objectives can improve representations. Specifically, we show how a reconstruction objective can be used as an auxiliary loss to alleviate some of the learning pitfalls of the contrastive loss and improve video representations (chap. 4).

1.3.2 Interpreting Deep Representations

In this thesis, we focus on making a contribution in the following two areas in the interpretability literature: (I) Attribution; (II) Visualizations of intermediate activations.

Attribution. Interpretability research has primarily focused on the attribution problem i.e. example-level explanations that justifies a model's output. In computer vision, the paradigm has been to provide visual explanations of "where" a model is looking, typically which areas of the image activate each classifier, or each neuron; or recreating a synthetic view of what the "canonical" image looks like to activate one neuron. There are mainly two classes of techniques: backpropagation and perturbation analysis. Backpropagation techniques leverage the gradient [Simonyan et al., 2014], or some variant of it [Springenberg et al., 2014; Smilkov et al., 2017], to track the information from the network's output back to its input to determine which input features are most sensitive to the output class neuron. Backpropagation techniques can be further improved by combining the gradient with network weights or activations at certain layers [Zhou et al., 2016; Selvaraju et al., 2017; Zhang et al., 2018]. Perturbation analysis, on the other hand, visualize how perturbations of an input affect the output of a model [Zeiler and Fergus, 2014; Fong and Vedaldi, 2017; Zhou et al., 2015; Ribeiro et al., 2016]. In this thesis, we improve upon the shortcomings of perturbation analysis [Zeiler and Fergus, 2014; Fong and Vedaldi, 2017] by removing all tunable hyper-parameters from the optimization problem in a new approach called extremal perturbations (chap 7).

Visualizations of intermediate activations. Another area of interest in this thesis is to explain what happens at the feature or model level, such as what a single or combination of hidden unit encodes. Deconvolution [Zeiler and Fergus, 2014] project the feature representations back to the input pixel space to understand input stimuli that most activate feature units. Activation maximisation [Simonyan et al., 2014] learns an input image that maximally activates a given filter. Feature inversion [Mahendran and Vedaldi, 2015] learns an image that reconstructs a network's intermediate activations

while leveraging a natural image prior for visual clarity. Subsequent works tackled the problem of improving the natural image prior for feature inversion and/or activation maximization [Nguyen et al., 2016, 2017; Ulyanov et al., 2020; Olah et al., 2017; Zhou et al., 2018; Mordvintsev et al., 2018]. Recently, some methods have measured the performance of single [Zhou et al., 2019; Zhou et al., 2018] and combination [Kim et al., 2018; Fong and Vedaldi, 2018] of filter activations on probe tasks like classification and segmentation to learn an alignment between hidden units and visual semantic concepts. In this thesis, we show that we can extend our extremal perturbation framework to the intermediate layers of a network to allow us to visualize which channels (or concepts) are salient for a classification decision. Moreover, we also propose to use clustering techniques to map deep representations to a human-interpretable psuedo-label. The intuition is that one can interpret representations by looking not at a single image but collections of them. e.g. grouping them. While this has been done for images [Asano et al., 2020b; Caron et al., 2018], it is not trivial for data consisting of multiple modalities, and we show how to effectively cluster multi-modal video data (chap. 8).

1.4 Thesis Outline and Contributions

In this section, we summarize the contributions of this thesis, and provide an outline of the chapters. The thesis is divided into two parts – (I) Learning Representations, (II) Interpreting Representations.

For Chapters 2 to 8, we summarise the main contributions below. Finally, Chapter 9 discusses the impact of this work and avenues for future exploration.

Part I: Learning Representations Using Multi-Modal Self-Supervision

In chapter 2, we introduce Generalized Data Transformations (GDT), a framework for multi-modal self-supervision using noise contrastive training. We show the benefits of composing data transformations such as time shift, time reversal, modality splitting for learning strong video representations. We also show the benefits of using large-scale datasets for multi-modal self-supervision by pre-training these video representations on

two large scale datasets such as IG65M [Ghadiyaram et al., 2019] and HT100M [Miech et al., 2019]. We set state-of-the-art performance on video action recognition and audio classification, when these pretrained video representations are finetuned on smaller-scale datasets such as HMDB-51 [Kuehne et al., 2011], UCF-101 [Soomro et al., 2012], ESC-50 [Piczak, 2015] and DCASE-2014 [Stowell et al., 2015].

In chapter 3, we propose two key improvements for multi-modal self-supervision, namely, incorporating within-modal invariance learning and using transformers for late temporal aggregation. Most multi-modal representation learning works [Arandjelovic and Zisserman, 2017; Korbar et al., 2018; Owens and Efros, 2018] only compare representations from different modalities, with the motivation, that this is a good prior for learning semantic representations. We show that cross-modal learning can be further improved by comparing representations of differently cropped versions of the video input, thereby encoding within-modal invariance. We show that this can be done efficiently by performing the cropping in feature space rather than input space. We also propose the use of a shallow transformer aggregation layer instead of temporal average pooling as an orthogonal, but additive, contribution to improve video representation learning.

In chapter 4, we perform multi-modal self- supervision using video and text as modalities. Similarly to our work with GDT, we show the scalability of having not to rely on labels by training on the large-scale HT100M video dataset [Miech et al., 2019], consisting of over 120 million video clips. We also attempt to alleviate one of the pitfalls of noise contrastive learning – false negatives – by adding a reconstruction loss to the learning objective. By forcing the model to reconstruct the caption from a video representation that is weighted combination of different video embeddings from a support-set, we hope it forces the model to discover semantic relationships between different videos.

In chapter 5, we demonstrate the benefits of using captions in multiple languages as another "modality" for multi-modal self-supervision. We introduce a new dataset, Multi-HT100M, which is a multilingual version of the large-scale HT100M dataset in 9 languages. With large-scale pretraining on this dataset using a multilingual text encoder and video encoder, we are able to perform zero-shot text-to-video retrieval in

new languages and even improve English-to-video retrieval state-of-the-art on common benchmarks such as MSR-VTT [Xu et al., 2016] and VATEX [Wang et al., 2019b]. With a small amount of finetuning, our model can generalize to a new domain, text-to-image search, where we also improve upon the state-of-the-art.

In chapter 6, we propose Motionformer, a transformer architecture for learning video representations that aggregates features along implicitly determined motion paths. We show that this motion inductive bias is key for injecting temporal information in these transformer models and allow us to set state-of-the-art performance on various video action recognition datasets.

Part II: Interpreting Representations

In chapter 7, we propose a principled approach to the attribution problem using extremal perturbations. Given a network's output decision, extremal perturbations allows a human user to identify which parts of the input image are most salient for the network's outputs. Furthermore, we show how the extremal perturbation framework can be extended to intermediate deep representations of the neural network, allowing a human user to visualize the key features encoded in the representation that were important for model's output. While attribution at the input layer is easy to visualize using a heatmap, at an intermediate layer, it's more difficult. We show how combining feature inversion with our extremal perturbation framework, one can visualize the combination of channels (features) most salient for model's output decision.

In chapter 8, we explore interpretability of deep representations using clustering. A representation is a mapping from data to a vector in R^d and by definition is naturally un-interpretable to a human. We therefore propose to map representations to a high-level pseudo-label that a human can understand. To achieve this, we develop a technique that can automatically discover clusters from multi-modal representations via Sinkhorn-knopp [Cuturi, 2013], an efficient optimal transport algorithm.

1.4.1 Publications

Chapters 2 to 8 each contain a paper that has been peer-reviewed and accepted for publication in a conference, or is currently under review. The publications included in this thesis are:

Chapter 2: On Composition of Transformations for Self-Supervised Learning

Mandela Patrick*, Yuki M. Asano*, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, Andrea Vedaldi. In *International Conference of Computer Vision 2021.*

Chapter 3: Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning

Mandela Patrick*, Yuki M Asano*, Bernie Huang*, Ishan Misra, Florian Metze, João F. Henriques, Andrea Vedaldi. In *International Conference of Computer Vision 2021.*

Chapter 4: Support-set bottlenecks for video-text representation learning

Mandela Patrick*, Po-Yao Huang*, Yuki Asano*, Florian Metze, Alexander Hauptmann, João F. Henriques, Andrea Vedaldi. In *International Conference of Learning Representations 2021.*

Chapter 5: Multilingual Multimodal Pre-training for Zero-Shot Cross-Lingual Transfer of Vision-Language Models

Po-Yao Huang*, **Mandela Patrick***, Junjie Hu, Graham Neubig, Florian Metze, Alexander Hauptmann. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies 2021.*

Chapter 6: Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers

Mandela Patrick*, Dylan Campbell*, Yuki M Asano*, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, João F. Henriques. Under review.

Chapter 7: Understanding deep networks via extremal perturbations and smooth masks

1. Introduction

Ruth Fong*, **Mandela Patrick***, Andrea Vedaldi. In *International Conference of Computer Vision 2019*.

Chapter 8: Labelling unlabelled videos from scratch with multi-modal self-supervision

Yuki M Asano*, **Mandela Patrick***, Christian Rupprecht, Andrea Vedaldi. In *Advances in Neural Information Processing Systems 2020*.

Part I

Learning Representations

2

On Composition of Transformations for Self-Supervised Learning

This work will be presented at the International Conference on Computer Vision (ICCV) 2021.

On Compositions of Transformations in Contrastive Self-Supervised Learning

Mandela Patrick[1,2*] Yuki M. Asano[2*] Polina Kuznetsova[1] Ruth Fong[2]
João F. Henriques[2] Geoffrey Zweig[1] Andrea Vedaldi[1,2]
[1] Facebook AI Research
mandelapatrick@facebook.com
[2] Visual Geometry Group, University of Oxford
yuki@robots.ox.ac.uk

Abstract

In the image domain, excellent representations can be learned by inducing invariance to content-preserving transformations via noise contrastive learning. In this paper, we generalize contrastive learning to a wider set of transformations, and their compositions, for which either invariance or distinctiveness is sought. We show that it is not immediately obvious how existing methods such as SimCLR can be extended to do so. Instead, we introduce a number of formal requirements that all contrastive formulations must satisfy, and propose a practical construction which satisfies these requirements. In order to maximise the reach of this analysis, we express all components of noise contrastive formulations as the choice of certain generalized transformations of the data (GDTs), including data sampling. We then consider videos as an example of data in which a large variety of transformations are applicable, accounting for the extra modalities – for which we analyze audio and text – and the dimension of time. We find that being invariant to certain transformations and distinctive to others is critical to learning effective video representations, improving the state-of-the-art for multiple benchmarks by a large margin, and even surpassing supervised pretraining. Code and pretrained models are available[1].

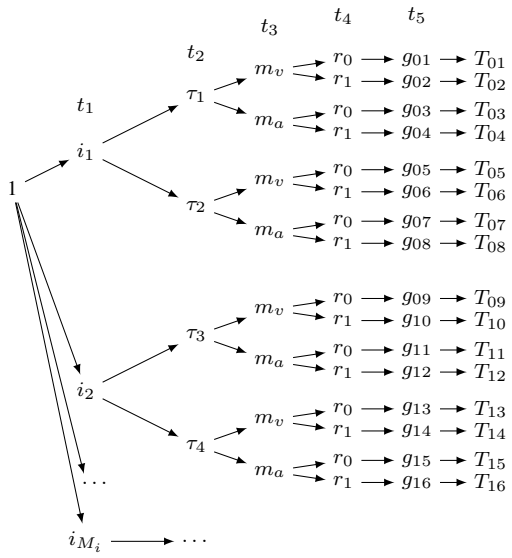


Fig. 1: Hierarchical sampling process of generalized data transformations (GDTs). Shown here are the five transformations analyzed for the audio-visual training case and their compositions: data-sampling (t_1), time-shift (t_2), modality splicing (t_3), time-reversal (t_4), and augmentation transformations, g (t_5) to learn video representations via noise contrastive learning.

1. Introduction

Works such as MoCo [31], SimCLR [13], SwAV [12] and BYOL [25] have shown that it is possible to pre-train state-of-the-art image representations without the use of any manually-provided labels. Furthermore, many of these approaches use variants of noise contrastive learning [26, 27]. Their idea is to learn a representation that is *invariant* to transformations that leave the meaning of an image unchanged (e.g. geometric distortion or cropping) and *distinctive* to changes that are likely to alter its meaning (e.g. replacing an image with another chosen at random).

These prior works have also shown that the choice of transformations is of primary importance for performance [12, 13]. This is not just a matter of selecting a

*Joint first authors
[1]https://github.com/facebookresearch/GDT

certain type of transformation, but also to specify how different transformations should be composed, and how these compositions should be sampled to from batches for training the model. So far, these choices have been mostly driven by intuition, with little formal understanding of why certain choices may be preferable, and how these choices can be generalized.

In this work, we answer some of these questions via a formal analysis of *composable transformations in contrastive learning*. Our analysis shows how invariance and distinctiveness to individual transformations can be meaningfully incorporated in the same learning formulation. It also provides some principles to guide the construction of the training batches. We interpret existing sampling schemes, such as the one in SimCLR, as special cases with certain potential advantages and disadvantages. We do so by showing how these constructions can be extended systematically to any composition of invariant and distinctive transformations.

Furthermore, we demonstrate the utility of our analysis by exploring contrastive methods for learning representations of video data. Compared to images, videos contain a time dimension and multiple modalities, which have been shown to provide effective learning cues; for instance [60] leverages multiple modalities, and [15, 41] the time dimension. We show how these effects can be incorporated in a uniform manner in contrastive learning by considering a suitable class of *generalized data transformations* (GDTs). GDTs capture standard augmentations, as well as temporal transformations, modality slicing and data sampling. The advantages of using GDTs is that they allow us to base the entire design of the learning formulation (e.g., how to write a coherent learning objective and how to sample batches) on a small number of design principles that our analysis has identified.

With this, we make some notable findings for contrastive video representation learning. First, we show that using this wider class of transformations greatly exceeds the performance that can be obtained by a vanilla applications of image-centric methods such as SimCLR to video data. By leveraging time and multiple modalities, we obtain large performance gains, almost *doubling* the performance. Second, we show that just learning representations that are invariant to more and more transformations is *not* optimal, at least when it comes to video data; instead, combining invariance to certain factors with distinctiveness to others performs better. To the best of our knowledge, this is the first time such an effect has been demonstrated in contrastive learning.

We also set the new state of the art in audio-visual representation learning, with both small and large video pretraining datasets on a variety of downstream tasks. In particular, we achieve 94.1% and 67.4% on the standardized UCF-101 [67] and HMDB-51 [42] action recognition benchmarks, when pretrained on HowTo100M [50], and 95.2% and 72.8% respectively when pretrained on IG65M [21].

2. Related work

Self-supervised learning from images and videos. A variety of pretext tasks have been proposed to learn representations from unlabelled **images**. Some tasks leverage the spatial context in images [17, 56] to train CNNs, while others create pseudo classification labels via artificial rotations [23], or clustering features [6, 10, 11, 12, 22, 37]. Colorization [83, 84], inpainting [62], solving jigsaw puzzles [57], as well as the contrastive methods detailed below, have been proposed for self-supervised image representation learning. Some of the tasks that use the space dimension of images have been extended to the space-time dimensions of **videos** by crafting equivalent tasks. These include jigsaw puzzles [40], and predicting rotations [38] or future frames [28]. Other tasks leverage the temporal dimension of videos to learn representations by predicting shuffled frames [53], the direction of time [76], motion [74], temporal ordering [43, 80], and playback speed [9, 14, 19]. These pretext-tasks can be framed as GDTs.

Multi-modal learning. Videos, unlike images, are a rich source of a variety of modalities such as speech, audio, and optical flow, and their correlation can be used as a supervisory signal. This idea has been present as early as 1994 [16]. Only recently, however, has multi-modal learning been used to successfully learn effective representations by leveraging the natural correspondence [2, 4, 5, 7, 54, 60] and synchronization [15, 41, 59] between the audio and visual streams. A number of recent papers have leveraged speech as a weak supervisory signal to train video representations [46, 49, 55, 68, 69] and recently [1], who use speech, audio and video. Other works incorporate optical flow and other modalities [29, 30, 64, 85] to learn representations. In CMC [70], representations are learned with different views such as different color channels or modalities to solely induce multi-view invariance. In contrast, our work extends this to and analyses multi-modal transformations and examines their utility as an invariant *or* distinctive learning signal.

Noise Contrastive Loss. Noise contrastive losses [26, 27] measure the similarity between sample pairs in a representational space and are at the core of several recent works on unsupervised feature learning. They yield good performance for learning image [13, 31, 33, 35, 45, 52, 58, 70, 71, 77] and video [3, 28, 34, 46, 49, 54, 66, 68, 82] representations, and circumvent the need to explicitly specify what information needs to be discarded via a designed task.

We leverage the noise contrastive loss as a learning framework to encourage the network to learn desired invariance and distinctiveness to data transformations. The GDT framework can be used to combine and extend many of these cues, contrastive or not, in a single noise contrastive formulation.

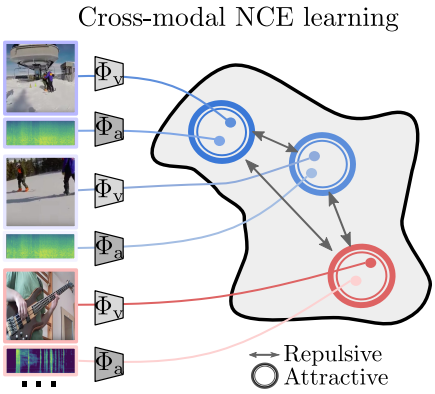


Fig. 2: **Example instantiation.** The embedding is learned via noise contrastive (NCE) learning. Here we show the case of audio-visual sample and time-shift distinctiveness: video-audio embeddings from the same video at the same time are pulled together, while audio-visual sample pairs from different videos and different starting times are pushed apart.

3. Method

We generalize contrastive methods such as CPC [58], PIRL [52], MoCo [31], SimCLR [13], and SwAV [12] to learn representations that can be invariant or distinctive to any number of transformations.

Given a collection x of data such as images or videos, we generate training samples

$$x(t_1, \dots, t_M) \in \mathcal{X}.$$

by applying a sequence of M transformations $T = (t_1, \dots, t_M)$ to the collection. We consider typical transformations such as data augmentations (e.g., randomly cropping an image). We also find it useful to express in the same manner other operations such as extracting a specific image or video from the collection or extracting a specific modality from a video. We call these *generalized data transformations* (GDTs).

To provide a concrete example, in a standard contrastive learning formulation such as SimCLR, the first transformation $t_1 = i \in \{1, \dots, |x|\}$ extracts an image x_i from the collection x and the second transformation $t_2 = g$ applies to it a random augmentation, so that we can write $x(t_1, t_2) = g(x_i)$. The goal is to learn a representation $\Phi : \mathcal{X} \to \mathbb{R}^d$ that identifies the image regardless of the augmentation; in other words, Φ should be invariant to the choice of t_2 and distinctive for the choice of t_1.

We wish to generalize this construction to richer data such as videos. Compared to images, videos contain multiple modalities and additional dimensions, which allows to consider qualitatively different transformations such as time shift, time reversal, and modality slicing. This generalization is however non-trivial. First, when considering $M > 2$ transformations, we have a choice of making the representation invariant or distinctive to each of them independently. For instance, video representations may benefit from being distinctive to time shift and/or time reversal rather than invariant to them. It is not immediately obvious how contrastive learning should be modified to incorporate these different choices. Another less apparent but important issue is how training data batches should be formed. Contrastive learning formulations minimize, in fact, a loss that involves comparing (contrasting) the representations of different samples, and is thus *not decomposable*. In practice, the loss is approximated by sampling batches of data, and how this is done has a major effect on the performance. In the previous example of SimCLR, if transformation (t_1, t_2) is included in the batch, so must be a complementary transformation (t_1, t_2') that differs only in the second factor $t_2 \neq t_2'$. This is required in order to learn the desired invariance. It also means that transformations in a batch cannot be sampled independently. A way to guarantee that both (t_1, t_2) and (t_1, t_2') are in the batch is to consider all possible combinations $\mathcal{T}_1 \times \mathcal{T}_2$ of two sets of transformations $\mathcal{T}_1$ and $\mathcal{T}_2$. However this is statistically inefficient because it applies the same augmentations $\mathcal{T}_2$ to all images in the batch. Instead, SimCLR samples at random $B/2$ images and then applies to them B independently sampled augmentations. This is better than the scheme above that would only use $B/|\mathcal{T}_1| = 2$ different augmentations. However, it is unclear how this strategy for sampling diverse transformations can be extended to $M > 2$ factors. This is studied next.

3.1. Compositional contrastive learning

Given a batch $\mathcal{T}$ of data transformations, we consider the learning objective:

$$\mathcal{L}(\Phi; \mathcal{T}) = -\sum_{T, T' \in \mathcal{T}} c(T, T') w(T, T') \cdot \log \left(\frac{e^{\langle \Phi(x(T)), \Phi(x(T')) \rangle / \rho}}{\sum_{T'' \in \mathcal{T}} w(T, T'')\, e^{\langle \Phi(x(T)), \Phi(x(T'')) \rangle / \rho}} \right). \quad (1)$$

where $\rho > 0$ is a temperature parameter. The *contrast function* $c(T, T') \in \{0, 1\}$ has the following interpretation: when $c(T, T') = 1$, then the representations $\Phi(x(T))$ and $\Phi(x(T'))$ are pulled together (invariance), and when $c(T, T') = 0$ they are pushed apart (distinctiveness). For example, in SimCLR, we set $c(T, T') = c((i, g), (i', g')) = \delta_{i=i'}$ to push apart the representations of different images (i, i') while remaining invariant to a transformation pair (g, g'). The *weight function* w is a second binary function that focuses learning on more informative transformation pairs; for instance, SimCLR sets $w(T, T') = \delta_{T \neq T'}$ to

avoid focusing learning invariance to identical transformation $T = T'$ as this is trivially satisfied. Next, we provide a semi-formal analysis of this formulation, leaving the details to Appendix A.1.

Multiple invariances and distinctiveness. The key to extending eq. (1) to $M > 2$ transformation is to build the function $c(T, T')$. We do this one factor a time. If we wish the representation to be distinctive to factor t_m, we set $c(t_m, t'_m) = \delta_{t_m = t'_m}$. If we wish it to be invariant to it, we set $c(t_m, t'_m) = 1$. In lemmas 4 and 5 (Appendix A.1), we show that, given these choices, the only consistent definition of $c(T, T')$ is the product $\prod_{m=1}^{M} c(t_m, t'_m)$. The intuition is as follows: The representation Φ should distinguish samples $x(T)$ and $x(T')$ if, and only if, at least one of the distinctive factors in T and T' differs.

Forming a batch. Given c, the remaining challenge is to sample training batches $\mathcal{T}$ appropriately. We start by deriving some requirements for $\mathcal{T}$ and then develop a sampling scheme that satisfies them (none of these are guaranteed by sampling T and T' independently). (i) First, in order for eq. (1) not to be identically zero, $c(T, T')$ should be non-zero for at least *some* choices of T and T' in the batch. (ii) Furthermore, when $c(T, T') = 1$, this should not be for the trivial case $T = T'$ (the one that SimCLR discounts by setting $w(T, T') = 0$). Based on the discussion above, the condition $c(T, T') = 1 \wedge T \neq T'$ means that all distinctive factors in T and T' agree and that at least an invariant factor differs. (iii) Additionally, for the fraction in eq. (1) not to be a constant, if $c(T, T') = 1$, there should be another T'' in the batch such that $c(T, T'') = 0$. The latter means that at least one distinctive factor in T and T'' differs.

Short of considering all possible combinations of transformations (which, as explained above, can be statistically inefficient), we can sample a batch $\mathcal{T}$ that satisfies these constraints as follows. We describe this process for the case of $M = 3$ transformations, but note that it extends immediately to any M (this is done in Appendix A.1.1 and A.1.2). First, we sample K_1 versions of the first distinctive transformations t_1. Then, for each t_1, we sample K_2 transformations t_2, also distinctive. Finally, for each choice of (t_1, t_2), we sample K_3 invariant transformations t_3.[2] We thus obtain a batch of $|\mathcal{T}| = K_1 K_2 K_3$ transformations.

This scheme has several desirable properties. First, for every $T = (t_1, t_2, t_3)$, there is another $T' = (t_1, t_2, t'_3)$ that agrees on the distinctive factors and differs in the invariant one (properties (i) and (ii)). Second, there is a $T'' = (t_1, t'_2, t'_3)$ or $T'' = (t'_1, t'_2, t'_3)$ that differs in one or more distinctive factors (property (iii)). Third, the construction is balanced, in the sense that the number of transformations that share a particular factor value t_m is the same for all values of t_m (this number is $|\mathcal{T}|/(K_1 \cdots K_m)$). Furthermore, SimCLR is obtained as a special case. Please see lemmas 6 and 7 (Appendix A.1) for an in-depth discussion.

Limitations. Despite the benefits, this scheme has also some limitations. The main issue is that a difference in factor t_m generally implies a difference in all subsequent factors as well, meaning that the representation may be unable to observe and thus learn to discriminate changes in all individual factors. In Appendix A.1.3, we show why this is unlikely to be an issue for the practical cases considered here and in the literature. However, we also suggest other practical cases where this *can* be a significant issue, affecting even methods such as SimCLR.

3.2. Properties of Generalized Data Transformations

In this section, we show that GDT's batch sampling strategy is statistically more efficient than naively-sampled pairs for contrastive learning. We do this by showing that GDT's objective has the same mean but a lower variance than sampling batches with eq. 1 directly, which would either enumerate all possible pairs of transformations (which is prohibitively expensive) or subsample it by sampling transformations independently. We assume that the distinctive transformations are injective. This must be approximately true, otherwise it is impossible for any method to be distinctive to such transformations. In fact, we can prove the following result:

Theorem 1. *Given a set of transformations $\mathcal{T}$, of which the distinctive transformations are injective, GDT is an unbiased estimate $\hat{\mathcal{L}}$ of the generalized contrastive loss (eq. 1), i.e. $\mathbb{E}[\hat{\mathcal{L}}] = \mathcal{L}(\Phi; \mathcal{T})$. Furthermore, consider a batch of sampled compositions of M transformations, with size $\prod_j^M K_j$, where K_m is the number of samples for the mth transformation. Define $K_I = \prod_{j \in I} K_j$ and $K_V = \prod_{j \in V} K_j$, where I and V are the subsets of indices corresponding to invariant and distinctive transformations, respectively. Denote by $\mathcal{L}_{jj'}$ and $\sigma^2_{jj'}$ the mean and variance of the partial sum of the objective (eq. 1) on the set $\mathcal{X}_j \times \mathcal{X}_{j'}$, with $\mathcal{X}_j = \{x(T^I, T^V_j) : T^I \in \mathcal{T}_I\}$, i.e. the sample pairs corresponding to distinctive transformations with indices j and j'. Then, the variance of the GDT estimate is*

$$\mathbb{V}[\hat{\mathcal{L}}] = \frac{1}{K_V^4 K_I^2} \sum_{jj'}^{K_V, K_V} \sigma^2_{jj'}.$$

The naive estimate's variance, on the other hand, is

$$\mathbb{V}[\hat{\mathcal{L}}_d] = \frac{1}{K_V^2 K_I^2} \sum_{jj'}^{K_V, K_V} \sigma^2_{jj'} + \frac{1}{K_V^2} \sum_{jj'}^{K_V, K_V} \left(\mathcal{L}_{jj'} - \mathcal{L}\right)^2,$$

which is larger by a multiplicative factor of K_V^2 and a further additive factor. *Proof:* See Appendix A. □

[2] Note that the sampling ordering is arbitrary; in particular, it needs not to be the same as the ordering in which transformations are applied to the data.

This states that sampling data with GDT yields reduced variance, resulting in higher-quality gradients for learning the *same* objective (since the estimate is unbiased), which is reflected empirically in our strong performance on numerous datasets and benchmarks. We note that this may apply to other methods built on the same sampling strategy but which compose transformations in different ways than GDT, as long as the requirements (i-iii) for forming a batch (sec. 3.1) are satisfied.

3.3. Application to video data

As a concrete instantiation of our framework, we consider video data and transformations of the type $T = (t_1, t_2, t_3, t_4, t_5) = (i, \tau, m, r, g)$, as shown in fig. 1, as follows. The first component i **selects** a video in the dataset. We sample $K_i \gg 2$ indices/videos and assume distinctiveness, so that $c(i, i') = \delta_{i=i'}$. The second component τ contrasts different **temporal shifts**. We sample $K_\tau = 2$ different values of a delay τ uniformly at random, extracting a 1s clip $x_{i\tau}$ starting at time τ. For this contrast, we will test the distinctiveness and invariance hypotheses, as [41] indicate that the former may be preferable. The third component m contrasts **modalities**, projecting the video $x_{i\tau}$ to either its visual or audio component $m(x_{i\tau})$. We assume invariance $c(m, m') = 1$ and always sample two such transformations m_v and m_a to extract both modalities, so $K_m = 2$. The fourth component r contrasts **time reversal** [63, 76], which has not previously been explored in a contrastive or cross-modal setting. This is given by a transformation $r \in \mathcal{R} = \{r_0, r_1\}$, where r_0 is the identity and r_1 flips the time dimension of its input tensor, so $K_r = 2$. The final component g applies a spatial and aural **augmentation** $x(T) = g(r(m(x_{i\tau})))$, also normalizing the data. We assume invariance $c(g, g') = 1$ and pick $K_g = 1$, i.e. augment each datum at this level in the sampling hierarchy. These choices lead to $K = K_i K_\tau K_m K_r K_g = 8K_i$ transformations T in the batch $\mathcal{T}$ (in ablations, we also test a subset of these choices).

While we focus on modality splitting, time reversal and shift, note that we could use any transformation that can yield a useful learning signal, such as speed [9, 14, 18, 36, 75, 81] and temporal ordering [20, 43, 53, 80].

Modality splitting. The modality splitting transformation m is useful to capture correlation between modalities [4, 7, 41, 60, 76]. Modality splitting means that the nature of the sample $x(i, \tau, m, r, g)$ is either a sequence of frames ($m = m_v$) or a sound ($m = m_a$). Formally, this means that $x(i, \tau, m, r, g)$ is an element of the direct sum $\mathcal{X}_v \oplus \mathcal{X}_a$ of visual and audio signals; likewise g, r and Φ are defined on this direct sum. In practice, this means that the transformation g comprises a pair of augmentations (g_v, g_a), where $g_v(v)$ extracts a fixed-size tensor by resizing to a fixed resolution of a random spatial crop of the input video v, and $g_a(a)$ extracts a spectrogram representation of the audio signal followed by SpecAugment [61] with frequency and time masking. Likewise, $\Phi = (\Phi_v, \Phi_a)$ comprises a pair of neural networks, one for each modality, both valued in $\mathbb{R}^d$ (refer to Appendix A.3.4 for architectural details). In the Appendix A.3.1, we show that modality splitting is key for performance; thus, we extend SimCLR weight function w to focus learning on only cross-modal pairs: $w(T, T) = \delta_{i \neq i} \cdot \delta_{m \neq m'}$.

3.4. Discussion: utility of GDT

With our framework, we can now generalize current state of the art contrastive learning approaches such as SimCLR in a systematic and practical manner. The theory above and in Appendix A.1 tells us what is the meaning of composing transformations, how a batch should be sampled and why, how this can be achieved by using a hierarchical sampling scheme that extends SimCLR, and what are the limitations of doing so. A particular benefit is to allow to specify individually, for each transformation, if invariance or distinctiveness is sought, whereas previous works lack this distinction and largely considered learning only invariances (SimCLR [13], AVID [54]), or distinctiveness (AoT [76]) to all factors. This property allows the flexible utilization of dataset specific transformations in the case of prior knowledge, or, as we have shown in this study, the exploration of useful signals by enumeration. Finding the best transformation signals can even be further optimized by methods such as Bayesian optimization. Finally, compared to a direct application of previous state-of-the-art methods image-based methods such as SimCLR [13], PIRL [52], and MoCo [31], we can also seamlessly incorporate important cues such as cross-modal correlation, greatly improving downstream performance (see table A.1).

4. Experiments

We compare self-supervised methods on pretraining audio-visual representations. Quality is assessed based on how well the pretrained representation transfers to downstream tasks. We conduct a study on video-audio, as well as video-text unsupervised representation learning to show the generality of our framework and then compare our best setup to the state of the art.

Self-supervised pretraining. For pretraining, we consider two standard pretraining datasets: Kinetics-400 [39] and HT100M [50] and use R(2+1)D-18 [72] and a 2D ResNet [32] as encoders (see Appendix for further details). We also explore how our algorithm scales to even larger, less-curated datasets and train on IG65M [21] as done in XDC [2].

Downstream tasks. To assess the pretrained representation f_v, we consider standard action recognition benchmark datasets, UCF-101 [67] and HMDB-51 [42]. We test the performance of our pretrained models on the tasks of finetuning

Table 1: **Learning hypothesis ablation.** Results on action classification performance on HMDB-51 is shown for finetuning accuracy (Acc) and frozen retrieval (recall@1) after pretraining on Kinetics-400 for 50 epochs. GDT can leverage signals from both invariance and stronger distinctiveness transformation signals. We consider data-sampling (DS), time-reversal (TR) and time-shifting (TS).

	DS	TR	TS	Mod.	**Acc.**	**R@1**
SimCLR-like: DS-distinctiveness only						
(a)	**d**	·	·	V	44.6	11.8
(b)	**d**	**i**	·	V	36.9	13.3
(c)	**d**	·	**i**	V	35.9	15.3
(d)	**d**	**i**	**i**	V	37.8	13.9
Cross-modal						
(e)	**d**	·	·	AV	52.4	21.8
(f)	**d**	**i**	·	AV	58.8	22.6
(g)	**d**	·	**i**	AV	57.4	23.5
(h)	**d**	**i**	**i**	AV	59.9	24.8
Cross-modal +1 distinctive factor						
(i)	**d**	**d**	·	AV	57.8	26.1
(j)	**d**	·	**d**	AV	58.7	22.1
(k)	**d**	**d**	**i**	AV	61.1	25.4
(l)	**d**	**i**	**d**	AV	**61.4**	**27.1**
Cross-modal + 2 distinctive factors						
(m)	**d**	**d**	**d**	AV	57.2	20.5

the pretrained representation, conducting few-shot learning and video action retrieval. The full details are given in the Appendix.

4.1. Analysis of generalized data transformations

In this section, we conduct an extensive study on each parameter of the GDT transformation studied here, $T = (i, \tau, m, r, g)$, and evaluate the performance by finetuning our network and conduncting video retrieval on the HMDB-51 action recognition benchmark.

SimCLR-like baseline. First, we use the framework to test a direct extension of SimCLR to video data, as shown in Table 1(a)-(d). By this, we mean utilizing only the visual modality (V), and only invariance to transformations, which is standard in all recent self-supervised methods [13, 31, 77]. For this, we consider GDTs of the type $T = (i, m, \tau, r, g)$ described above and set $K_i = 512$ (the largest we can fit in our setup). In row (a), we pick only the video modality ($m = m_v$ so $K_m = 1$). We also sample a single shift τ (so $K_\tau = 1$), which results in data augmentation but does not learn shift invariance, and no time reversal $r = 1$ (so $K_r = 1$) — these are denoted with a · in the table. However, we do sample two visual augmentations g ($K_g = 2$), emulating SimCLR and learning invariance to that factor.

Table 2: **GDT on video-text HT100M dataset**. We also find the positive effect of including more modalities and find non-trivial combinations of beneficial transformations previously unexplored.

	DS	TR	TS	Mod.	**Acc**
SimCLR-like					
(a)	**d**	·	·	V	36.1
Video-text cross-modal					
(b)	**d**	·	·	VT	59.2
(c)	**d**	**d**	·	VT	61.5
(d)	**d**	·	**d**	VT	62.9
(e)	**d**	**d**	**i**	VT	63.8
(f)	**d**	**i**	**d**	VT	**64.4**
(g)	**d**	**d**	**d**	VT	**64.4**

We also set all transformation components to invariance ($c(t_m, t'_m) = 1$) except the first that does sample selection. In row (b-d) we also experiment with adding invariance to *time shift* (TS) and *time reversal* (TR), by setting $K_\tau = 2$ and $K_r = 2$. We find that doing so consistently degrades the finetuning accuracy performance, but increases the retrieval performance somewhat, indicating that the model is not able to fully leverage these augmentation signals in a meaningful way.

Cross-modal learning. Next, in rows (e-h) we repeat this experiment, but using both audio-visual modalities (AV) by setting $K_m = 2$. In this case, as explained above, we set the weight w to only consider cross-modal interactions and set $K_g = 1$. We note two facts: First, the performance increases substantially (+7.8% (e) vs (a-d)). Second, now TS and TR invariance leads to significant improvements (up to +7.5%).

Invariance vs distinctiveness. Next, in rows (i-l), we explore the effect of being invariant or distinctive to individual transformations, which is unique to our method. Comparing row (h) to rows (k) and (l), we see that switching to distinctiveness to one of TS or TR further improves performance (up to +1.5%). On the other hand, 'ignoring' either (· symbols in lines (g) and (j)) is worse than learning invariance ((h) and (l)), with a difference of around 2.5%. Finally, in row (m) we find that being distinctive for both TS and TR at the same time is worse, suggesting that a mix of distinctiveness and invariance is preferable. This is particularly true for the retrieval metric (column R@1).

4.2. Textual modality

In table 2, we demonstrate the generality of our approach by using ASR captions as an alternative modality (instead of audio) for the HowTo100M dataset [50]. For the text encoder, we use a simple Word2Vec [51] embedding with a MLP (further details are provided in the Appendix). Com-

paring table 2(a) with (b), we find that switching from SimCLR to a cross-modal baseline increases performance by more than +22%. Furthermore, we find gains of 3.7% when switching from data-sampling distinctiveness only (row (b)) to incorporating further distinctivenesses (rows c-d). Finally, we find that – as in the video-audio case – combining time-shift distinctiveness with time-reversal invariance leads to particularly strong representations (row (f)), yielding benefits of over +5% compared to data-sampling distinctiveness alone. Compared to video-audio learning (table 1(m)), we find the case of distinctive-only for video-text learning (table 2(g)) to be highly competitive, highlighting the need to explore the set of possible transformation signals to achieve the best downstream performance.

Intuition. While we only analyse a subset of possible transformations for video data, we nevertheless find consistent signals across both video-audio and video-text learning: Inclusion of further distinctivenesses to TS and TR always improve upon the basecase and the best setup is achieved for TS distinctiveness and TR invariance. One explanation for this might be that there is useful signal in both of these transformations that are not captured by previous "augmentation-only" naive noise-contrastive formulations. For example, for time-shift (TS), the model profits from having to differentiate different points in time, e.g. between an athlete running vs an athlete landing in a sandpit, which could be both in the same video. This intuitively serves as a hard negative for the model, increasing its discriminative power. For time reversal (TR), many actions depicted such as moving an object are inherently invariant to reversing time, as shown in [65], therefore yielding a gain when used as an augmentation. In [76], they show that humans have a 20% error-rate when classifying a video's direction of time in Kinetics-400, thus demonstrating that Kinetics-400 has subsets of videos that look realistic even when reversed. These findings that additional distinctiveness combined with invariances improve video representation learning are noteworthy, as they contradict results from the image self-supervised learning domain, where learning pretext-invariance can lead to more transferable representations [52]. Even when compared to previous self-supervised learning approaches for video-data, such as predicting the arrow of time [76], our method yields new insights by showing that a unique combination of distinctivenesses and invariances performs best, at least on the training sets considered. Combining these points, the strong performance of GDT is founded in its ability to leverage highly informative, yet "free", signal that we have from construction of the transformations.

4.3. Qualitative analysis

Here, we study what effect the different transformations we let our model be invariant and distinctive to have on our learned representations. For this, we compare against the SimCLR baseline of Tab.1(a) and compare the average standard deviation of the normalized features for 10 time-shifted clips per video for 3000 randomly selected Kinetics-400 validation set.

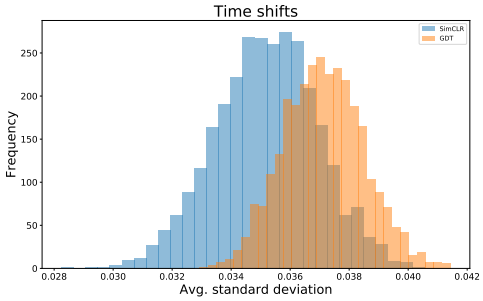


Fig. 3: Learning distinctiveness to time-shifts: our GDT model from Tab.1(j) is able to differentiate features from the same video at different times better than a simple SimCLR variant (Tab.1(a)).

Table 3: **Video retrieval and Few Shot Learning.** Retrieval accuracy in (%) via nearest neighbors at various levels of neighborhood sizes and few shot learning accuracy (%) via a k-nearest neighbor on frozen representations.

		HMDB		UCF	
		1	5	1	5
kNN	Random	3.0	3.5	2.3	4.6
	3DRot [38]	–	–	15.0	31.5
	GDT (ours)	**14.3**	**15.4**	**26.7**	**44.6**
Retrieval	SP-Net [9]	-	-	13.0	28.1
	VCP [14]	7.6	24.4	18.6	33.6
	M-DPC [29]	7.7	25.7	20.2	40.4
	VSP [14]	10.3	26.6	24.6	41.9
	CoCLR [30]	23.2	43.2	53.3	69.4
	SeLaVi [5]	24.8	47.6	52.0	68.6
	GDT (ours)	**26.1**	**51.7**	**62.8**	**79.0**

4.4. Comparison to the state of the art

Given one of our best learning setups from Sec. 4.1 (row (l)), we train for longer and compare our feature representations to the state of the art on standard downstream benchmarks.

4.4.1 Downstream benchmarks

For **few-shot classification**, as shown in table 3, we significantly beat the 3D-Rotnet [38] baseline on UCF-101 by more than 10% on average for each shot with our Kinetics-400 pretrained model.

Table 4: **State-of-the-art on video action recognition with full-finetuning.** Self- and fully-supervisedly trained methods on UCF-101 and HMDB-51 benchmarks.

Method	Data	Top-1 Acc%	
		HMDB	UCF
Supervised [79]	K-400+IN	75.9	96.8
Supervised [2]	K-400	65.1	94.2
AoT [76]	K-400	-	79.4
MultiSensory [59]	K-400	-	82.1
SeLaVi [5]	K-400	47.1	84.2
PEMT [44]	K-400	–	85.2
XDC [2]	K-400	52.6	86.8
AV Sync+RotNet [78]	K-400	54.6	87.0
CoCLR [30]	K-400	54.6	87.9
SeCo [82]	K-400	55.6	88.3
AVTS [41]	K-400	56.9	85.8
CPD [46]	K-400	57.7	88.7
AVID [54]	K-400	60.8	87.5
CM-ACC [47]	K-400	61.8	90.2
GLCM [48]	K-400	61.9	91.2
GDT (ours)	K-400	**62.3**	**90.9**
MIL-NCE [49]	HT100M	61.0	91.3
GDT (ours)	HT100M	**67.4**	**94.1**
XDC [2]	IG65M	68.9	**95.5**
GDT (ours)	IG65M	**72.8**	95.2

For **video retrieval**, we report recall at 1 and 5 retrieved samples for split-1 of the HMDB-51 and UCF-101 datasets in table 3.Using our model trained on Kinetics-400, GDT significantly beats all other self-supervised methods. In particular, we outperform CoCLR [30], a recent state-of-the-art self-supervised method, that uses optical flow as another view to mine hard positive to improve instance discrimination learning for video representations. Moreover, we surpass SeLaVi, an audio-visual clustering and representation learning method, by 2% and 10% on average on recall at 1 and 5 for HMDB-51 and UCF-101.

For **video action recognition**, we finetune our GDT pretrained network for UCF-101 and HMDB-51 video classification, and compare against state-of-the-art self-supervised methods in table 4. When pretrained on the Kinetics datasets, we find that our GDT pretrained model achieves very good results, outperforming all recent methods. In particular, we outperform audio-visual pretraining methods, AVTS [41], SeLaVi [5] and XDC [2], by large margins using the same architecture (R(2+1)D-18) and dataset (Kinetics-400), showing the effectiveness of our GDT pre-training approach. We also surpass AVID [54], the state-of-the-art audio-visual representation learning method, by 1.5% on HMDB-51 and 3.8% on UCF-101. AVID uses a variant of the pre-training scheme of our baseline approach that extends noise contrastive learning to the audio-visual domain as in Table 1, row (e). However, while AVID simply encodes sample distinctiveness and invariance to modality in its visual representations, we are able to encode invariances and distinctiveness to additional transformations, which significantly improves our performance. Our approach is also more sample efficient, as we are able to achieve our results with 300 less epochs of training. Finally, when pretrained on HT100M, we achieve strong gains of $+6.4\%$ on HMDB-51 and $+2.8\%$ on UCF-101 compared to the state-of-the-art video text method, MIL-NCE [49]. Similar to AVID, MIL-NCE uses a variant of the baseline cross-modal contrastive framework to learn representations, while we are able to improve upon this baseline by learning invariance and distinctiveness to additional transformations such as time reversal and time shift. Moreover, with HT100M pre-training, we outperform the Kinetics supervised baseline using the same architecture when finetuned on HMDB-51 (67.4 vs 65.1) and are on par for UCF-101 (94.1 vs 94.2). We further show the scalability and flexibility of our GDT framework by pretraining on the IG65M dataset [21]. With this, our visual feature representation sets a new state of the art among all self-supervised methods, particularly by a margin of $>4\%$ on the HMDB-51 dataset. On UCF-101, we set similar state-of-the-art performance with XDC. Along with XDC, we beat the Kinetics supervised pretraining baseline using the same architecture and finetuning protocol.

5. Conclusion

We introduced the framework of Generalized Data Transformations (GDTs), which allows one to capture, in a single noise-contrastive objective, cues used in several prior contrastive and non-contrastive learning formulations, as well as easily incorporate new ones. The framework shows how new meaningful compositions of transformations can be obtained, encoding valuable invariance and distinctiveness that we want our representations to learn. Following this methodology, we achieved state-of-the-art results for self-supervised pretraining on standard downstream video action recognition benchmarks, even surpassing supervised pretraining. Overall, our method significantly increases the expressiveness of contrastive learning for self-supervision, making it a flexible tool for many multi-modal settings, where a large pool of transformations exist and an optimal combination is sought.

References

[1] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020.

[2] Humam Alwassel, Bruno Korbar, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020.

[3] Ankesh Anand, Evan Racah, Sherjil Ozair, Yoshua Bengio, Marc-Alexandre Côté, and R Devon Hjelm. Unsupervised state representation learning in atari. 2019.

[4] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017.

[5] Yuki M Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. *NeurIPS*, 2020.

[6] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[7] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016.

[8] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019.

[9] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, 2020.

[10] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[11] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[12] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020.

[14] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692*, 2020.

[15] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016.

[16] Virginia R. de Sa. Learning classification with unlabeled data. In *NeurIPS*, 1994.

[17] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[18] Dave Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *CVPR*, 2020.

[19] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proc. CVPR*, 2017.

[20] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017.

[21] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

[22] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words. In *CVPR*, 2020.

[23] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[24] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[25] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020.

[26] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

[27] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[28] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, 2019.

[29] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020.

[30] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020.

[31] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[33] Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019.

[34] R Devon Hjelm and Philip Bachman. Representation learning with video deep infomax, 2020.

[35] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR*, 2019.

[36] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *ECCV*, 2020.

[37] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation, 2018.

[38] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

[39] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[40] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019.

[41] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[42] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[43] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017.

[44] Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. In *ICLR*, 2021.

[45] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *ICLR*, 2021.

[46] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination. *arXiv preprint arXiv:2001.05691*, 2020.

[47] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Active contrastive learning of audio-visual video representations. In *ICLR*, 2021.

[48] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Contrastive self-supervised learning of global-local audio-visual representations, 2021.

[49] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

[50] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

[51] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[52] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

[53] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.

[54] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. 2020.

[55] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *CVPR*, 2020.

[56] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[57] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[58] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[59] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.

[60] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.

[61] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *INTERSPEECH*, 2019.

[62] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016.

[63] Lyndsey C Pickup, Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T Freeman. Seeing the arrow of time. In *CVPR*, 2014.

[64] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020.

[65] Will Price and Dima Damen. Retro-Actions: Learning 'close' by time-reversing 'open' videos. 2019.

[66] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *NeurIPS*, 2016.

[67] Khurram Soomro, Amir Roshan Zamir, and Mubarak

Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[68] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.

[69] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.

[70] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. 2020.

[71] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. *arXiv preprint arXiv:2005.10243*, 2020.

[72] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[73] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(2579-2605):85, 2008.

[74] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019.

[75] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020.

[76] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018.

[77] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

[78] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[79] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.

[80] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019.

[81] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv preprint arXiv:2006.15489*, 2020.

[82] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*, 2021.

[83] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *Proc. ECCV*, 2016.

[84] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017.

[85] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, 2019.

A. Appendix

A.1. Foundations of compositional contrastive learning

In this section, we develop more formally a basic theory of compositional contrastive learning formulation, providing rigorous grounds for the approach described in Sec. 3.

Consider the problem of learning a function $f : \mathcal{X} \to \mathcal{Y}$. In a contrastive setting, we are not given information about the values of f; instead, we are given a *contrast function* $c : \mathcal{X} \times \mathcal{X} \to \{0, 1\}$ which only tells for which pairs of points x_1 and x_2 f is the same and for which it differs:

Definition 1. *The function f is* compatible *with the contrast c if, and only if, for all $x_1, x_2 \in \mathcal{X}$:*

$$c(x_1, x_2) = \delta_{f(x_1)=f(x_2)}.$$

A contrast function cannot be arbitrary:

Lemma 1. *The predicate $c(x_1, x_2) = 1$ is an equivalence relation if, and only if, there exists a function f compatible with c.*

Proof. If $c(x_1, x_2) = 1$ defines an equivalence relation on $\mathcal{X}$, then such a function is given by the projection on the quotient $\hat{f} : \mathcal{X} \to \mathcal{X}/c = \mathcal{Y}$. On the other hand, setting $c(x_1, x_2) = \delta_{f(x_1)=f(x_2)} = 1$ for any given function f is reflexive, symmetric and transitive because the equality $f(x_1) = f(x_2)$ is. □

Definition 2. *The contrast function c is admissible if, and only if, $c(x_1, x_2) = 1$ defines an equivalence relation.*

Full knowledge of the contrast function c only specifies the level sets of the function f:

Lemma 2. *Let f be any function compatible with the admissible contrast c. Then, we can write $f = \iota \circ \hat{f}$ as the composition of an injection $\iota : \mathcal{X}/f \to \mathcal{Y}$ and the (unique) projection $\hat{f} : \mathcal{X} \to \mathcal{X}/c$ of $\mathcal{X}$ onto the equivalence classes $\mathcal{X}/c$ of the equivalence relation $c(x_1, x_2) = 1$.*

Proof. From elementary algebra, we can decompose any function $f : \mathcal{X} \to \mathcal{Y}$ as

$$f : \mathcal{X} \xrightarrow{\hat{f}} \mathcal{X}/f \xrightarrow{\iota} \mathcal{Y}$$

where ι is an injective function and $\hat{f}$ projects $\mathcal{X}$ to the quotient $\mathcal{X}/f$, i.e. the collection of subsets $X \subset \mathcal{X}$ where $f(x)$ is constant (level sets). The latter are also the equivalence classes of the relation $f(x_1) = f(x_2)$. Due to definition 1, this is the same equivalence relation given by the contrast c, so that $\mathcal{X}/f = \mathcal{X}/c$. □

Note that, in our contrastive learning formulation, we do *not* define the contrast c on the sample space $\mathcal{X}$, but rather on the transformation space $\mathcal{T}$. The following lemma suggests that defining a contrast $c(T, T')$ on transformations instead of data samples is usually acceptable:

Lemma 3. *Let $c : \mathcal{T} \times \mathcal{T} \to \{0, 1\}$ be an admissible contrast function defined on a set (e.g., a batch) of generalized data transformations $\mathcal{T}$. Furthermore, let x be a dataset and let $x(T) \in \mathcal{X}$ be the sample indexed by transformation T. If $x(T) = x(T') \Rightarrow T = T'$ (i.e. different transformations output different samples), then setting $\tilde{c}(x(T), x(T')) = c(T, T')$ defines part of an admissible sample contrast function $\tilde{c}$ on $\mathcal{X}$.*

Proof. The expression above defines $\tilde{c}$ on the sample space $\tilde{\mathcal{X}} = \{x(T) : T \in \mathcal{T}\} \subset \mathcal{X}$. Reflexivity, symmetry and transitivity are inherited from c. However, if the same data point $x(T) = x(T')$ can be obtained from two different transformations T and T', the definition is ill posed. The hypothesis in the lemma guarantees that this is not the case. □

A.1.1 Compositional transformations

Next, we consider the case in which $T = (t_1, \ldots, t_M)$ is a composition of individual transformations t_m, each with its own contrast t_m:

Definition 3. *We say that a contrast function $c(t_m, {t'}_m)$ is distinctive if it is given by $\delta_{t_m={t'}_m}$. We say that it is invariant if it is identically one.*

The following lemma provides a formula for the overall contrast function $c(T, T)$ given the contrasts for the individual factors.

Lemma 4. *Let $c(t_m, t'_m) = 1$ be admissible contrast functions, either distinctive or invariant. Then, the product $c(T, T') = \prod_{m=1}^{M} c(t_m, t'_m)$ is also admissible.*

Proof. The reflexive and symmetric properties are obviously inherited. For the transitive property, note that $c(T, T') = 1$ if, and only if, $\forall m : c(t_m, t'_m) = 1$. Hence:

$$\begin{aligned} c(T, T') = c(T', T'') = 1 \\ \Rightarrow \quad \forall m : c(t_m, t'_m) = c(t'_m, t''_m) = 1 \\ \Rightarrow \quad \forall m : c(t_m, t''_m) = 1 \quad \Rightarrow \quad c(T, T'') = 1. \end{aligned}$$

□

Finally, we show that, essentially, the formula above is the only reasonable one. For this, we only require $c(T, T')$ to be monotonic in the individual factors; i.e., if more factors become 1, then $c(T, T')$ can only grow:

Definition 4. *We say that $c(T, T')$ is monotonic in the individual factors if, and only if, for any three transformations T, T', T'' such that $c(t_m, t'_m) \leq c(t_m, t''_m)$ for all the factors, then we also have $c(T, T') \leq c(T, T'')$.*

Next, we show that c can only have a very limited form:

Lemma 5. *Suppose that the admissible monotonic contrast $c(T, T')$ is expressible solely as a function of the individual admissible contrasts $c(t_m, t'_m)$ for $m = 1, \ldots, M$. Then, up to a permutation of the transformations, we can always write*

$$c(T, T') = \prod_{i=1}^{m} c(t_i, t'_i)$$

where $0 \leq m \leq M$. In particular, $m = M$ is the only option that is guaranteed not to ignore some of the factors.

Proof. From the assumptions, we can write

$$c(T, T') = h \circ v(T, T')$$

where h is a function of the binary vector

$$v(T, T') = (c(t_1, t'_1), \ldots, c(t_M, t'_M)) \in \mathbb{B}^M.$$

Furthermore, since invariant factors are constant, they do not affect the function; hence, without loss of generality we can assume that all factors are distinctive.

Since all factors are distinctive, we can construct two transformations $T' = (t'_1, \ldots, t'_M)$ and $T'' = (t''_1, \ldots, t''_M)$ such that $v(T', T'') = (0, \ldots, 0)$ (i.e., all the contrasts $c(t'_m, t''_m)$ are null). If $c(T', T'') = 1$, then, due to monotonicity, $c(T', T'')$ is identically 1 and the lemma is proved for $m = 0$.

If not, let $c(T', T'') = h(0, \ldots, 0) = 0$. Then, for any given binary vector v, we can construct a transformation $T = (t_1, \ldots, t_M)$ such that $v(T, T') = v$ and $v(T, T'') = \bar{v}$ as follows:

$$t_m = \begin{cases} t'_m, & \text{if } v_m = 1, \\ t''_m, & \text{otherwise.} \end{cases}$$

We cannot have $c(T, T') = h(v) = h(\bar{v}) = c(T, T'') = 1$; otherwise, due to the transitivity of c, we would have $c(T', T'') = h(0, \ldots, 0) = 1$, which contradicts our assumption. Hence, h must partition the space of binary vectors in two halves, the ones for which $h(v) = 1$ and their complements $h(\bar{v}) = 0$.

Now let v be a vector with the minimal number of 1 such that $h(v) = 1$. Again without loss of generality, we can assume this is of the type $v = (1, \ldots, 1, 0, \ldots, 0)$ with m ones in front. Due to monotonicity, all vectors of type $v' = (1, \ldots, 1, v_{m+1}, \ldots, v_M)$ must also have $h(v') = 1$; by taking their complement, the previous result shows that all vectors $v'' = (0, \ldots, 0, v_{m+1}, \ldots, v_M)$ must have $h(v'') = 0$. This is also the case for any vector of the type $(v_1, \ldots, v_m, 0, \ldots, 0)$ where any $v_i = 0$ for $1 \leq i \leq m$ (because m is the minimum number of ones required for $h(v) = 1$). We conclude that $h(v) = 1$ if, and only if, $(v_1, \ldots, v_m) = (1, \ldots, 1)$. □

A.1.2 Forming batches

Let $\hat{\mathcal{T}}_1 \times \cdots \times \hat{\mathcal{T}}_M$ be a composite space of generalized data transformations, so that data points are indexed as $x(t_1, \ldots, t_M)$. Furthermore, let $c(t_m, t'_m)$ be corresponding admissible contrast functions and let $c(T, T')$ be their product, as in lemma 4. As before, we assume that the functions are of two kinds:

- invariant: $c(t_m, t'_m) = 1$.
- distinctive: $c(t_m, t'_m) = \delta_{t_m = t'_m}$.

Let $I \subset \{1, \ldots, M\}$ be the subset of indices m corresponding to the invariant transformations and $D = \{1, \ldots, M\} \backslash I$ the distinctive ones.

Let $\text{sample}(\hat{\mathcal{T}}_m; K_m)$ be a stochastic operator that samples $K_m \leq |\hat{\mathcal{T}}_m|$ transformations from $\hat{\mathcal{T}}_m$ without replacement. We sample a batch recursively:

- Let $\mathcal{T}_1 = \text{sample}(\hat{\mathcal{T}}_1; K_1)$
- Let $\mathcal{T}_m = \bigcup_{T \in \mathcal{T}_{m-1}} T \circ \text{sample}(\hat{\mathcal{T}}_m; K_m)$

At each level of the recursion, each transformation is extended by sampling K_m more transformations (note that no two identical transformations can be generated in this manner). Hence $|\mathcal{T}_M| = K_1 \cdots K_M$.

Lemma 6. *There are exactly $(\prod_m K_m)(\prod_{m \in I} K_m)$ pairs of transformations $(T, T') \in \mathcal{T}_M \times \mathcal{T}_M$ for which $c(T, T') = 1$. Of these, exactly $\prod_m K_m$ are trivial pairs ($T = T'$). Hence, there are $(\prod_m K_m)(\prod_{m \in I} K_m - 1)$ non-trivial pairs for which $c(T, T') = 1$.*

Lemma 7. *For each $T \in \mathcal{T}_M$, there are exactly $(\prod_m K_m) - (\prod_{m \in I} K_m)$ pairs (T, T'') such that $c(T, T'') = 0$.*

For example, in SimCLR $M = 2$, $D = \{1\}$, $I = \{2\}$, $K_1 = B/2$, $K_2 = 2$, $|\mathcal{T}_2| = B$, there are $B(2-1) = B$ non-trivial pairs of transformations for which $c(T, T') = 1$, and, for each T, there are $B - 2$ pairs of transformations for which $c(T, T'') = 0$.

The lemmas above suggest that we should pick $K_m \geq 2$ for at least one invariant factor and at least $K_m \geq 2$ for at least one distinctive factor, as otherwise eq. (1) is degenerate.

A.1.3 Limitations

In general, we want more restrictive requirements than the one described above. When learning f, difficult (and therefore interesting) cases amount to: learning to be sensitive

to 'small' variations in the distinctive factors and learn to be insensitive to 'large' variations in the invariant factors. For the former, we would like f to observe variations in a single distinctive factor at a time, as these are the 'smallest'. For these individual variations to exist at all in the batch, we should choose $K_m \geq 2$ for all distinctive factors $m \in D$.

Even so, the hierarchical scheme in general *prevents* us from observing all individual variations. In fact, suppose that two transformations T and T' in $\mathcal{T}_M$ differ for factor m (i.e. $t_m \neq t'_m$). Then, the remaining factors $t_{m+1} \neq t'_{m+1}, \ldots$ also differ in general because successive transformations are sampled independently in different branches of the tree. This means that we cannot, in general, observe a change in t_m *alone*, so the function f may not learn to be distinctive to this 'minimal' change in isolation.

Note that this is a limitation that affects our sampling scheme as well as existing methods such as SimCLR. Fortunately, in practice this is often not an issue. There are in fact two mitigating factors, which apply to most existing formulations, including the new ones presented here.

First, some transformations spaces $\hat{\mathcal{T}}_m$ are very small, and in fact binary (e.g., modality splitting, time reversal). In this case, $K_m = 2$ means that transformations are sampled exhaustively, so for level m the hierarchical sampling scheme does extract all possible combinations of transformations.

Second, in other cases the issue is moot due to the nature of the transformations and the data. For instance, in SimCLR the first transformation t_1 amounts to sampling a certain image x_i, and the second transformation t_2 amounts to sampling two data augmentation $g_{1i}(x_i)$ and $g_{2i}(x_i)$, different for each image. The issue here is that we cannot observe a change in the index i for the same augmentation $g(x_1)$ and $g(x_2)$, as these data points do not exist in the batch. This means that the representation f can only learn to distinguish two different images x_i that also have two different augmentations applied to them. Because of the particular nature of the training data (ImageNet) this is likely irrelevant since different images x_i are unrelated in any case, so applying transformations does not significantly alter their statistical relationships.

However, note that this is not *always* the case. For instance, if SimCLR was applied to a dataset of pre-aligned faces (for the purpose of learning face recognition), then being unable to contrast different faces $g(x_1)$ and $g(x_2)$ under the same transformation g would make negative pairs far to easy to discriminate.

A.2. Reduction in variance theorem

A.2.1 Proof of theorem 1

For ease of notation, we will express eq. 1 as the expected value of a loss function ℓ, which subsumes the weight (w), contrast (c), feature extractor (Φ) and log-softmax functions:

$$\mathcal{L} = \mathbb{E}_{T,T' \sim \hat{\mathcal{T}}} \left[\ell \left(x(T), x(T') \right) \right]. \tag{2}$$

The expectation is over pairs of transformations in $\hat{\mathcal{T}} = \hat{\mathcal{T}}_1 \times \ldots \times \hat{\mathcal{T}}_M$, the space of all compositions of transformations, which can be applied to the data x. Note that eq. 1 contains a sum over a third transformation (T'') to compute the softmax's normalization, which is also subsumed by ℓ in eq. 2 as this third transformation is not essential for the rest of the proof. We will separate each transformation into invariant and distinctive parts, $T = (T^I, T^V)$ respectively with $T^I \in \hat{\mathcal{T}}_I$ and $T^V \in \hat{\mathcal{T}}_V$ (see sec. A.1.2). Note that this separation is merely a notational convenience; the individual transformations can be applied to the data in *any* order, with $x(T^I, T^V) = x(t_1 \circ \ldots \circ t_M)$, and each t_i belonging to either T^I or T^V. Then, eq. 2 becomes:

$$\mathcal{L} = \mathbb{E}_{T^I, T'^I \sim \hat{\mathcal{T}}_I,\, T^V, T'^V \sim \hat{\mathcal{T}}_V} \left[\ell \left(x(T^I, T^V), x'(T'^I, T'^V) \right) \right].$$

Consider a mini-batch of data sample pairs and their associated transformation compositions, $\mathcal{B}_{\text{direct}} = \left\{ T_i^I, T_i^V, T'^I_i, T'^V_i \right\}_{i=1}^{K_I^2 K_V^2}$, sampled as $T_i^I, T'^I_i \sim \hat{\mathcal{T}}_I$ and $T_i^V, T'^V_i \sim \hat{\mathcal{T}}_V$. The batch size is a function of $K_I = \prod_{j \in I} K_j$ and $K_V = \prod_{j \in V} K_j$, the number of sampled invariant and distinctive transformations in our method, respectively. The batch size of $K_I^2 K_V^2$ was chosen to allow a direct comparison. The expected value of the loss over this batch is then the simple empirical average:

$$\hat{\mathcal{L}}_{\text{d}} = \frac{1}{K_I^2 K_V^2} \sum_i^{K_I^2 K_V^2} \ell \left(x(T_i^I, T_i^V), x(T'^I_i, T'^V_i) \right). \tag{3}$$

Now consider the domain of transformed samples $\mathcal{X} = \{ x(T^I, T^V) : T^I \in \hat{\mathcal{T}}_I, T^V \in \hat{\mathcal{T}}_V \}$. Due to the assumed injectivity of all $t \in T^I$, we may partition the domain using one partition $\mathcal{X}_j = \{ x(T^I, T_j^V) : T^I \in \hat{\mathcal{T}}_I \}$ per distinctive transformation T_j^V, i.e.: $\mathcal{X} = \cup_j^{K_V} \mathcal{X}_j$, with $\mathcal{X}_j \cap \mathcal{X}_{j'} = \emptyset$, $\forall j, j'$. The probability distribution of the samples has density $p(T^I, T^V)$, and the density in each partition is thus $p_j(T^I) = K_V p(T^I) \delta_{T^I \in \mathcal{X}_j}$, with δ the indicator function.

GDT can then be interpreted as a stratified sampling method, with one stratum (partition) per *pair* of distinctive transformations. The domain being sampled by the expectation in eq. 2 is $\mathcal{X}^2 = \cup_{jj'}^{K_V, K_V} \mathcal{X}_j \times \mathcal{X}_{j'}$, and stratified sampling consists of sampling an equal number of K_I^2 sample pairs from each of the K_V^2 partitions:

$$\hat{\mathcal{L}} = \frac{1}{K_V^2 K_I^2} \sum_{ii'jj'}^{K_I, K_I, K_V, K_V} \ell \left(x(T_i^I, T_j^V), x(T_{i'}^I, T_{j'}^V) \right). \tag{4}$$

Note the subtle difference from eq. 3 in the summation ranges, and that the *same* samples and transformations are

reused for both elements of each pair, instead of being sampled independently to fill a mini-batch. To make the following derivations easier, note that we can equivalently express eq. 4 as:

$$\hat{\mathcal{L}} = \frac{1}{K_V^2} \sum_{jj'}^{K_V,K_V} \hat{\mathcal{L}}_{jj'},$$

with $\hat{\mathcal{L}}_{jj'} = \frac{1}{K_I^2} \sum_{ii'}^{K_I,K_I} \ell\left(x(T_i^I, T_j^V), x(T_{i'}^I, T_{j'}^V)\right)$. We will first show that this pairwise stratified sampling is an unbiased estimate of eq. 2:

$$\begin{aligned}
\mathbb{E}[\hat{\mathcal{L}}] &= \frac{1}{K_V^2} \sum_{jj'}^{K_V,K_V} \mathbb{E}[\hat{\mathcal{L}}_{jj'}] \\
&= \frac{1}{K_V^2} \sum_{jj'}^{K_V,K_V} \mathcal{L}_{jj'} \\
&= \mathcal{L},
\end{aligned}$$

where we use the expectation $\mathcal{L}_{jj'}$ of the loss function evaluated on the partition jj' (corresponding to distinctive transformations with indices j and j'), as $\mathcal{L}_{jj'} = \mathbb{E}_{T^I \in \mathcal{X}_j, T'^I \in \mathcal{X}_{j'}}\left[\ell\left(x(T^I, T_j^V), x(T'^I, T_{j'}^V)\right)\right]$.

Similarly, we can also define each partition's loss variance $\sigma_{jj'}^2 = \mathbb{V}_{T^I \in \mathcal{X}_j, T'^I \in \mathcal{X}_{j'}}\left[\ell\left(x(T^I, T_j^V), x(T'^I, T_{j'}^V)\right)\right]$. Then, from eq. 4 we obtain directly

$$\begin{aligned}
\mathbb{V}[\hat{\mathcal{L}}] &= \frac{1}{K_V^4} \sum_{jj'}^{K_V,K_V} \mathbb{V}[\mathcal{L}_{jj'}] \\
&= \frac{1}{K_V^4 K_I^2} \sum_{jj'}^{K_V,K_V} \sigma_{jj'}^2.
\end{aligned}$$

As a point of comparison, the variance of the direct sampling estimate is:

$$\begin{aligned}
\mathbb{V}[\hat{\mathcal{L}}_{\text{d}}] &= \frac{1}{K_V^2 K_I^2} \left((\mathbb{E}_{(T^I, T'^I) \in \mathcal{X}^2}[\ell^2(x(T^I, T^V), x(T'^I, T'^V) \right. \\
&\qquad \left. x(T'^I, T'^V))] - \mathcal{L}^2) \right) \\
&= \frac{1}{K_V^2 K_I^2} \left(\frac{1}{K_V^2} \sum_{jj'}^{K_V,K_V} \mathbb{E}_{T^I \in \mathcal{X}_j, T'^I \in \mathcal{X}_{j'}} \right. \\
&\qquad \left. \left[\ell^2\left(x(T^I, T_j^V), x(T'^I, T_{j'}^V)\right)\right] - \mathcal{L}^2 \right) \\
&= \frac{1}{K_V^2 K_I^2} \left(\frac{1}{K_V^2} \sum_{jj'}^{K_V,K_V} \left(\mathcal{L}_{jj'}^2 + \sigma_{jj'}^2\right) - \mathcal{L}^2 \right) \\
&= \frac{1}{K_V^4 K_I^2} \sum_{jj'}^{K_V,K_V} \left(\sigma_{jj'}^2 + \left(\mathcal{L}_{jj'} - \mathcal{L}\right)^2\right) \\
&\geq \frac{1}{K_V^4 K_I^2} \sum_{jj'}^{K_V,K_V} \sigma_{jj'}^2
\end{aligned}$$

completing the proof. □

A.3. Additional experimental results

A.3.1 Modality ablation

In table A.1, we provide the results of running our baseline model (sample-distinctiveness only) within-modally instead of across modalities and find a sharp drop in performance.

Table A.1: **Within vs cross-modal learning.** Results on action classification performance on HMDB-51 and UCF-101 is shown for finetuning accuracy (Acc) and frozen retrieval (recall@1) after pretraining on Kinetics-400 for 50 epochs.

	HMDB		UCF	
	Acc.	**R@1**	**Acc.**	**R@1**
Within-modal	37.8	13.9	76.4	28.0
Cross-modal	**52.4**	**21.8**	**87.6**	**54.8**

A.3.2 Dataset details

The Kinetics-400 dataset [39] is human action video dataset, consisting of 240k training videos, with each video representing one of 400 action classes. After filtering out videos without audio, we are left with 230k training videos, which we use for pretraining our model.

HT100M [50] is a large-scale instructional video collection of 1.2 million Youtube videos, along with automatic speech recognition transcripts. There are more than 100 million clips (ASR segments) defined in HowTo100M.

HMDB-51 [42] consists of 7K video clips spanning 51 different human activities. HMDB-51 has three train/test splits of size 5k/2k respectively.

UCF-101 [67] contains 13K videos from 101 human action classes, and has three train/test splits of size 11k/2k respectively.

IG65M [21] is a large-scale weakly supervised dataset collected from a social media website, consisting of 65M videos of human action events. We use the all the videos in the dataset for pretraining.

A.3.3 Preprocessing details

The video inputs are 30 consecutive frames from a randomly chosen starting point in the video. These frames are resized such that the shorter side is between 128 and 160, and a center crop of size 112 is extracted, with color-jittering applied. A random horizontal flip is then applied with probability 0.5, and then the inputs' channels are z-normalized using

mean and standard deviation statistics calculated across each dataset.

One second of audio is processed as a $1 \times 40 \times 99$ image, by taking the log-mel bank features with 40 filters and 99 time-frames after random volume jittering between 90% and 110% is applied to raw waveform, similar to [4]. The spectrogram is then Z-normalized, as in [41]. Spec-Augment is then used to apply random frequency masking to the spectrogram with maximal blocking width 3 and sampled 1 times. Similarly, time-masking is applied with maximum width 6 and sampled 1 times.

For the text, we remove stop words from the narrations as in [50]. For each narration, we take a maximum of 16 consecutive words covering a max duration of 4 seconds as in [49].

A.3.4 Pretraining details

We use R(2+1)D-18 [72] as the visual encoder f_v and ResNet [32] with 9 layers as the audio encoder f_a unless otherwise noted; both encoders produce a fixed-dimensional output (512-D) after global spatio-temporal average pooling. For the text encoder, we use the Google News self-supervised pre-trained word2vec (d=300) embedding [51], that is linearly projected to 2048D and max-pooled as in [49]. After the inputs are encoded by their respective modality encoders, the vectors are then passed through two fully-connected layers with intermediate size of 512 to produce 256-D embeddings as in [8] which are normalized by their L2-norm [77]. The embedding is used for computing the contrastive loss, while for downstream tasks, a linear layer after the global spatio-temporal average pooling is randomly initialized. For NCE contrastive learning, the temperature ρ is set as $1/0.07$. For optimizing these networks, we use SGD. The SGD weight decay is 10^{-5} and the SGD momentum is 0.9. We use a mini-batch size of 8 on each of our 64 GPUs giving an effective batch size of 512 for distributed training. The initial learning rate is set to 0.01 which we linearly scale with the number of GPUs, after following a gradual warm-up schedule for the first 10 epochs [24]. For Kinetics, we train for 100 epochs (3 days), while for HT100M, we train for 40 epochs (3 days).

A.3.5 Ablation experiment details

For the ablations, we only pretrain for 50 epochs on the Kinetics-400 dataset, and 20 epochs on the HT100M dataset, since it is a much larger dataset.

For downstream evaluation, we only evaluate on the first fold of HMDB-51 but found the performance between folds to be close (within 1%).

A.3.6 Evaluation details

All evaluation code is provided in the Supplementary Material.

Video Action Recognition. During training, we take 10 random clips of length 32 frames from each video. For video clip augmentations, we follow a standard protocol as in [41]. During evaluation, we uniformly sample 10 clips from each video, average softmax scores, and predict the class having the highest mean softmax score. We then measure the mean video top-1 accuracy across all videos and all official folds. During training, we use SGD with initial learning rate 0.0025, which we gradually warm up to $2 \cdot 10^{-2}$ in the first 2 epochs. The weight decay is set to $5 \cdot 10^{-3}$ and momentum to 0.9. We use a mini-batch size of 32 and train for 12 epochs with the learning rate multiplied by $5 \cdot 10^{-2}$ at 6 and 10 epochs. We compare our GDT pretrained model with both self-supervised methods, and supervised pretraining, and report average top-1 accuracies on UCF101 and HMDB-51 action recognition task across three folds in table 4.

Few-shot classification We follow the protocol in [38] and evaluate our our GDT pretrained network using few-shot classification on the UCF-101 dataset, and additionally on HMDB-51. We randomly sample n videos per class from the train set, average the encoder's global average pooled features from ten clips per training sample and measure classification accuracy performance on the validation set using a k-nearest neighbor classifier, with k set to 1.

Video Retrieval. We follow the standard protocol as outlined in [80]. We use the split 1 of UCF101, and additionally HMDB-51. We uniformly sample 10 clips per video, and average the max-pooled features after the last residual block for each clip per video. We use these averaged features from the validation set to query the videos in the training set. The cosine distance of representations between the query clip and all clips in the training set are computed. When the class of a test clip appears in the classes of k nearest training clips, it is considered to be correctly predicted. We report accuracies for $k = 1, 5, 20$ and compare with other self-supervised methods on UCF101 and HMDB-51 in table 3.

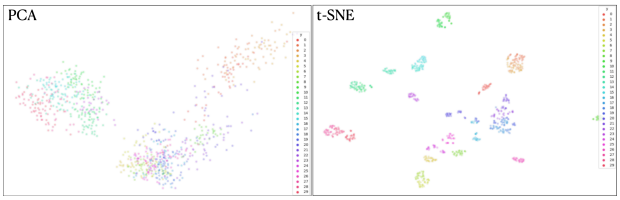


Fig. A.1: Feature visualizations with PCA and t-SNE on 30 videos of a single, random class of HMDB-51. For each video, we sample 10 temporal clips and encode video-ID with color. Embeddings are generated from our time-shift distinct model (Tab.1 (l)).

A.3.7 Additional Qualitative analysis

In fig. A.1, we present a PCA and t-SNE [73] plots of the features obtained by our model (DS-d, TR-d, TS-d) (Tab. 1, row (l)). We observe that even comparing to videos of the same action category, the individual clips are well separated, showing that the model is learning to distinguish between different time intervals.

3

Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning

This work will be presented at the International Conference on Computer Vision (ICCV) 2021.

Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning.

Mandela Patrick*, Po-Yao Huang*, Ishan Misra, Florian Metze, Andrea Vedaldi
Facebook AI Research
mandelapatrick,berniehuang,imisra,fmetze,vedaldi@fb.com

Yuki M. Asano*, João Henriques
Oxford University
yuki, joao@robots.ox.ac.uk

Abstract

The quality of the image representations obtained from self-supervised learning depends strongly on the type of data augmentations used in the learning formulation. Recent papers have ported these methods from still images to videos and found that leveraging both audio and video signals yields strong gains; however, they did not find that spatial augmentations such as cropping, which are very important for still images, work as well for videos. In this paper, we improve these formulations in two ways unique to the spatio-temporal aspect of videos. First, for space, we show that spatial augmentations such as cropping do work well for videos too, but that previous implementations, due to the high processing and memory cost, could not do this at a scale sufficient for it to work well. To address this issue, we first introduce Feature Crop, a method to simulate such augmentations much more efficiently directly in feature space. Second, we show that as opposed to naïve average pooling, the use of transformer-based attention improves performance significantly, and is well suited for processing feature crops. Combining both of our discoveries into a new method, Space-Time Crop & Attend (STiCA) we achieve state-of-the-art performance across multiple video-representation learning benchmarks. In particular, we achieve new state-of-the-art accuracies of 67.0% *on HMDB-51 and* 93.1% *on UCF-101 when pre-training on Kinetics-400. Code and pretrained models are available*[1].

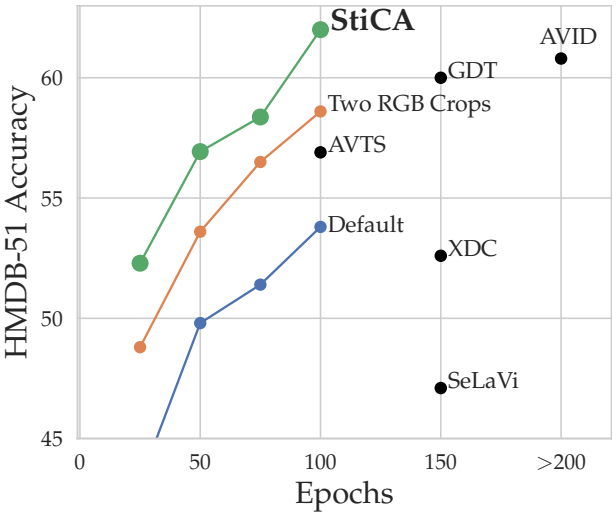


Figure 1: HMDB-51 accuracy vs epoch. Our method, **STiCA**, combines space-time crops in feature space with self-attention of time in latent space. This yields significant benefits not only in performance but also in speed compared to cropping in input space using *two RGB crops*, or simply using the *default* cross-modal only loss. Compared to recent state-of-the-art cross-modal self-supervised learning methods (XDC [6], GDT [107], AVID-CMA [98], SeLaVi [9]) pre-trained on Kinetics-400 [69] STiCA is able to achieve significantly better results in fewer epochs.

1. Introduction

Visual representations have evolved significantly in the last two decades. The first generation of representations comprises algorithms such as SIFT [87] and HOG [30] that were designed manually. The second generation comprises representations learned from data by using deep neural networks and manual supervision [31, 59, 76]. We are now transitioning to the third generation, where representations are learned from data without using any manual annotations by means of self-supervision. Current self-supervised rep-

*Equal contribution.
[1]https://github.com/facebookresearch/GDT

resentations, obtained from methods such as MoCo [57], SimCLR [24] or SwAV [20], convincingly outperform supervised ones on downstream tasks such as image classification, segmentation and object detection. Furthermore, most of these methods are based on noise-contrastive instance discrimination, which was proposed in ExemplarCNNs [39] and put in its current form in [143] and [102]. The idea is to learn representations that are invariant to irrelevant factors of variations, modelled by strong augmentations such as image cropping, while remaining distinctive for the identity of the image.

Noise-contrastive learning is of course not limited to still images. In particular, a number of recent approaches [54, 94, 98, 107] have used noise-contrastive formulations to learn visual or audio-visual representations. However, these methods are not as well developed as their counterparts for still images, with current state-of-the-art methods [54, 107] still lagging behind their supervised counterparts.

In this paper, we identify two areas in which current video representation learning formulations are lacking and improve on them, thus significantly improving upon the current state of the art in this area.

The first shortcoming is the lack of a sufficient encoding of **spatial invariances**. For still images, learning spatial invariances has been shown to be one of the most important factors for performance [20, 24]. Almost all methods achieve some form of spatial invariance simply by applying different spatial augmentations to the images in different epochs of training. However, learning spatial invariances in this manner requires a slow training process that lasts for many epochs ($\sim$800). Authors have suggested that packing several augmentations of the same image in a single data batch is more effective as it provides a much stronger and more direct incentive for the network to learn invariances [20].

For videos, both strategies are less feasible. Training a model for 200 epochs on Kinetics-400 [69] already requires around 1.5K GPU hours on recent Nvidia V100 architectures, and with recent datasets such as IG65M [45] and HowTo100M [95] only a handful of epochs can realistically be completed. On the other hand, including multiple augmentations of the same video in a batch rapidly exhausts the memory of GPUs. Since batch sizes per GPU are already in the single digits due to the size of video data, including several augmentations is unfeasible. This is particular detrimental for recent contrastive learning approaches such as [24, 58], where reducing the batch size means reducing the pool of negative contrastive samples.

In order to solve this problem, we propose to move spatial augmentations to the feature space, in a manner specifically tailored to contrastive learning. Instead of extracting a large number R of different augmentations in the input RGB space, we extract only two of them, apply the trunk of the neural network to extract corresponding features, and then extract $R/2$ more augmentations directly in feature space. In this way, one needs to evaluate the slow and memory taxing feature extraction part of the network only twice, regardless of the number of augmentations that are produced. We show that this *feature-level augmentation* significantly improves representation learning performance.

The second challenge that we tackle is how to best encode **temporal information** in self-supervised video representation learning. Currently, most self-supervised video representation learning approaches use 3D-CNNs [21, 133, 134, 145] that compute convolutions across space and time, but the final representation is generated by naïve global average pooling over space and time, crucially discarding temporal ordering.

In order to address this shortcoming, in this work we propose to use a contextualized pooling function based on the transformer architecture [136] for both self-supervised pretraining and supervised finetuning. The intuition is that, via multi-head self-attention, the transformer can capture temporal dependencies much better than average pooling, especially for longer inputs. Transformers can also benefit from our feature-level crops, as the latter resemble the common approach of randomly masking the inputs to the transformer [62]. Experimental results show that this modification improves the performance of the learned video representations substantially, and is cumulative with the benefit of feature crops, at about the same cost of average pooling.

We combine both of our proposed improvements into a new self-supervised learning approach: Space-Time Attention and Cropping (**STiCA**). To summarize, with STiCA we make the following three main contributions:

- We demonstrate the benefits of stronger spatial invariances in self-supervised video representation learning for the first time and we propose feature-level augmentation to implement the latter efficiently.
- We propose to use transformers to model time more effectively in self-supervised video representations, replacing average as the pooling function.
- We demonstrate strong performance gains by using the two techniques and obtain state-of-the-art performance on two standard benchmarks (67.0% on HMDB-51 and 93.1% on UCF-101).

2. Related Works

Self-supervised Image Representation Learning. Self-supervised learning uses pretext tasks to automatically and easily generate differentiable learning signals from the data itself in order to train convolutional neural networks. A variety of pretext tasks have been proposed such as colorization [149, 150], predicting artificial rotations [46], in-

painting [106], spatial context [35, 101], and clustering features [11, 18, 19, 20, 64, 83]. Recently, contrastive methods [50, 51] have proven to be particularly effective at learning transferable image representations [13, 24, 49, 57, 96, 102, 131].

Self-supervised Video Representation Learning. For videos, pretext tasks often seek to leverage the temporal dimension to learn representations. Such tasks include predicting clip and sequence order [79, 97, 146], future events [52, 53], the arrow of time [141], 3D geometric transformations [65, 71], playback speed [14, 40, 63, 138], or motion statistics [137].

Multi-Modal Learning. The co-occurrence and synchronicity of multiple modalities from videos have been used to learn visual representations from both audio-video [6, 7, 9, 74, 91, 98, 103, 107], and speech-video [5, 73, 85, 93, 94, 99, 108, 124, 126, 127] data. Multi-modal representation learning has several practical applications: lip reading [3, 26, 27], audio-visual source separation and localization [2, 4, 8, 56, 151, 152], speech recognition [1, 112], efficient inference [43, 75], egocentric action recognition [70] and audio-visual navigation [22].

Data Augmentations. Data augmentation has proven to be useful in training deep learning models in many domains, from vision [28, 29, 147] to speech [104]. Data transformations are the foundation of most self-supervised works, and there has been early attempts to even learn the optimal distribution of transformations [16, 29]. Particularly for contrastive learning, the choice of data transformations has been shown to be particularly important to learn desirable invariances and equivariances [96, 107, 131, 132].

Transformations in Feature-Space. Some works have proposed forms of augmentation in feature-space, by adding noise and linear transformations [130], and by associating samples to prototypes in feature-space [78]. These augmentations do not correspond to interpretable geometric operations, however. Crops in feature-space are commonly used in supervised detection pipelines, such as Faster R-CNN and region-based architectures [116], and in earlier detectors based on manually-engineered features [30]. However, the objective of these transformations is to enumerate a space of outputs (e.g. bounding box predictions) for supervised prediction. In self-supervised learning, while [66] uses feature mixing to create harder negatives for contrastive learning, we are instead interested in using feature crop augmentation to achieve spatial invariance.

Temporal Modeling. Videos extend images by adding a temporal dimension. Therefore, there has been a large family of research that has looked into how to model temporal information in videos. Early works incorporated temporal information via average pooling of frame/clip-level features [48, 68, 139], while later work used 3D convolution neural networks [133, 134, 145] and recurrent-neural networks [37]. Other approaches leverage long-term temporal convolutions [135], self-attention [140], relation networks [153], multi-scale temporal convolutions [61], or optical flow in a two stream network [120].

Transformers in Vision. With the success of the transformer architecture [136] in natural language processing [62], transformers are being used in various vision domains such as image representation learning [23, 32, 38, 119, 142], image generation [105], object detection [17, 86], few-shot learning [36], video action recognition [15, 47, 100, 140], video question-answering [67], image-text [84, 88, 125, 128, 129] and video-text [42, 73, 108, 126, 127, 155] representation learning.

3. Method

Our goal is to learn a general-purpose *data representation* $\Phi : \mathcal{X} \to \mathcal{Z} = \mathbb{R}^D$ that maps data $\boldsymbol{x} \in \mathcal{X}$ to feature vectors $\boldsymbol{z} = \Phi(\boldsymbol{x})$. In the supervised setting, representations are learned end-to-end as components of larger systems that solve certain tasks of interest, such as image or video classification, under the assumption that supervision is available to drive the learning process. When supervision is not available, representations can still be learned via self-supervision by means of suitable pretext tasks. Among the latter, *noise contrastive learning* is one of the most popular and successful ones [24, 102]. We summarize this background next and discuss our extensions in the following sections.

3.1. Background: Multi-modal contrastive learning

The idea is to train the representation Φ to identify data points up to the addition of noise or, more generally, the application of certain nuisance transformations. To this end, let $g : \mathcal{X} \to \mathcal{X}$ be transformations sampled in a set $\mathcal{G}$ of possible nuisances (for example random image crops). Let $\mathrm{sim}(\boldsymbol{z}', \boldsymbol{z}'')$ be a similarity function comparing representations $\boldsymbol{z}'$ and $\boldsymbol{z}''$, such as the cosine similarity:

$$\mathrm{sim}(\boldsymbol{z}', \boldsymbol{z}'') = \frac{\langle \boldsymbol{z}', \boldsymbol{z}'' \rangle}{\|\boldsymbol{z}'\| \, \|\boldsymbol{z}''\|}.$$

Consider a dataset or batch $\mathcal{B} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$ of data samples. Slightly modifying [24], for each sample $\boldsymbol{x}_i$, draw a set of random nuisance transformations $\{g_{\alpha i}\}_{1 \leq i \leq N}$ and let $\boldsymbol{z}_{\alpha i} = \Phi(g_{\alpha i}(\boldsymbol{x}_i))$ be the representations of the transformed samples. Likewise, consider a second set β of transformations $\{g_{\beta i}\}_{1 \leq i \leq N}$. The noise contrastive loss (NCE) is given by:

$$\mathcal{L}(\alpha, \beta) = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{\frac{1}{\tau} \mathrm{sim}(\boldsymbol{z}_{\alpha i}, \boldsymbol{z}_{\beta i})}}{\sum_{j=1}^{N} e^{\frac{1}{\tau} \mathrm{sim}(\boldsymbol{z}_{\alpha i}, \boldsymbol{z}_{\beta j})}} \quad (1)$$

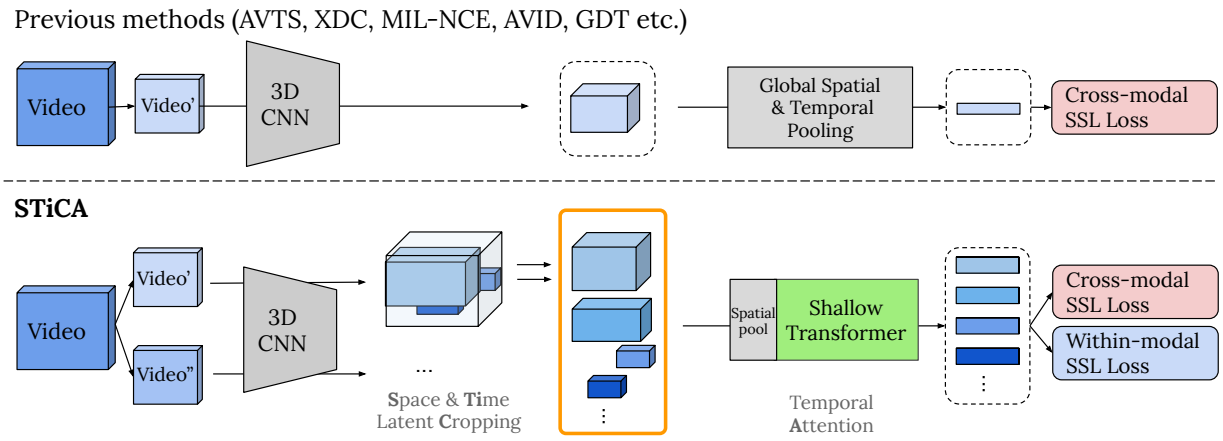


Figure 2: Approach Overview. We present a self-supervised approach that learns video representations without labels. **(Top)** Prior work in video representation learning did not capture spatial invariances, as taking many crops of the input (at varying locations and scales), quickly gets expensive in both compute and memory. **(Bottom)** The proposed method generates a large variety of views from only two RGB-crops by cropping in latent space and is particularly tailored to self-supervised contrastive learning. The latent crops are essentially masked features, which are then further processed by a light-weight temporal transformer. Compared to global pooling, this allows our method to further capture the rich temporal signal.

where $\tau > 0$ is a temperature parameter. This loss pulls together the representations of samples that only differ by the transformation while pushing apart the others. Note that this definition is not symmetric in the two arguments α and β (i.e., $\mathcal{L}(\alpha, \beta) \neq \mathcal{L}(\beta, \alpha)$. Note also that we can introduce any number of transformation sets $\alpha, \beta, \gamma, \ldots$ and, for each pair, we can obtain a different variant of eq. (1).

Recently, works such as [107] have ported this technique to the video domain by contrasting modalities. Each video $\boldsymbol{x} = (\boldsymbol{v}, \boldsymbol{a})$ consists of a visual component $\boldsymbol{v}$ and an audio component $\boldsymbol{a}$. One consider two sets of transformations g_v, extracting and augmenting the visual component, and g_a, extracting and augmenting the audio component. We still write $\Phi(g_\alpha(\boldsymbol{x}))$ for the feature computed for either visual and audio components, but the symbol means that a modality-specific neural network is applied as needed.[2]

With this, we can derive three variants of eq. (1), involving mixed visual-audio and homogeneous visual-visual and audio-audio comparisons. Their combinations are:

$$\lambda_{va}\mathcal{L}(v, a) + \lambda_{av}\mathcal{L}(a, v) + \lambda_{vv}\mathcal{L}(v_1, v_2) + \lambda_{aa}\mathcal{L}(a_1, a_2). \tag{2}$$

where λ_{va}, λ_{av}, λ_{vv} and λ_{aa} are non-negative mixing weights.

Challenge 1: Encoding within-modality invariance. While all terms in 2 code for desirable invariances of the representation, several recent papers [91, 98, 107] have found that the mixed term λ_{va} is far more important than the other two; in fact, performance *degrades* if one sets $\lambda_{aa}, \lambda_{vv} \neq 0$, meaning that within-modal invariances are not successfully leveraged. Our hypothesis is that within-modality invariance can be beneficial, and that these early negative results are due to the fact that current learning formulations are ineffective at capitalizing on this signal.

As suggested in Sec. 1, the fact that video data is large means that the batch size used in learning must be small. As a consequence, a batch can contain only a very small number of different augmentations of the same video sample. In current multi-modal learning formulations, each video is already transformed twice in order to extract video and audio components, so cross-modal invariance is learned well. However, the downside is that there is no space left in the batch for multiple visual or audio augmentations. Thus, within-modality invariance is learned only indirectly — in particular, as noted in Sec. 1, two different visual or audio augmentations of the same video are visited by the model only after an entire training epoch. Next, we address this issue by making it feasible to extract several within-modality transformations in the same batch even for video data.

3.2. Efficient spatial cropping for augmentation

It has been found that self-supervised learning benefits from, and requires more and stronger augmentations compared to the supervised counterpart for optimal performance [24]. In particular, several papers [10, 20, 24] have suggested that, in the case of still images, the most important type of augmentation is *cropping*. Namely, given an RGB image $\boldsymbol{x} \in \mathbb{R}^{3\times H\times W}$ with three channels and height and width H and W respectively, a crop is given by a box $B = (x_{\min}, x_{\max}, y_{\min}, y_{\max})$. The image tensor is first

[2] In other words, $\Phi = (\Phi_v, \Phi_a)$ is really a pair of networks, producing embedding vectors $\boldsymbol{z}_\alpha$ that are compatible regardless of the modality $\alpha \in \{v, a\}$.

cropped as:

$$C_B(\boldsymbol{x}) = \boldsymbol{x}_{[:,\ y_{\min}:y_{\max},\ x_{\min}:x_{\max}]} \tag{3}$$

where the : symbol is used to denote an index range. The cropped tensor is then resized to a tensor $\tilde{\boldsymbol{x}} = g(\boldsymbol{x}) = R_{H_0W_0}(C_B(\boldsymbol{x})) \in \mathbb{R}^{3\times H_0\times W_0}$ with a given height and width $H_0 \times W_0$. In practice, $R_{H_0W_0}$ may also apply additional augmentations such as color jittering, as detailed in the experiments.

As for the visual part $\boldsymbol{v} \in \mathbb{R}^{3\times T\times H\times W}$ of a video, the situation is similar, except that the video also contains an additional temporal dimension T. To avoid extreme spatial jittering and keep objects aligned, a spatial crop is usually taken at the same location in the input space throughout the whole temporal dimension, so we consider the tube $B = (x_{\min}, x_{\max}, y_{\min}, y_{\max}, t_{\min}, t_{\max})$ and define $\tilde{\boldsymbol{v}} = g_v(\boldsymbol{v}) = R_{H_0W_0}(C_B(\boldsymbol{v})) \in \mathbb{R}^{3\times T_0\times H_0\times W_0}$ by extending (3) in the obvious way.

The deep neural network $\boldsymbol{z} = \Phi(\tilde{\boldsymbol{v}})$ mapping $\boldsymbol{v}$ to its corresponding code $\boldsymbol{z}$ is fed with tensors with two spatial dimensions and a temporal one. Such networks, often called 3D for this reason, include R3D [55], S3D [145] and R(2+1)D [134]. As customary in deep convolutional neural networks, they first produce an intermediate tensor with lower space-time resolution and then pool the latter to obtain a single code vector for the entire video. We explicitly break this down into three functions

$$\Phi(\tilde{\boldsymbol{v}}) = (\mathcal{P}_t \circ \mathcal{P}_s \circ \Psi)(\tilde{\boldsymbol{v}}) \tag{4}$$

Here, the first function is a 3D convolutional neural network $\Psi(\tilde{\boldsymbol{v}}) \in \mathbb{R}^{D\times T_1\times H_1\times W_1}$ producing a tensor with reduced resolution $T_1 < T_0$, $H_1 < H_0$, $W_1 < W_0$. The operators $\mathcal{P}_s$ and $\mathcal{P}_t$ collapse, respectively, spatial and time dimensions via average pooling.

Now consider implementing term $\mathcal{L}(v_1, v_2)$ in 2. In this case, one samples from each video $\boldsymbol{x}_i$ two different space-time crops $g_{v_1 i}(\boldsymbol{x}_i)$ and $g_{v_2 i}(\boldsymbol{x}_i)$, each corresponding to random tubes B_1 and B_2 respectively. The tubes are not sampled entirely independently, however, as they have the same temporal extent $(t_{\min}, t_{\max})$.

Naïve multiple spatial cropping In practice, [20, 24, 82, 96] show that taking multiple image crops improves self-supervised image representations. We can achieve a similar effect for videos by summing losses $\mathcal{L}(v_\alpha, v_\beta)$ for sets of visual transformations $v_\alpha \neq v_\beta$, obtained by sampling multiple space-time tubes for each video, but this is practically difficult, both due to the large memory footprint and the compute overhead of the slow 3D CNN for each crop.

The Multi-Crop approach introduced by SwAV [20] in the image domain combined with our asymmetric contrastive formulation (1) can partially reduce the complexity. For Multi-Crop, we consider three crop sizes $\alpha \in \{L_1, L_2, S\}$ where L_1 and L_2 stands for large and S for small. The use of a small crop allows to reduce the memory consumption when the representation Φ is computed. We then have losses:

$$\mathcal{L}(v_{L_1}, v_{L_2}) + \mathcal{L}(v_{L_2}, v_{L_1}) + \mathcal{L}(v_{L_1}, v_S) + \mathcal{L}(v_{L_2}, v_S).$$

While operating on small videos saves some computation, in practice this approach is insufficient to allow using more than a handful of crops in total.

Efficient cropping in feature space. As illustrated in Fig. 2, a much more efficient alternative to cropping the input video is to crop intermediate features.

To do so, we first apply the trunk Ψ of the representation to an input-space crop of the visual component of the video $\tilde{\boldsymbol{v}} = R_{H_0W_0}(C_B(\boldsymbol{v})) \in \mathbb{R}^{D\times T_1\times H_1\times W_1}$. Then we can efficiently construct a new view of this data by applying the *Feature Crop* $C_{\bar{B}}$ directly on each intermediate representation, yielding

$$\bar{\boldsymbol{v}} = C_{\bar{B}}(\Psi(\tilde{\boldsymbol{v}})) = \Psi(\tilde{\boldsymbol{v}})_{[t_{\min}:t_{\max},\ y_{\min}:y_{\max},\ x_{\min}:x_{\max}]} \tag{5}$$

Since the operator $C_{\bar{B}}$ is lightweight, it can be used to compute several such random views efficiently; by comparison, cropping the input RGB images requires recomputing the trunk Ψ multiple times.

In practice, given an input video $\boldsymbol{v}$, we generate the following views. First, we apply two crops in RGB space, producing two large crops L_1 and L_2. Then, for each of those, we use the operator (5) to generate m medium-sized and n small-sized crops $\mathcal{T}_i = \{M_1 L_i, \ldots, M_m L_i, S_1 L_i, \ldots, S_n L_i\}$. We define an overall within-modality loss by summing losses for each pairs of views in $\mathcal{T}$ with exception of pairs where both crops are small:

$$L_{vv} = \sum_{\alpha,\beta} \mathcal{L}(v_\alpha, v_\beta) + \mathcal{L}(v_\beta, v_\alpha), \quad \text{where}$$
$$(\alpha, \beta) \in (\mathcal{T}_1 \times \mathcal{T}_2) - (\mathcal{S}_1 \times \mathcal{S}_2) \tag{6}$$

Note that there are $2((m+n)^2 - n^2)$ terms in this loss. This is a far greater number of comparison than afforded by the two initial input-space RGB crops.

Receptive Field of Feature Crops and Preventing Shortcut Learning. Noise contrastive learning works better when you can reduce the mutual information between the input pairs [132] as its harder for the network to cheat. This can be achieved by taking multiple spatial crops of images in the input space and independently applying different augmentations, such as color jittering and Gaussian blurring, to the cropped inputs. However, as mentioned above, taking more than 2 crops in input space is both memory and computationally infeasible for multi-modal video data. Crops in

feature space, on the other hand, allows us to take multiple crops for noise contrastive learning. However, since CNNs have large receptive fields that easily cover the full frame, there may be shortcut learning with feature crops as information may leak between the crops from same feature map. To alleviate this, we take feature crops from two originally augmented video clips, allowing us to make NCE comparisons *across* modalities and individual augmentations (such as color jitter), leading to a beneficial reduction in mutual information. Furthermore, while the theoretical receptive fields of units in later layers are indeed very large, units tend to be sensitive to an effective area which is significantly smaller than the theoretical receptive field [90, 154], further reducing the mutual information between inputs for noise contrastive learning.

3.3. Temporal modelling with transformers

We now discuss our second improvement: better modelling of time.

Challenge 2: Modelling time better. Contrary to spatial invariance, models should not be fully invariant to time as the latter can encode causality and with it semantics: a video of someone starting a fire is very different from its reversed version, in which someone extinguishes it. In standard 3D networks, features in the trunk are sensitive to the temporal order, but this information is lost in the final stage, where temporal averaging is applied. We argue that the value of the lost information increases with the length of the video, and that this information can be leveraged by switching to a different pooling function.

Temporal transformer. We propose to tackle this issue by replacing average pooling in time $\mathcal{P}_t$ in eq. (4) with a transformer $\mathcal{P}_{\text{transf}}$. Transformers [136] have been shown effective for representing sequential inputs in the NLP domain [62, 81, 114, 115].

After spatial averaging, the output $\boldsymbol{h} = \mathcal{P}_s(\Psi(\tilde{\boldsymbol{v}})) \in \mathbb{R}^{D\times T_1}$ of the network has one feature vector per time step, and is thus amenable to processing by a transformer. The feature $\boldsymbol{h}$, which differs in latent time-dimension size from its uncropped variant can be seen as masking the transformer's attention. Masking attention has been used in transformer encoder-decoder training to prevent the model from cheating [33] and encourage it to leverage information from the context.

We use a shallow and light-weight transformer on top of our feature cropping procedure, which we show to be sufficient to reap the benefit of better temporal modelling incurring only a very small computational cost. We use 2-layers and 4 self-attention heads and provide further details on the transformer architecture in the Appendix.

3.4. Overall loss

Our combined model, **STiCA**, better learns space-time invariances and relationships by cropping in space-time and leveraging temporal attention with a transformer. For training, we sample N videos in a batch and, for each of them, compute two 'large' visual crops in RGB space, $2(n+m)$ small and medium feature crops (Sec. 3.2), and an audio augmentation a. With those, the overall objective is obtained by summing the within-modality loss L_{vv} from eq. (6) to the cross modality losses:

$$L = \lambda_{vv} L_{vv} + \lambda_{va} L_{va}, \tag{7}$$

where $L_{va}{=}\mathcal{L}(v_{L_1}, a){+}\mathcal{L}(v_{L_2}, a){+}\mathcal{L}(a, v_{L_1}){+}\mathcal{L}(a, v_{L_2})$.

4. Experiments

We first describe the datasets (Sec. 4.1) and implementation details (Sec. 4.2) for pretraining. In Sec. 4.3, we describe the downstream tasks for evaluating the representation obtained from self-supervised learning. In Sec. 4.4, we ablate the various components of our method, and the importance of temporal context and multi-modality in Sec. 4.5. Lastly, in Sec. 4.6, we compare with prior work in video and multi-modal representation learning.

4.1. Data

We pretrain on the Kinetics-400 dataset [69], which contains about 230K training videos and 13K validation videos belonging to 400 action classes. This dataset is the "ImageNet" for video representation learning due to its moderate size and being public, allowing for broad access and comparability. After pretraining, we evaluate using video action retrieval and action recognition on HMDB-51 [77] and UCF-101 [121]. HMDB-51 [77] consists of 7K video clips spanning 51 different human activities. HMDB-51 has three train/test splits of size 5K/2K respectively. UCF-101 [121] contains 13K videos from 101 human action classes, and has three train/test splits of size 11K/2K respectively.

4.2. Implementation details

Following [107], we use the R(2+1)-18 [134] network as visual encoder and ResNet [59] with 9 layers as audio encoder. We train for 100 epochs and use 30 frames with temporal stride of 1 at sampling rate of 30fps at spatial resolution of 112×112 as input. In our ablations, we evaluate the learned representation by finetuning the visual encoder on fold 1 of the HMDB-51 [77] action recognition dataset. Further implementation details are given in the Appendix.

4.3. Downstream tasks

Video action retrieval. For video retrieval, we follow the standard protocol described in [146]. We use the split 1

Table 1: Comparison experiments and ablations. We compare key parameters and settings of our proposed method. We report results model performance at epoch 100 and with 30 frames and without transformer unless noted otherwise.

Cropping-strategy	Resolution	GPU-h/epoch	Acc.
Default	1×112^2	17.3	54.0
Two RGB Crops	2×112^2	29.3	58.6
Multi RGB Crops [20]	$2\times112^2 + 1\times96^2$	46.7	59.3
Ours (Feature Crop)	2×112^2 + latent	29.3	**60.4**

(a) Cropping yields benefits but requires more compute. Our feature crops are efficient *and* outperform [20]. Note that all models are trained for 100 epochs.

l-Spatial size		l-Temporal size		Acc.
M	S	M	S	
1×7^2		1×4		54.0
1×6^2	2×4^2	1×4		59.9
1×6^2	2×4^2	2×3	1×2	58.4
2×6^2	4×4^2	2×3	1×2	**60.4**

(b) Feature crops. Heavier augmentations in latent (l) space and time lead to better representations.

Pretraining	Finetuning	Acc
$\mathcal{P}_t$	$\mathcal{P}_t$	54.0
$\mathcal{P}_t$	$\mathcal{P}_{\text{transf}}$	54.6
$\mathcal{P}_{\text{transf}}$	$\mathcal{P}_t$	52.1
$\mathcal{P}_{\text{transf}}$	$\mathcal{P}_{\text{transf}}$	**60.3**

(c) Pooling. Compared to Average-Pooling ($\mathcal{P}_t$), Transformer-based pooling ($\mathcal{P}_{\text{transf}}$) gives stronger performance.

Transf.?	Layers	Params	GFLOPS	Acc.
✗	0	37.2M	77.7	54.0
✗	0	42.8M	80.0	57.3
✓	2	42.4M	77.8	**60.3**
✓	4	47.7M	77.8	58.3

(d) Architecture. Using up to two transformer layers gives gains, not due to more trainable parameters.

C_{space}	C_{time}	T?	Acc.
✗	✗	✗	54.0
✓	✗	✗	59.9
✓	✓	✗	60.4
✓	✓	✓	**62.0**

(e) Combined gains. Feature crop in space C_{space} and time C_{time} and transformer pooling (T) add cumulative benefits.

Method	**RGB-Crops**			**Multi-scale RGB-Crops**			**Feature Crops**	
	1x	2x	4x*	2x112 + 1x96	2x112 + 2x96	2x112 + 6x96*	(1x7, 1x4)	(2x6 + 4x4, 2x3 + 1x2)
GPU-h/epoch	17.3	29.3	60.0	46.7	53.3	100.7	29.3	30.0

(f) Speed. Input-crops are slow: $*$ methods require reducing batch sizes (see Appendix) as activations do not fit on GPU.

of UCF-101, and additionally HMDB-51. We uniformly sample 10 clips per video, max pool and then average the features after the last residual block for each clip per video. We use these averaged features from the validation set to query the videos in the training set. If the class of a retrieved video matches the class of query video, we count it as a match. We measure recall at k=1, 5, 20.

Video action recognition. As is standard in the literature, we evaluate our pretrained representations by finetuning our visual backbone on the video action recognition task on HMDB-51 and UCF-101 datasets. We closely follow the finetuning schedule of GDT [107]. During finetuning, we use SGD with initial learning rate 0.0025, which we gradually warm up to 0.02 in the first 2 epochs The weight decay is set to 0.005 and momentum to 0.9. We use a mini-batch size of 32 and train for 12 epochs with the learning rate multiplied by 0.05 at 6 and 10 epochs. For training, we randomly sample 1s clips per video, and during evaluation, we uniformly sample 10 clips from each video and apply 3-crop evaluation as in [41].

4.4. Comparison experiments and ablations

Cropping augmentation. In Tab. 1a, we ablate the importance of spatial augmentation in learning video representations. We compare our proposed Feature Crop augmentation, $C_{\bar{B}}$, to the recently proposed Multi-Crop augmentation strategy [20] and other baseline approaches. Multi-Crop has proven to be effective in image self-supervised learning because it forces the model to learn local-to-global associations, by explicitly enforcing invariance between features of large-crops and those of multiple small crops. While effective, it can be particularly computationally intensive, which, with our hardware, limits its use to only two large crops and one small crop when applied to video representation learning. Our proposed Feature Crop is not only more efficient, but outperforms Multi-Crop by 1.1% when the learned representations is applied to action classification in HMDB-51. By cropping in feature space, we achieve a similar effect but can increase the number of small crops from 1 to 6 without increasing compute time.

Feature crop parameters. In Tab. 1b, we study the parameters of our Feature Cropping approach. We find that even our basic variant, which does one medium 6×6 crop and two 4×4 small crops (by cropping a 7×7 tensor) increases performance by nearly 6%, which is a relative improvement of more than 10%. If we further increase the number of crops in time and space, the performance increases from 59.9% to 60.4%.

Pooling Function. In Tab. 1c, we test temporal aggregation. We find that using a shallow transformer significantly outperforms simple average pooling by more than 5%; however, transformer pooling must be used both for pre-training the representation and for finetuning it on the target dataset.

Transformer architecture. In Tab. 1d, we test variants of the transformer architecture, including ablating iit alto-

Frames		Accuracy	
Pretrain	Finetune	GAP	Transf.
30	30	54.0	**60.3**
60	60	62.4	**66.1**
90	90	58.0	**66.9**

Table 2: Temporal context. We report results with different number of frames on finetuning accuracy.

λ_{va}	λ_{vv}	F. Crop?	Acc.
0	1	No	43.3
1	0	No	54.0
0.5	0.5	No	58.6
0.5	0.5	Yes	**60.3**

Table 3: Loss. Combining within-modal and cross-modal loss with Feature-crops is key.

gether. We find that temporal modelling as measured by downstream performance peaks at two layers, likely due to optimization difficulties of deeper transformers with SGD. We also compare to a model with approximately the same number of parameters as our 2-layer transformer (achieved by increasing the networks' last block's hidden dimension to 640). We find that the transformer still yields gains of 3%, indicating that it not the number of parameters but the modelling of time that is crucial for strong performance.

Combining Feature Crops and Transformer Pooling. In Tab. 1e, we show that combining Feature Crops in space and time, and then adding transformer pooling yield additive gains, with the best result obtained by combining all effects (which corresponds to **STiCA**). This shows that space-time augmentations and transformer pooling are complementary.

Cropping efficiency. In Tab. 1f, we compare training times (normalized to GPUs×hours) for Kinetics-400 epochs for the various spatial crops considered. We make two observations: First, the compute cost of RGB crops scales proportionally to their number because a full forward pass is required for each crop. Second, using a larger number of RGB crops eventually requires to decrease the batch size, which increases significantly the training time. In contrast, the cost of Feature Crop remains roughly constant no matter the number of crops.

4.5. Temporal Context and Multi-modality

Length of temporal context. In Tab. 2, we show the importance of leveraging longer context to improve video self-supervised representation learning. Similar to the supervised regime [135, 140], we observe improved accuracy as we increase the number of frames used during pretraining and fine-tuning. More importantly, the transformer pooling layer is better able to exploit this additional context, outperforming average pooling by over 4% for all frame lengths. Notably, there is a drop in performance when using GAP for extremely long contexts (90 frames).

Loss. Lastly, in Tab. 3, we study the effect of combining multi-modal learning signals with our contributions. In the first row, we have the baseline of naively extending SimCLR [24] to the video domain, by learning invariances to spatial augmentations of two large-crops. Compared to this, the cross-modal baseline (row 2) already achieves gains of more than 10%. While adding a within-modal invariance adds another 4.6%, we find that the best performance is obtained with our feature crops, adding another 1.7% in performance and showing its unique potential to supplement cross-modal signals.

[3]Concurrent work.

Method	**Architecture**	**Dataset**	**Top-1 Acc%**	
			HMDB	UCF
Supervised	R(2+1)D-18	K-400	70.4	95.0
Multisensory [103]	R3D-18	K-400	-	82.1
SeLaVi [9]	R(2+1)D-18	K-400	47.1	83.1
TempTrans [63]	R3D-18	K-400	49.8	79.3
PEMT [80]	SlowFast	K-400	-	85.2
XDC [6]	R(2+1)D-18	K-400	52.6	86.2
MemDPC [53]	R-2D3D	K-400	54.5	86.1
AVSF [144]	AVSF	K-400	54.6	87.0
AVTS [74]	MC3-18	K-400	56.9	85.8
CPD [85]	R3D-50	K-400	57.7	88.7
AVID [98]	R(2+1)D-18	K-400	60.8	87.5
GDT [107]	R(2+1)D-18	K-400	60.0	89.3
ACC [91]	R3D-18	K-400	61.8	90.2
GLCM [92]	R3D-18	K-400	61.9	91.2
CoCLR [54]	S3D	K-400	62.9	90.6
CVLR [113][3]	R3D-50	K-400	66.7	92.2
Ours: STiCA	R(2+1)D-18	K-400	**67.0**	**93.1**
L3-Net [7]	VGG-16	AS	40.2	72.3
SeLaVi [9]	R(2+1)D-18	VGGS	53.1	87.7
Speech2Act [99]	S3D-G	Movie	58.1	–
DynamoNet [34]	ResNext101	Y8M	58.6	87.3
MIL-NCE [94]	S3D	HT	61.0	91.3
AVTS [74]	MC3-18	AS	61.6	89.0
AVID [98]	R(2+1)D-18	AS	64.7	91.5
Textual [124]	S3D-G	WVT-70M	65.3	90.3
GDT [107]	R(2+1)D-18	AS	66.1	92.5
ACC [91]	R(2+1)D-18	AS	67.2	93.5
ELo [111]	R(2+1)D-50	Y2M	67.4	93.8
XDC [6]	R(2+1)D-18	IG65M	68.9	**95.5**
GDT [107]	R(2+1)D-18	IG65M	72.8	95.2
MMV [5]	TSM-50x2	AS+HT	**75.0**	95.2

Table 4: Comparison to SoTA for action recognition. Dashed line indicates position of our Kinetics-400 model in comparison to models trained with many more videos. We follow standard evaluation protocol across 3-folds. For linear evaluation results see Tab. 7.

4.6. Comparison with the state of the art

Video Action Recognition. In Tab. 4, we evaluate our pretraining approach on the standard HMDB-51 and UCF-101 action recognition benchmarks after pretraining on the Kinetics-400 dataset. Firstly, we find our model outperforming the similar NCE-based GDT [107] model by 7.0% and 3.8% on HMDB-51 and UCF-101. We further sig-

	UCF			HMDB		
Recall @	1	5	20	1	5	20
VCOP [146]	14.1	30.3	51.1	7.6	22.9	48.8
VCP [89]	18.6	33.6	53.5	7.6	24.4	53.6
MemDPC [53]	20.2	40.4	64.7	7.7	25.7	57.7
VSP [25]	24.6	41.9	62.7	10.3	26.6	76.8
SeLaVi [9]	52.0	68.6	84.5	24.8	47.6	75.5
CoCLR [54]	55.9	70.8	82.5	26.1	45.8	69.7
GDT [107]	57.4	73.4	88.1	25.4	51.4	75.0
Ours: STiCA	**59.1**	**76.2**	**88.1**	**26.3**	**49.2**	**76.4**

Table 5: Comparison to SoTA for retrieval. Nearest neighbor action retrieval performance @$k = \{1, 5, 20\}$.

nificantly outperform the current state-of-the art methods CoCLR [54] by 4.1% and 2.5% and CVLR [113] by 2.6% and 1.0% on HMDB-51 and UCF-101, respectively. Even more impressively, our approach is able to out-perform most prior works that use AudioSet [44] pre-training, which is around $10\times$ larger than Kinetics-400. This shows how effective and data-efficient our approach is, significantly closing the gap to supervised learning.

Video Action Retrieval. Lastly, we directly evaluate the transfer-ability of our pretrained representations on action retrieval on UCF-101 and HMDB-51. Similarly to full fine-tuning setting, we outperform all prior works.

5. Conclusion

We have address two shortcomings of current self-supervised video representation learning: insufficient spatial invariance, especially compared to the image domain, and inadequate modelling of time. We have introduced STiCA, improving spatial invariance at very little cost by implementing cropping in feature space, and improving modelling of time via a shallow transformer. Our method brings self-supervised video representation learning one step closer to the supervised case, providing significant gains w.r.t. the state-of-the-art.

References

[1] Triantafyllos Afouras, Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Deep audio-visual speech recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 3

[2] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. The conversation: Deep audio-visual speech enhancement. *Interspeech*, 2018. 3

[3] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. Deep lip reading: A comparison of models and an online application. 2018. 3

[4] Triantafyllos Afouras, Andrew Owens, Joon Son Chung, and Andrew Zisserman. Self-supervised learning of audio-visual objects from video. In *ECCV*, 2020. 3

[5] Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *NeurIPS*, 2020. 3, 8, 14, 15

[6] Humam Alwassel, Bruno Korbar, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 1, 3, 8, 14, 15

[7] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 3, 8, 14

[8] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. 3

[9] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 1, 3, 8, 9

[10] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *ICLR*, 2020. 4

[11] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020. 3

[12] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016. 14

[13] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019. 3

[14] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, 2020. 3

[15] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?, 2021. 3

[16] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, 2018. 3

[17] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 3

[18] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 3

[19] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019. 3

[20] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2, 3, 4, 5, 7

[21] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2

[22] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman.

Soundspaces: Audio-visual navigation in 3d environments. In *ECCV*, 2020. 3

[23] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 3

[24] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3, 4, 5, 8, 14

[25] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692*, 2020. 9

[26] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. *CVPR*, 2017. 3

[27] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Workshop on Multi-view Lip-reading, ACCV*, 2016. 3

[28] E. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. In *CVPRW*, 2020. 3

[29] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2020. 3

[30] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 1, 3

[31] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[32] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations, 2020. 3

[33] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. 6

[34] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *ICCV*, 2019. 8

[35] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 3

[36] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020. 3

[37] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 3

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. 3

[39] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9), 2015. 2

[40] D. Epstein, Boyuan Chen, and Carl Vondrick. Oops! predicting unintentional action in video. In *CVPR*, 2020. 3

[41] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 7

[42] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 3

[43] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020. 3

[44] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *ICASSP*, 2017. 9

[45] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 2

[46] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018. 2

[47] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *CVPR*, 2019. 3

[48] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan C. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017. 3

[49] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 3

[50] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 3

[51] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 3

[52] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCVW*, 2019. 3

[53] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020. 3, 8, 9, 15

[54] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *NeurIPS*, 2020. 2, 8, 9, 15

[55] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *ICCVW*, 2017. 5

[56] David Harwath, Adria Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*, 2018. 3

[57] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual rep-

resentation learning. In *CVPR*, 2020. 2, 3

[58] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv.cs*, abs/1911.05722, 2019. 2

[59] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 1, 6

[60] Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. In *CVPR*, 2019. 14

[61] Noureldien Hussein, Efstratios Gavves, and Arnold W. M. Smeulders. Timeception for complex action recognition. In *CVPR*, 2019. 3

[62] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018. 2, 3, 6

[63] S. Jenni, Givi Meishvili, and P. Favaro. Learning video representations by transforming time. In *ECCV*, 2020. 3, 8

[64] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019. 3

[65] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 3, 15

[66] Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020. 3

[67] Yash Kant, Dhruv Batra, Peter Anderson, Alex Schwing, Devi Parikh, Jiasen Lu, and Harsh Agrawal. Spatially aware multimodal transformers for textvqa. In *ECCV*, 2020. 3

[68] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 3

[69] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 2, 6

[70] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 3

[71] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. 3

[72] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 15

[73] Bruno Korbar, Fabio Petroni, Rohit Girdhar, and Lorenzo Torresani. Video understanding as machine translation, 2020. 3

[74] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 3, 8, 14

[75] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019. 3

[76] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 1

[77] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 6, 14

[78] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *ECCV*, 2020. 3

[79] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 3

[80] Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. In *International Conference on Learning Representations*, 2021. 8

[81] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020. 6

[82] Hao Li, Xiaopeng Zhang, Ruoyu Sun, Hongkai Xiong, and Qi Tian. Center-wise local image mixture for contrastive representation learning, 2020. 5

[83] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020. 3

[84] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019. 3

[85] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination. *arXiv preprint arXiv:2001.05691*, 2020. 3, 8

[86] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020. 3

[87] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. ICCV*, 1999. 1

[88] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 3

[89] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *AAAI*, 2020. 9

[90] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks, 2017. 6

[91] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Learning audio-visual representations with active contrastive coding, 2020. 3, 4, 8, 14

[92] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Contrastive self-supervised learning of global-local audio-visual representations, 2021. 8

[93] Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from video and text via large-scale discriminative clustering. In *Proc. ICCV*, 2017. 3

[94] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan

Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 2, 3, 8, 15

[95] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019. 2

[96] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 3, 5

[97] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 3

[98] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. *arXiv preprint arXiv:2004.12943*, 2020. 1, 2, 3, 4, 8, 14

[99] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *CVPR*, 2020. 3, 8

[100] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network, 2021. 3

[101] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 3

[102] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2, 3

[103] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. 3, 8

[104] D. Park, William Chan, Y. Zhang, Chung-Cheng Chiu, Barret Zoph, E. D. Cubuk, and Quoc V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. In *INTERSPEECH*, 2019. 3

[105] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018. 3

[106] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 3

[107] Mandela Patrick, Yuki Markus Asano, Ruth Fong, João F. Henriques, G. Zweig, and A. Vedaldi. Multi-modal self-supervision from generalized data transformations. *ArXiv*, abs/2003.04298, 2020. 1, 2, 3, 4, 6, 7, 8, 9, 14

[108] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning, 2020. 3

[109] Karol J. Piczak. Environmental sound classification with convolutional neural networks. *MLSP*, 2015. 14

[110] Karol J. Piczak. Esc: Dataset for environmental sound classification. In *ACM Multimedia*, 2015. 14, 15

[111] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020. 8, 15

[112] Gerasimos Potamianos, Chalapathy Neti, Guillaume Gravier, Ashutosh Garg, and Andrew W Senior. Recent advances in the automatic recognition of audiovisual speech. *Proceedings of the IEEE*, 91(9):1306–1326, 2003. 3

[113] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. 2020. 8, 9

[114] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 6

[115] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 6

[116] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 3

[117] Guido Roma, Waldo Nogueira, and Perfecto Herrera. Recurrence quantification analysis features for environmental sound recognition. *WASPAA*, 2013. 14

[118] Hardik B. Sailor, Dharmesh M Agrawal, and Hemant A Patil. Unsupervised filterbank learning using convolutional restricted boltzmann machine for environmental sound classification. In *INTERSPEECH*, 2017. 14

[119] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020. 3

[120] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 3

[121] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 6

[122] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D. Plumbley. Detection and classification of acoustic scenes and events. *TM*, 2015. 14

[123] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 2015. 14, 15

[124] Jonathan C. Stroud, D. Ross, Chen Sun, Jun Deng, R. Sukthankar, and C. Schmid. Learning video representations from textual web supervision. *ArXiv*, abs/2007.14937, 2020. 3, 8

[125] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020. 3

[126] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019. 3, 15

[127] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 3

[128] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019. 3

[129] Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded

supervision. In *EMNLP*, 2020. 3

[130] V Terrance and W Taylor Graham. Dataset augmentation in feature space. In *Proceedings of the international conference on machine learning (ICML), workshop track*, 2017. 3

[131] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 3

[132] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning. In *NeurIPS*, 2020. 3, 5

[133] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2, 3

[134] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2, 3, 5, 6

[135] Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term Temporal Convolutions for Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 3, 8

[136] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3, 6

[137] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019. 3

[138] Jiangliu Wang, Jianbo Jiao, and Y. Liu. Self-supervised video representation learning by pace prediction. In *ECCV*, 2020. 3

[139] Limin Wang, Yuanjun Xiong, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 3

[140] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 3, 8

[141] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 3

[142] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020. 3

[143] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2

[144] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020. 8, 15

[145] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. In *ECCV*, 2018. 2, 3, 5

[146] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019. 3, 6, 9

[147] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3

[148] Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models?, 2020. 14

[149] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[150] Richard Zhang, Phillip Isola, and Alexei A Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, 2017. 2

[151] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, 2019. 3

[152] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018. 3

[153] Bolei Zhou, Alex Andonian, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 3

[154] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015. 6

[155] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020. 3

6. Appendix

6.1. Implementation Details

While videos in Kinetics are 10 seconds long, we randomly sample either 1-second (30 frames), or 2-second (60 frames) clips from the 30fps videos. For the R(2+1)-D-18 visual encoder, the dimensions of the $res5$ feature map before spatial pooling is 512 x T x 7 x 7 for a 112 x 112 resolution video, where $T = 4$ for 30-frame (1 second) input, and $T = 8$ for 60-frame (2 second) input. After spatial pooling, we use either average pooling or a transformer as the temporal pooling function for the visual encoder, but always use average pooling for the audio encoder. The transformer's layers dimensionality are set to 512-D. Both encoders produce a fixed-dimensional representation vectors after temporal aggregation (512-D). Both vectors are then passed through two fully-connected layers with intermediate size of 512 to produce 256-D embedding vectors $\boldsymbol{z}$ as in [107]. We use these embeddings in our loss eq. (7) and train our model for 100 epochs. For the visual component of the video, we use a 30 frame RGB clip as input, at 30 fps covering 1 second. The video clip has a spatial resolution of 112×112 pixels. For input data augmentation, we apply random crops, horizontal flips, Gaussian blur and color jittering, all clip-wise consistent, following the protocol of SimCLR [24], and we ablate multiple settings for spatial and temporal feature cropping sizes. For the audio input, we extract a 1-second log-mel spectrogram of dimension 257×199 starting at the same time as the visual component. We also apply volume jittering to increase the robustness of our audio features. We optimize this model using SGD with momentum 0.9, weight decay 10^{-5} and learning rate 0.64, with a warm-up period of 10 epochs. For NCE contrastive learning, the temperature τ is set as 0.1 for cross-modal loss, and 0.5 for the within-modal loss. We use a mini-batch size of 8 on each of our 64 GPUs giving an effective batch size of 512 for distributed training. In our ablations, we evaluate the learned representation by fine-tuning the visual encoder on fold 1 of the HMDB-51 [77] action recognition dataset.

6.1.1 State-of-the-Art Experiment Details

For our state-of-the-art model, we train for 100 epochs, using R(2+1)-D-18 visual encoder with transformer temporal attention pooling, and Resnet-9 for audio encoder. We use 60 frames as input, and feature-crop augmentation (space: $2\times6^2+4\times4^2$ & time: $2\times3+1\times2$).

6.2. Transformer Architecture Details

We use a 2-layer transformer, with 4 attention heads, and hidden dimension 512. The input to the transformer is the spatially averaged output of the last convolutional layer of R(2+1)D-18 video backbone. The transformer contextualizes features across time to output a fixed feature length representation of dimension 512, which is then passed to MLP head for contrastive learning. While transformers generally benefit from being optimized with Adam [148], we adhere to using SGD for simplicity. We also do not observe any stability issues, likely because the transformer is quite shallow.

Method	**Pretraining**	**Acc%**	
		DCASE	ESC50
Autoencoder [12]	-	-	39.9
Random Forest [110]	-	-	44.3
Piczak ConvNet [109]	-	-	64.5
RNH [117]	-	72	-
Ensemble [122]	-	77	-
ConvRBM [118]	-	-	86.5
AVTS [74]	K400	91	76.7
XDC [6]	K400	–	78.0
AVID [98]	K400	93	79.1
ACC [91]	K400	–	79.2
Ours: STiCA	K400	**94**	**81.1**
SoundNet [12]	SNet	88	74.2
L3-Net [7]	SNet	93	79.3
AVTS [74]	SNet	94	82.3
DMC [60]	SNet	–	82.6
AVTS [74]	AS	93	80.6
XDC [6]	AS	–	85.8
MMV [5]	AS	–	86.1
AVID [98]	AS	96	89.2
GDT [107]	AS	**98**	88.5
ACC [91]	AS	–	**90.8**
Human [110]	–	–	81.3

Table 6: Audio classification. Downstream task accuracies on standard audio classification benchmarks on DCASE2014 and ESC50. Dataset abbreviations **A**udio**S**et, **K**inetics**400**, **S**ound**Net**,

7. Additional experiments

7.1. Audio Classification

For completeness, we also present audio classification results on ESC-50 [108] and DCASE-2014 [123]. ESC-50 [110] is an environmental sound classification dataset which has 2K sound clips of 50 different audio classes. ESC-50 has 5 train/test splits of size 1.6K/400 respectively. DCASE2014 [123] is an acoustic scenes and event classification dataset which has 100 training and 100 testing sound clips spanning 10 different audio classes. We demon-

Method	Architecture	Dataset	Top-1 Acc%	
			HMDB	UCF
RotNet3D [65]	S3D	K600	24.8	47.7
CBT [126]	S3D+BERT	K600	29.5	54.0
MemDPC [53]	R-2D3D	K400	30.5	54.1
AVSF [144]	AVSF	K400	44.1	77.4
CoCLR [54]	S3D	K400	46.1	74.5
Ours: STiCA	R(2+1)D-18	K400	**48.2**	**77.0**
MIL-NCE [94]	S3D	HT	53.1	82.7
XDC [6]	R(2+1)D-18	IG65M	56.0	85.3
MMV [5]	R(2+1)D-18	AS	60.0	83.9
ELo [111]	R(2+1)D-50	Y8M	64.5	–

Table 7: Comparison to state-of-the-art. Transfer learning results on UCF-101 and HMDB-51 when video backbone is frozen.

strate competitive performance relative to the state-of-the-art, despite training on a much smaller and less audio-rich Kinetics-400 dataset. We extract 10 equally spaced 2-second sub-clips from each full audio sample of ESC-50 [110] and 60 1-second sub-clips from each full sample of DCASE2014 [123]. We save the activations that result from the audio encoder to quickly train the linear classifiers. We use activations after the last convolutional layer of the ResNet-9 and apply a max pooling with kernelsize (1,3) and stride of (1,2) without padding to the output. For both datasets, we then optimize a L2 regularized linear layer with batch size 512 using the Adam optimizer [72] with learning rate $1\text{x}10^{-4}$, weight-decay set to $5\text{x}10^{-4}$ and the default parameters. The classification score for each audio sample is computed by averaging the sub-clip scores in the sample, and then predicting the class with the highest score. The mean top-1 accuracy is then taken across all audio clips and averaged across all official folds.

7.2. Linear probing results

In Tab. 7, we compute the linear classification results of our model compared to other recent methods. We find that our best model has competitive 3-fold linear evaluation results of 48.2% on HMDB-51 and 77.0% on UCF-101.

7.3. Supervised training on K-400

Here we experiment with training supervisedly on Kinetics-400 and observing the effect of using feature cropping (with the configuration 2 medium and 2 small latent space crops). The experimental results are given in Tab. 8 We find that even though our method is designed for contrastive cross-modal pretraining, using feature crops can help in training in a supervised manner too.

Fm-Crop	HMDB-51 Top-1 Acc.
✗	67.6
✓	69.0

Table 8: Supervised Training. We train the R(2+1)D+Transformer architecture supervisedly on Kinetics-400 with and without feature crops enabled.

4

Support-set bottlenecks for video-text representation learning

This work was presented at the International Conference on Learning Representations (ICLR) 2021 as a spotlight.

SUPPORT-SET BOTTLENECKS FOR VIDEO-TEXT REPRESENTATION LEARNING

Mandela Patrick[*], **Po-Yao Huang**[*], **Florian Metze & Andrea Vedaldi**
Facebook AI
{mandelapatrick,berniehuang,fmetze,vedaldi}@fb.com

Alexander Hauptmann
Language Technologies Institute
Carnegie Mellon University
alex@cs.cmu.edu

Yuki M. Asano[*]**& João Henriques**
Visual Geometry Group
University of Oxford
{yuki,joao}@robots.ox.ac.uk

ABSTRACT

The dominant paradigm for learning video-text representations – noise contrastive learning – increases the similarity of the representations of pairs of samples that are known to be related, such as text and video from the same sample, and pushes away the representations of all other pairs. We posit that this last behaviour is too strict, enforcing dissimilar representations even for samples that are semantically-related – for example, visually similar videos or ones that share the same depicted action. In this paper, we propose a novel method that alleviates this by leveraging a generative model to naturally push these related samples together: each sample's caption must be reconstructed as a weighted combination of other support samples' visual representations. This simple idea ensures that representations are not overly-specialized to individual samples, are reusable across the dataset, and results in representations that explicitly encode semantics shared between samples, unlike noise contrastive learning. Our proposed method outperforms others by a large margin on MSR-VTT, VATEX, ActivityNet, and MSVD for video-to-text and text-to-video retrieval.

1 INTRODUCTION

Noise contrastive learning (Gutmann & Hyvärinen, 2010) is emerging as one of the best approaches to learn data representations both for supervised (Khosla et al., 2020) and unsupervised regimes (Chen et al., 2020c). The idea is to learn a representation that discriminates any two data samples while being invariant to certain data transformations. For example, one might learn a representation that identifies a specific image up to arbitrary rotations (Misra & van der Maaten, 2020). In a multi-modal setting, the transformations can separate different modalities, for example, by extracting the audio and visual signals from a video. The resulting noise contrastive representation associates audio and visual signals that come from the same source video, differentiating others (Patrick et al., 2020).

The noise contrastive approach is motivated by the fact that the transformations that are applied to the data samples leave their 'meaning' unchanged. For example, rotating an image does not change the fact that it contains a cat or not (Gidaris et al., 2018). However, in most cases, we expect to find many data samples that share the same content without being necessarily related by simple transformations (e.g. think of any two images of cats). Existing noise contrastive formulations are unaware of these relationships and still try to assign different representations to these samples (Wu et al., 2018), despite the fact that they are semantically equivalent. If the representation is learned for a downstream task such as semantic video retrieval, this might degrade performance.

This suggest that there might be other learning signals that could complement and improve pure contrastive formulations. In this paper, we explore this idea in the case of learning from two modali-

[*]Joint first authors.

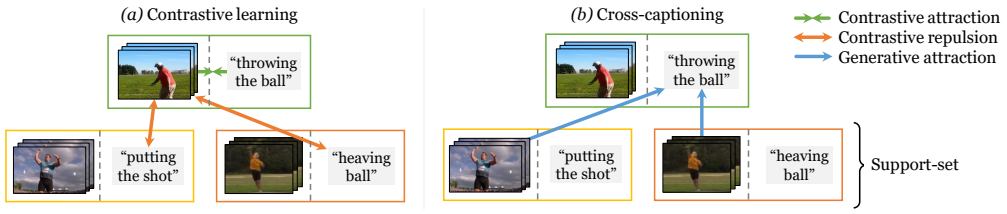


Fig. 1: **Cross-modal discrimination and cross-captioning.** Our model learns from two complementary losses: (a) Cross-modal contrastive learning learns strong joint video-text embeddings, but every other sample is considered a negative, pushing away even semantically related captions (orange arrows). (b) We introduce a generative task of cross-captioning, which alleviates this by learning to reconstruct a sample's text representation as a weighted combination of a support-set, composed of video representations from other samples.

ties: videos and text, in the form of video transcripts or captions. Given a state-of-the-art contrastive formulation that learns from these two modalities, we investigate complementary pretext objectives to improve it. First, we consider the *(instance) captioning* task, namely mapping a video to the corresponding text, casting this as a conditional stochastic text generation problem. We show that this brings only a modest benefit.

We observe that the captioning task is highly sample-specific, as the goal is to produce a caption which describes a specific video and not any other video, and thus it suffers from the same disadvantages (discouraging concept sharing among samples) as contrastive learning. Thus, we propose to address this issue by switching to a different text generation task. The idea is to modify the text generator to take as input a learnable mixture of a support-set of videos, which we call *cross-instance captioning*. The mixture weights are generated by comparing the learned video representations to captions' representations in an online way over the batch. The limited set of support samples acts as a bottleneck that encourages extraction of shared semantics. In this manner, the embeddings can associate videos that share similar captions even if the contrastive loss tries to push them apart.

We show that, when the captioning task is added in this manner, it brings a sensible improvement to already very strong video representation learning results, further improving our own state-of-the-art baseline by a significant margin.

2 Related Works

Learning data representations from unlabelled data has been a long standing goal of machine learning. These approaches are called "self-supervised learning" because the learning signals, termed pretext tasks, are obtained from the data itself. In the image and video domain, pretext tasks include colorization (Zhang et al., 2016), rotation (Gidaris et al., 2018), or clustering (Asano et al., 2020a;b; Caron et al., 2018; Ji et al., 2018), while in the natural language domain, masked language modeling (Devlin et al., 2019), and next word prediction (Mikolov et al., 2013; Pennington et al., 2014) are extremely popular. These pretext tasks can be broadly classified into two classes: generative and discriminative.

Discriminative approaches learn representations by differentiating input samples, using objectives such as the contrastive loss (Gutmann & Hyvärinen, 2010; Hadsell et al., 2006). Discriminative approaches have proven to be particularly successful for image (Chen et al., 2020c; He et al., 2020; Misra & van der Maaten, 2020; Wu et al., 2018) and video (Han et al., 2019; Morgado et al., 2020; Patrick et al., 2020) representation learning. Generative approaches, on the other hand, try to reconstruct its input. GANs (Donahue & Simonyan, 2019; Goodfellow et al., 2014; Radford et al., 2015), autoencoders (Hinton & Salakhutdinov, 2006) and sequence-to-sequence models (Huang et al., 2020; Sutskever et al., 2014) are popular generative models. In this work, we show the importance of combining both discriminative and generative objectives to learn effective video-text representations.

The success of representation learning has also been due to advances in model architectures, such as the Transformer (Vaswani et al., 2017). BERT (Devlin et al., 2019) demonstrated that a transformer

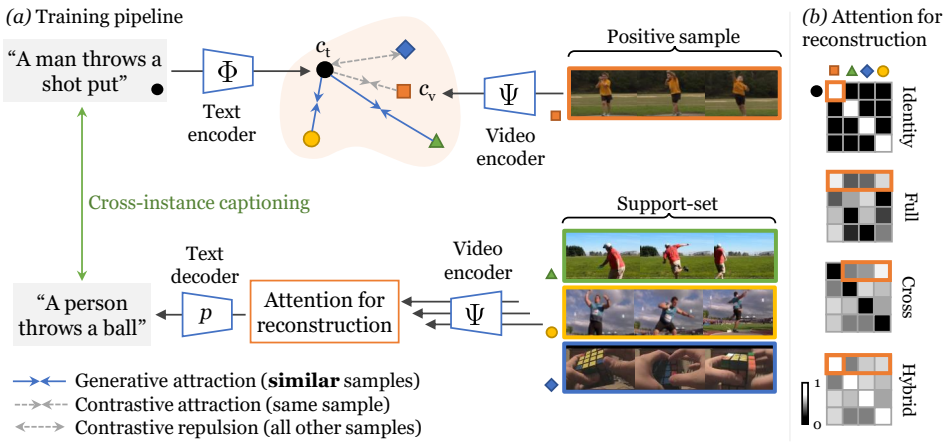


Fig. 2: **(a)** Our cross-modal framework with the discriminative (contrastive) objective and the generative objective. The model learns to associate video-text pairs in a common embedding space with text and video encoders (top). Meanwhile, the text must also be reconstructed as a weighted combination of video embeddings from a support-set (bottom), selected via attention, which enforces representation sharing between different samples. **(b)** Weights matrices (attention maps) used in each cross-captioning objective (see section 3.1.2).

architecture pretrained on large-scale textual data can learn transferable text representations that can be fine-tuned on a variety of downstream tasks. Subsequent works (Clark et al., 2020; Lewis et al., 2020a;b; Radford et al., 2019; Raffel et al., 2019) have improved upon the transformer architecture or training objective to learn even better representations. Inspired by the success of transformers in the NLP domain, several works have leveraged transformers to learn transferable image (Chen et al., 2020a; Desai & Johnson, 2020; Sariyildiz et al., 2020) or multi-modal image-text (Chen et al., 2019; Li et al., 2020a; 2019; Lu et al., 2019; Su et al., 2019; Tan & Bansal, 2019) and video-multilingual text (Huang et al., 2021) representations. In this work, we leverage the transformer architecture to better encode and represent text and video.

Large-scale training data has enabled the more effective pretraining of image (Sun et al., 2017; Yalniz et al., 2019), video (Ghadiyaram et al., 2019; Thomee et al., 2016) and textual representations (Raffel et al., 2019). The release of the HowTo100M dataset (Miech et al., 2019), a large-scale instructional video dataset, has spurred significant interest in leveraging large-scale pretraining to improve video-text representations for tasks such as video question-answering (Lei et al., 2018), text-video retrieval (Liu et al., 2019) and video captioning (Zhou et al., 2018b) on smaller datasets such as YouCookII (Zhou et al., 2018a), MSVD (Venugopalan et al., 2015a), MSR-VTT (Xu et al., 2016), LSMDC (Rohrbach et al., 2017), DiDeMo (Hendricks et al., 2018) and ActivityNet (Krishna et al., 2017). Although semantically rich and diverse, instructional videos from the web are super noisy and therefore a few approaches have been proposed to combat this. A few works (Luo et al., 2020; Sun et al., 2019a;b; Zhu & Yang, 2020) extend the BERT model to accept both visual and textual tokens to learn high-level semantic video-text representations. Other works have leveraged the contrastive loss (Miech et al., 2020) and show that using the raw audio (Alayrac et al., 2020; Rouditchenko et al., 2020) and other modalities (Gabeur et al., 2020) can be used to better align and improve video-text representations. While all these approaches rely on a contrastive objective, VidTranslate (Korbar et al., 2020) shows that a generative objective can also be used to learn joint video-text representations. In contrast to Korbar et al. (2020), we show that combining contrastive and generative objectives to pre-train video-text representations on large-scale data such as HowTo100M is very effective. The generative objective serves as regularizer to mitigate the strictness of the instance discrimination task of the constrastive objective, showing benefits similar to approaches such as clustering (Caron et al., 2020; Li et al., 2020b) and feature mixing (Kalantidis et al., 2020) which have been applied in the image domain.

3 METHOD

We consider the problem of learning multimodal representations from a corpus $\mathcal{C}$ of video-text pairs (v, t), where v is a video and t is its corresponding text (caption or transcription). Our goal is to learn a pair of representation maps $c_v = \Psi(v)$ and $c_t = \Phi(t)$, with outputs in a d-dimensional embedding space $c_v, c_t \in \mathbb{R}^d$, where semantically similar instances are close to each other.

3.1 OBJECTIVE FOR LEARNING MULTIMODAL REPRESENTATIONS

We consider two learning objectives, also illustrated in Figure 1. The first is the contrastive objective, pushing embeddings c_t and c_v to be close if text t and video v come from the same sample and pushing them apart otherwise. This assumes that every sample is its own class and does not benefit from modelling similiarities *across* instances. The second objective is generative captioning. In its most basic variant, it maximizes the probability of generating the text t given the corresponding video v. However, we suggest that variants that explicitly promote concept sharing between instances will result in better downstream performance, in tasks such as video retrieval. These variants, illustrated in Figure 2, have in common that the caption t is reconstructed from a learned weighted combination over *other* videos $\hat{v}$. This is a form of attention (Bahdanau et al., 2014) which encourages the network to learn about which videos share similar semantics, compensating for the contrastive loss and grouping them implicitly.

In the following, we denote with $\mathcal{B} \subset \mathcal{C}$ a *batch* of multi-modal samples, i.e. a finite collection of video-text pairs $(t, v) \in \mathcal{C}$. For simplicity, we denote the batch as $\mathcal{B} = \{(t^i, v^i)\}_{i=1}^B\}$.

3.1.1 CONTRASTIVE OBJECTIVE

To define the contrastive objective, let $s(a, b) = \frac{a^\top b}{\|a\|\|b\|}$ be the similarity measure between vectors a and b. Following Faghri et al. (2018), we adopt the hinge-based triplet ranking loss with hard negative mining:

$$\mathcal{L}^{\text{contrast}} = \frac{1}{B}\sum_{i=1}^{B}\left[\max_j \left[\alpha - s(c_t^i, c_v^i) + s(c_t^i, c_v^j)\right]_+ + \max_j \left[\alpha - s(c_t^i, c_v^i) + s(c_t^j, c_v^i)\right]_+\right], \quad (1)$$

where α is the correlation margin between positive and negative pairs and $[\cdot]_+ = \max\{0, \cdot\}$ is the hinge function. In our experiments, we set $\alpha = 0.2$.

3.1.2 CROSS-CAPTIONING OBJECTIVES

In the conventional captioning, the decoder seeks to optimize the negative log-likelihood of a text sequence t given its corresponding video v:

$$\mathcal{L}^{\text{caption}} = -\frac{1}{B}\sum_{i=1}^{B}\log p(t^i|e_v^i). \quad (2)$$

Here, the log-likelihood is obtained via auto-regressive decoding (Vaswani et al., 2017) from an intermediate video embedding $e_v^i = \Phi'(v^i)$. For the cross-captioning objective, we modify this loss to condition the generation process on a weighted average of the embeddings of the *other* videos in the batch, which we call the *support-set*. The weights themselves, which can be interpreted as a batch-wise attention, are obtained as a softmax distribution with temperature T over batch indices based on the video embeddings, as follows:

$$\mathcal{L}^{\text{cross-captioning}} = -\frac{1}{B}\sum_{i=1}^{B}\log p(t^i|\bar{e}_v^i), \quad \bar{e}_v^i = \sum_{j\in\mathcal{S}_i}\frac{\exp\langle c_t^i, c_v^j\rangle/T}{\sum_{k\in\mathcal{S}_i}\exp\langle c_t^i, c_v^k\rangle/T}\cdot e_v^j. \quad (3)$$

By default, the summation in the softmax is conducted over a support set $\mathcal{S}_i$ containing all indices except i. In the experiments, we consider the following attention types for reconstruction. **Identity captioning** ($\mathcal{S}_i = \{i\}$) generates the caption from the corresponding video and reduces to the standard captioning objective, eq. (2). **Full support** ($\mathcal{S}_i = \{1, \ldots, B\}$) considers all videos as possible candidates for captioning. **Hybrid captioning** sets the weights in eq. (3) as the average of the

weights for identity captioning and full support. **Cross-captioning** ($\mathcal{S}_i = \{j \neq i\}$) considers all *but* the video that one wishes to caption. This variant forces the network to extract all information required for captioning from other videos in the batch. Figure 2 compares graphically these attention mechanisms.

Considering both discriminative and generative objectives for learning multimodal representations, our full objective is $\mathcal{L} = \mathcal{L}^{\text{contrast}} + \lambda \mathcal{L}^{\text{cross-captioning}}$, where λ balances two objectives. We set $\lambda = 10$ to ensure similar magnitudes for both losses in our experiments. In the training phase, we use Adam (Kingma & Ba, 2015) to minimize our loss. At inference time, we directly use $\Phi(t)$ and $\Psi(v)$ to encode video and text representations for retrieval.

3.2 Model Architecture

We now discuss the details of the encoders and decoder components in our architecture, illustrated in fig. 2. For the *text decoder* $p(t|e_v)$ in eq. (2) and (3), we use a pre-trained T-5 decoder (Raffel et al., 2019).

For the *video representation* $c_v = \Psi(v) = \Psi''(\Psi'(v))$, we use a video encoder $e_v = \Psi'(v)$ followed by a multi-layer transformer pooling head $c_v = \Psi''(e_v)$. The encoder $\Psi'(v)$ concatenates the output of pretrained ResNet-152 (He et al., 2016) and R(2+1)D-34 (Tran et al., 2018) networks applied to individual video frames, resulting in a code $e_v = [e_{v1} \cdots e_{vM}]$ where M is the maximum duration of a video clip. For the pooling head $c_v = \Psi''(e_v)$, we consider a transformer architecture to attend to important context and summarize it into a fixed-length representation c_v. For this, we follow MMT (Gabeur et al., 2020), but with two important differences. First, while MMT uses 7 expert features that results in $7\times$ the sequence length, we only use a transformer to attend to early-fused motion and appearance features as the video representation, thus significantly reducing the sequence length and computational cost. Second, instead of stacking 6 transformer layers to encode the visual stream as in MMT, we only use a shallow two-layer transformer architecture with additional pre-encoders, further increasing model efficiency. As temporal 1D-convolutional neural networks (CNNs) (LeCun et al., 1998) were shown to effectively capture temporal dependencies in videos (Dong et al., 2019), we integrate CNNs into our transformer pooling heads to better capture video temporal signals. In more detail, we compute $c_v = \Psi''(e_v)$ by chaining two transformer layers, each of the type:

$$\psi(e) = \text{BN}(\text{FFN}(e_{\text{attn}}) + e_{\text{attn}}), \;\; e_{\text{attn}} = \text{BN}(\text{MHA}(f(e)) + f(e)). \tag{4}$$

Here f is a pre-encoder that refines the video representation; we found empirically that a 1D CNN works well for this purpose. Then, we apply multi-head self-attention (MHA) (Huang et al., 2019; Vaswani et al., 2017) followed by a feed-forward network (FFN) with batch normalization (BN) (Ioffe & Szegedy, 2015). The architecture maps the input sequence e_v to a new 'contextualized' sequence of representation vectors; we take the first one as c_v.

The text representation decomposes in the same way as $c_t = \Phi(t) = \Phi''(\Phi'(t))$. The text encoder $e_t = \Phi'(t)$ uses a pretrained T-5 network resulting in a code $e_t = [e_{t1} \cdots e_{tN}]$, where N is the maximum length of a sentence. The pooling head $c_t = \Phi''(e_t)$ follows the same design as the video case, but f is set to a recurrent neural network (RNN) instead of a CNN. Please refer to the appendix for details.

In practice, for computational reasons, we use eq. (3) to finetune the parameters of all networks except the video encoder $\Psi'(v)$, which is fixed.

4 Experiments

We validate empirically the ability of our method to learn better representations for the downstream tasks of text-to-video and video-to-text retrieval. First, in sec. 4.2 we ablate various model components on the MSR-VTT dataset. Then, in sec. 4.3 we show that our best model significantly outperforms state-of-the-art retrieval systems on three datasets, MSR-VTT, ActivtyNet and VATEX. Finally, in sec. 4.4 we analyse qualitatively the effect of the attention mechanism used during training.

Table 2: **Model Architecture and Training Details Ablation.** Text→Video retrieval performance on MSR-VTT. Recall@$1, 5$, and Median Recall are shown.

(a) **Video Encoder**. Stronger features and combination improves performance.

Feature source	$R@1\uparrow$	$R@5\uparrow$	$MdR\downarrow$
R-152	20.8	46.2	6.0
R(2+1)D-34	23.7	53.2	4.0
R(2+1)D-34 +R-152	**27.2**	**55.2**	**3.0**

(b) **Feature Aggregation.** Learning temporal attention yields strong gains over pooling.

Temporal reduction	$R@1\uparrow$	$R@5\uparrow$	$MdR\downarrow$
Max	21.8	49.5	8.0
Mean	22.5	51.3	6.0
Multi-Head Attn	**27.2**	**55.2**	**3.0**

(c) **Text Encoder.** Stronger encoding of text improves retrieval.

Text Encoder	$R@1\uparrow$	$R@5\uparrow$	$MdR\downarrow$
W2V (GloVe)	22.1	49.8	6.0
T5-Small	24.5	51.2	3.0
T5-Base	**27.2**	**55.2**	**3.0**

(d) **Text Decoder.** Stronger decoding of text improves retrieval.

Text Encoder	Text Decoder	$R@1\uparrow$	$R@5\uparrow$	$MdR\downarrow$
T5-Base	T5-Small	26.2	54.2	3.0
T5-Base	T5-Base	**27.2**	**55.2**	**3.0**

(e) **Contrastive Loss.** Inter-modal Triplet loss yields the best performance.

Contrastive	$R@1\uparrow$	$R@5\uparrow$	$MdR\downarrow$
InfoNCE (inter+intra)	10.7	28.5	15.0
InfoNCE (inter)	10.8	29.0	14.5
Triplet (inter+intra)	26.8	**56.2**	3.0
Triplet (inter)	**27.2**	55.2	**3.0**

(f) **Support-set Size.** Retrieval degrades when reconstructing from too small and too large sets.

	Batch-size							**Memory bank**	
Size	8	16	32	64	128	256	512	2k	8k
R@1/5	18.5/45.6	20.7/49.9	25.2/54.6	27.2/55.2	**28.0/56.1**	26.9/55.0	25.3/53.5	26.8/54.7	26.2/52.7

4.1 Experimental Setup

Datasets. **HowTo100M** (Miech et al., 2019) is a large-scale instructional video collection of 1.2 million YouTube videos, along with automatic speech recognition transcripts. We use this dataset for our pre-training experiments. **MSR-VTT** (Xu et al., 2016) contains 10,000 videos, where each video is annotated with 20 descriptions. We report results on the 1k-A split (9,000 training, 1,000 testing) as in Liu et al. (2019). **VATEX** (Wang et al., 2019) is a multilingual (Chinese and English) video-text dataset with 34,911 videos. We use the official training split with 25,991 videos and report on the validation split as in HGR (Chen et al., 2020b). The **ActivityNet Caption** (Krishna et al., 2017) dataset consists of densely annotated temporal segments of 20K YouTube videos. We use the 10K training split to train from scratch/ finetune the model and report the performance on the 5K 'val1' split. The **MSVD** (Chen & Dolan, 2011) dataset consists of $80K$ English descriptions for 1,970 videos from YouTube, with each video associated with around 40 sentences each. We use the standard split of 1,200, 100, and 670 videos for training, validation, and testing (Liu et al., 2019; Venugopalan et al., 2015b; Xu et al., 2015).

Evaluation Metrics. To measure the text-to-video and video-to-text retrieval performance, we choose Recall at K (R@K) and Median Rank (MedR), which are common metrics in information retrieval.

Table 1: **Effect of learning objectives.** Text→Video retrieval on MSR-VTT.

	$R@1\uparrow$	$R@5\uparrow$	Md$R\downarrow$
None	25.9	53.0	4.0
Identity	26.4	51.9	4.0
Full	25.8	53.9	3.0
Hybrid	26.0	54.8	3.0
Cross	**27.2**	**55.2**	**3.0**

4.2 Ablations

In Tab. 2, we first only ablate the cross-modal retrieval part of our network architecture, while the generative objectives are analysed in Tab. 1.

Video Encoder. In Tab. 2a, we show the effect of the choice of visual input features. We find that for text-to-video retrieval at Recall at 1 and 5 ($R@1, R@5$), features obtained from a video R(2+1)D-34 ResNet achieve 2.9% and 7.0% higher performance compared to only image-frame based features from a ResNet-152. A further 3.5% and 2.0% can be gained by concatenating both features, yielding the strongest MdR of 3.0%.

Feature Aggregation. While the features from both video and image-based visual encoders have reduced spatial extent after a fully-connected layer, the temporal dimension can be reduced in various ways. In Tab. 2b, we find that our multi-head, parameterized attention reduction yields strong gains over the mean- or max-pooling baselines of over 4% for $R@1$. This shows that learning attention over the temporal dimension of fixed feature sets can give strong gains even without fine-tuning the encoder.

Text Encoder. In Tab. 2c, we find decent gains of 2.7% and 0.4% for R@1,5 for using T5-base, instead of T5-small. We do not use the T-5-Large model, as in Korbar et al. (2020), due to the prohibitively large relative model size increase of +220%.

Text Decoder. In Tab. 2d, we find that using a larger text decoder gives a 1% increase in performance when using the cross-captioning objective.

Contrastive Loss. To validate the choice of a triplet loss in eq. (1), in Tab. 2e, we compare the results of the InfoNCE contrastive loss (Oord et al., 2018) with a triplet loss, with both the intra and inter-intra modality variants. We find that InfoNCE (Oord et al., 2018) loss does not work well in our case, likely due to the difficulty in tuning this loss to have the right combination of temperature and batch-size.

Support-Set Size. Lastly, in Tab. 2f, we show the effect of the size of the support set used for cross-instance captioning. We find that our reconstruction loss indeed acts as a bottleneck, with both smaller and very large sizes degrading the performance.

Captioning Objective. In Tab. 1, we show the effect of the different variants of our learning objective eq. (3). First, we find that the naive addition of a reconstruction objective ("Identity") does not improve the contrastive-only baseline ("None") much. Considering reconstruction from other videos improves the performance more. In particular, the "Hybrid" variant, which combines "Identity" and "Full" (sec. 3.1.2) improves Recall at 1 and 5 from 25.9% and 53.0% to 26.0% and 54.8%, respectively. However, the best result by far ($27.2/55.2\%$) is obtained forcing captions to be reconstructed only from *other* videos, via our cross-instance attention mechanism ("Cross"). This variant cannot use information contained in a video to generate the corresponding caption and thus entirely relies on the model to discover meaningful relationship between different videos. This newly-proposed scheme seems to have the most beneficial effect for semantic retrieval.

Table 3: **Retrieval performance on the MSR-VTT dataset.** Models in the second group are additionally pretrained on HowTo100M.

	Text →Video				**Video →Text**			
	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$
Random Baseline	0.1	0.5	1.0	500.0	0.1	0.5	1.0	500.0
JSFusion (Yu et al., 2018)	10.2	31.2	43.2	13.0	–	–	–	–
HT100M (Miech et al., 2019)	12.1	35.0	48.0	12.0	–	–	–	–
JPoSE (Wray et al., 2019)	14.3	38.1	53.0	9.0	16.4	41.3	54.4	8.7
CE (Liu et al., 2019)	20.9	48.8	62.4	6.0	20.6	50.3	64.0	5.3
MMT (Gabeur et al., 2020)	24.6	54.0	67.1	4.0	24.4	56.0	67.8	4.0
Ours	**27.4**	**56.3**	**67.7**	**3.0**	**26.6**	**55.1**	**67.5**	**3.0**
VidTranslate (Korbar et al., 2020)	14.7	–	52.8	–	–	–	–	–
HT100M (Miech et al., 2019)	14.9	40.2	52.8	9.0	16.8	41.7	55.1	8.0
NoiseEstimation (Amrani et al., 2020)	17.4	41.6	53.6	8.0	–	–	–	–
UniVL (Luo et al., 2020)	21.2	49.6	63.1	6.0	–	–	–	–
AVLnet (Rouditchenko et al., 2020)	27.1	55.6	66.6	4.0	28.5	54.6	65.2	4.0
MMT (Gabeur et al., 2020)	26.6	57.1	**69.6**	4.0	27.0	57.5	69.7	3.7
Ours-pretrained	**30.1**	**58.5**	69.3	**3.0**	**28.5**	**58.6**	**71.6**	**3.0**

Table 4: **Retrieval performance on the VATEX dataset**

	Text →Video				Video →Text			
	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	Md$R\downarrow$	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	Md$R\downarrow$
Random Baseline	0.2	0.7	1.05	2000.5	0.02	0.1	1.02	2100.5
VSE (Kiros et al., 2014)	28.0	64.3	76.9	3.0	–	–	–	–
VSE++ (Faghri et al., 2018)	33.7	70.1	81.0	2.0	–	–	–	–
Dual (Dong et al., 2019)	31.1	67.4	78.9	3.0	–	–	–	–
HGR (Chen et al., 2020b)	35.1	73.5	83.5	2.0	–	–	–	–
Ours	**44.6**	**81.8**	**89.5**	**1.0**	**58.1**	**83.8**	**90.9**	**1.0**
Ours-pretrained	**45.9**	**82.4**	**90.4**	**1.0**	**61.2**	**85.2**	**91.8**	**1.0**

Table 5: **Retrieval performance on ActivityNet**

	Text →Video				Video →Text			
	$R@1\uparrow$	$R@5\uparrow$	$R@50\uparrow$	Md$R\downarrow$	$R@1\uparrow$	$R@5\uparrow$	$R@50\uparrow$	Md$R\downarrow$
Random Baseline	0.02	0.1	1.02	2458	0.02	0.1	1.02	2458
FSE(Zhang et al., 2018)	18.2	44.8	89.1	7.0	16.7	43.1	88.4	7.0
CE (Liu et al., 2019)	18.2	47.7	91.4	6.0	17.7	46.6	90.9	6.0
HSE (Zhang et al., 2018)	20.5	49.3	–	–	18.7	48.1	–	–
MMT (Gabeur et al., 2020)	22.7	54.2	93.2	5.0	22.9	54.8	93.1	4.3
Ours	**26.8**	**58.1**	**93.5**	**3.0**	**25.5**	**57.3**	**93.5**	**3.0**
MMT-pretrained (Gabeur et al., 2020)	28.7	61.4	94.5	3.3	**28.9**	**61.1**	94.3	4.0
Ours-pretrained	**29.2**	**61.6**	**94.7**	**3.0**	28.7	60.8	**94.8**	**2.0**

Table 6: **Retrieval performance on the MSVD dataset**

	Text →Video				Video →Text			
	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	Md$R\downarrow$	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	Md$R\downarrow$
VSE (Kiros et al., 2014)	12.3	30.1	42.3	14.0	–	–	–	–
VSE++ (Faghri et al., 2018)	15.4	39.6	53.0	9.0	–	–	–	–
Multi. Cues (Mithun et al., 2018)	20.3	47.8	61.1	6.0	–	–	–	–
CE (Liu et al., 2019)	19.8	49.0	63.8	6.0	–	–	–	–
Ours	**23.0**	**52.8**	**65.8**	**5.0**	**27.3**	**50.7**	**60.8**	**5.0**
Ours-pretrained	**28.4**	**60.0**	**72.9**	**4.0**	**34.7**	**59.9**	**70.0**	**3.0**

4.3 Comparison to State-of-the-Art

In this section, we compare the results of our method to other recent text-to-video and video-to-text retrieval approaches on various datasets. In Tab. 3 to 5, we show the results of our model applied to text-to-video and video-to-text retrieval on MSR-VTT, VATEX, ActivityNet and MSVD with and without pre-trainig on HowTo100M. Without pre-training, our method outperforms all others in all metrics and datasets. In particular, for the VATEX dataset, our retrieval performance at recall at 1 and 5 is 45.9% and 82.4%, exceeding recent state-of-the-art methods (Chen et al., 2020b) by a margin of 9%. For ActivityNet, our model outperforms MMT by a margin of 4% at recall at 1. With pre-training on HowTo100M, our performance further increases across the board. Notably, unlike MMT which uses 7 features, our model uses only 2 features and achieves state-of-the-art in most metrics.

4.4 Analysis

In order to better understand the effect of our learning objective, we visualize the soft attention of our best-performing cross-instance reconstruction model in fig. 3. As we can see in the top-left square, which shows the pairwise attention between all pairs of videos in the batch, it is highly focused, with the model mostly attending one or two other instances in the batch.

For the first video's caption reconstruction (second row), we find that the model solely attends to another musical performance video that is in the batch, ignoring the others. For the second video (third row), the model focuses on another sample that shows the sea but differs in most other aspects since there are no semantically-equivalent clips in the batch. The third video shares a similar scenario. These examples show that the bottleneck is effective at forcing the model to avoid memorising the video-caption association of each clip in isolation, and attempt to match other clips more broadly, since an exact (or very close) match is not guaranteed.

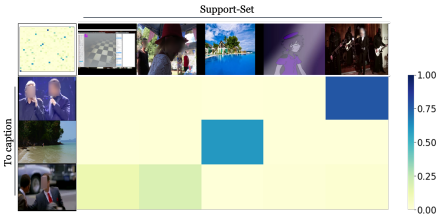


Fig. 3: **Support-set attention map.** Attention scores of all pairs in a batch (top-left square) and a subset of rows/columns (other squares) on VTT.

5 Conclusion

In this work, we studied classic contrastive learning methods such as the triplet loss to learn video-text representations for cross-model retrieval. We suggested that the contrastive approach might pull apart videos and captions even when they are semantically equivalent, which can hinder downstream retrieval performance. To mitigate this effect, we propose to consider a captioning pretext task as an additional learning objective. In particular, we show that cross-instance captioning can encourage the representation to pull together videos that share a similar caption, and are thus likely to be equivalent for retrieval. Leveraging these ideas, our model achieves state-of-the-art performance on the text-to-video and video-to-text retrieval tasks, on three datasets.

While we demonstrated these ideas in the specific case of text-to-video retrieval, they can in principle generalize to any setting that utilizes a contrastive loss, including self-supervised learning, provided that it is possible to learn reasonable conditional generators of a modality or data stream given another.

Acknowledgements

We are grateful for support from the Rhodes Trust (M.P.), the Royal Academy of Engineering (DFR05420, J.H), Facebook (M.P. and P.H.), EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems [EP/L015897/1] (M.P. and Y.A.) and the Qualcomm Innovation Fellowship (Y.A.). P.H. is also supported by the DARPA grant funded under the GAILA program (award HR00111990063).

REFERENCES

Jean-Baptiste Alayrac, A. Recasens, Rosália G. Schneider, R. Arandjelović, Jason Ramapuram, J. Fauw, Lucas Smaira, S. Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. *ArXiv*, abs/2006.16228, 2020.

Humam Alwassel, Bruno Korbar, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020.

Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. *arXiv preprint arXiv:2003.03186*, 2020.

Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020a.

Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020b.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

M. Caron, I. Misra, J. Mairal, Priya Goyal, P. Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 190–200. Association for Computational Linguistics, 2011.

Mark Chen, Alec Radford, Rewon Child, Jeff Wu, and Heewoo Jun. Generative pretraining from pixels. In *ICML*, 2020a.

Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. 2020b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020c.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Electra: Pre-training text encoders as discriminators rather than generators, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255, 2009.

Michael Denkowski and Alon Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the ninth workshop on statistical machine translation*, pp. 376–380, 2014.

Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *arXiv preprint arXiv:2006.06666*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.

Jeff Donahue and Karen Simonyan. Large scale adversarial representation learning. In *NeurIps*, 2019.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019.

Haodong Duan, Yue Zhao, Yuanjun Xiong, Wentao Liu, and Dahua Lin. Omni-sourced webly-supervised learning for video recognition. In *ECCV*, 2020.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*, 2018.

Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020.

Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

Jingyi Hou, Xinxiao Wu, Wentian Zhao, Jiebo Luo, and Yunde Jia. Joint syntax representation learning and visual cue translation for video captioning. In *ICCV*, 2019.

Po-Yao Huang, Xiaojun Chang, and Alexander Hauptmann. Multi-head attention with diversity for learning grounded multilingual multimodal representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 1461–1467, November 2019.

Po-Yao Huang, Junjie Hu, Xiaojun Chang, and Alexander Hauptmann. Unsupervised multimodal neural machine translation with pseudo visual pivoting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8226–8237. Association for Computational Linguistics, July 2020.

Po-Yao Huang, Mandela Patrick, Junjie Hu, Graham Neubig, Florian Metze, and Alexander Hauptmann. Multilingual multimodal pre-training for zero-shot cross-lingual transfer of vision-language models. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, June 2021.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation, 2018.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

Bruno Korbar, Fabio Petroni, Rohit Girdhar, and Lorenzo Torresani. Video understanding as machine translation, 2020.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *CVPR*, 2017.

H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. Tvqa: Localized, compositional video question answering. In *EMNLP*, pp. 1369–1379, 2018.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. Pre-training via paraphrasing, 2020a.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *ACL*, 2020b.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, Daxin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *AAAI*, 2020a.

Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations, 2020b.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. Jointly localizing and describing events for dense video captioning. In *CVPR*, 2018.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, 2019.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIps*, 2019.

Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation, 2020.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020.

Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pp. 19–27, 2018.

Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. *arXiv preprint arXiv:2004.12943*, 2020.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations, 2020.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020.

Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pp. 5533–5541, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Chris Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *International Journal of Computer Vision*, 2017.

Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, et al. Avlnet: Learning audio-visual language representations from instructional videos. *arXiv preprint arXiv:2006.09199*, 2020.

Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations, 2020.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations, 2019.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. 2017.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer, 2019a.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019b.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NeurIps*, pp. 3104–3112, 2014.

Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIps*, 2017.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4566–4575, 2015.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015a.

Subhashini Venugopalan, Marcus Rohrbach, Jeff Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. Sequence to sequence – video to text. In *ICCV*, 2015b.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4581–4591, 2019.

Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 305–321, 2018.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*, 2016.

Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. Citeseer, 2015.

I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification, 2019.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018.

Bowen Zhang, Hexiang Hu, and Fei Sha. Cross-modal and hierarchical modeling of video and text. In *ECCV*, pp. 374–390, 2018.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

Ziqi Zhang, Yaya Shi, Chunfeng Yuan, Bing Li, Peijin Wang, Weiming Hu, and Zheng-Jun Zha. Object relational graph with teacher-recommended learning for video captioning. In *CVPR*, 2020.

Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. 2018a.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018b.

Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

6 APPENDIX

The appendix is organized as follows: First, we provide more details about our model. Then we introduce the datasets and the experimental setup. Finally, we provide additional qualitative and quantitative experimental results for video-text retrieval and captioning.

6.1 MODEL DETAILS

Implementation details and hyper parameters. For our text encoder, we use the T5-base model pre-trained on the "Colossal Clean Crawled Corpus" (C4) (Raffel et al., 2019). We use its corresponding text tokenizer and encode a sentence into a sequence of 1024 dimensional vectors.

For our visual encoder, our model utilizes only the motion and the appearance features. For the motion feature, we use a 34-layer, R(2+1)-D (Tran et al., 2018) model pre-trained on IG65M (Ghadiyaram et al., 2019) and apply a spatial-temporal average pooling over the last convolutonal layer, resulting in a 512-dimensional vector. For the appearance feature, we use the 2048-dimension flattened pool-5 layer of the standard ResNet152 (He et al., 2016) pre-trained on Imagenet (Deng et al., 2009). We extract features at a rate of 1 feature per second and simply concatenate the two features, resulting in a 2560-dimension visual input stream. Noteworthily, instead of using 9 and 7 different types of visual features as in CE (Liu et al., 2019) and MMT (Gabeur et al., 2020), we use only the above 2 features and achieve on par or superior performance. Also, with early fusion, our model does not suffer from additional computation required for the extended sequence length in MMT. For the text decoder, we use the T5-base model decoder, also pre-trained on C4.

As illustrated in Fig. 4, our transformer pooling head is composed of a pre-encoder, a multi-head self-attention (MHA), and a feed-forward layer (FFN). For pre-encoders, we use a one-layer MLP with a d-dimensional output for mapping video features into the common embedding space. We use 1024-dimension bi-directional GRU as the text pre-encoder. For the 1D-CNN prior, we use kernels with size $[2, 3, 4, 6]$ as the visual and text pre-encoders. We set the embedding dimension to 1024 and use 4 attention heads in the transformer pooling layers. The hidden dimension of FFN is 2048.

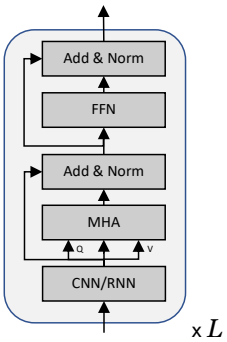


Fig. 4: Transformer pooling head.

Training and Inference time. Pre-training on 1.2 million HowTo100M videos takes around 160 GPU hours (NVIDIA V100) for 20 epochs. We speed up the pre-training process by distributing the workload over 8 GPUs. We use 1 GPU for the fine-tuning or training from scratch experiments. For the MSR-VTT 1k-A split, it takes 12 GPU hours to train our full model on 180K video-text pairs for 20 epochs. For Vatex, it takes 32 GPU hours to train on 260K video-text pairs for 30 epochs. For ActivityNet, it takes 2.5 GPU hours to train on 10K video-text paris for 28 epochs.

For inference, the encoding speed is around 250-300 video/sec and 200-250 text query/sec. The overall text-to-video search speed on 5,000 video-text pairs (5,000 text queries over 5,000 videos) is 30-34 seconds including encoding. The speed of text-to-video retrieval is similar to video-to-text retrieval.

6.2 EXPERIMENT DETAILS

The margin α of the max-margin loss is 0.2, and the temperature T is set to 0.1 as used in SimCLR Chen et al. (2020c). We use the Adam (Kingma & Ba, 2015) optimizer with a initial learning rate $5 \cdot 10^{-5}$ and clip gradients greater than 0.2 during the training phase. Dropout rate is 0.3 for all datasets besides ActivityNet (0.0).

As the average video/text lengths and videos available are quite different across datasets, we adjust our training scheme accordingly. When training on MSR-VTT, ActivtyNet and Vatex, batch-size is set to 64. For MSR-VTT training, we sample and truncate videos to 32 seconds, text to 100 tokens and train for 20 epochs. For Vatex, videos are at most 64 seconds and we train for 30 epochs. For ActivtityNet training, videos are at most 512 seconds and 256 tokens for the text part. We train

for 28 epochs on ActivityNet. For fine-tuning HowTo100M pre-trained model, we reduce training epochs into quarters.

6.3 DATASET DETAILS

HowTo100M (Miech et al., 2019) is a large-scale instructional video collection of 1.2 million Youtube videos, along with automatic speech recognition transcripts. There are more than 100 million clips (ASR segments) defined in HowTo100M. We use this dataset for pretraining.

MSR-VTT (Xu et al., 2016) contains 10,000 videos, where each video is annotated with 20 descriptions. For retrieval experiments and ablation studies, we follow the training protocol and defined in Gabeur et al. (2020); Liu et al. (2019); Miech et al. (2019) and evaluate on text-to-video and video-to-text search tasks on the 1k-A testing split with 1,000 video or text candidates defined by Yu et al. (2018). For captioning task, we evaluate on the standard testing split with 2,990 videos.

VATEX (Wang et al., 2019) is a multilingual (Chinese and English) video-text dataset with 34,911 videos. We use the official split with 25,991 videos for training. As the testing annotations are private in VATEX, we follow the protocol in Chen et al. (2020b) to split the validation set equally (1,500 validation and 1,500 testing videos) for model selection and testing. For each video, 10 English and 10 Chinese descriptions are available, and we only use the English annotations.

ActivityNet Dense Caption dataset consists densely annotated temporal segments of 20K YouTube videos. Following Gabeur et al. (2020); Zhang et al. (2018), we concatenate descriptions of segments in a video to construct "video-paragraph" for retrieval and captioning. We use the 10K training split to train from scratch/ finetune the model and report the performance on the 5K 'val1' split.

MSVD dataset consists of $80K$ English descriptions for 1,970 videos from YouTube, with each video associated with around 40 sentences each. We use the standard split of 1200, 100, and 670 videos for training, validation, and testing (Liu et al., 2019; Venugopalan et al., 2015b; Xu et al., 2015).

6.4 VIDEO CAPTIONING EXPERIMENTS

To measure captioning/text generation performance, we report BLEU4 (Papineni et al., 2002), METEOR (Denkowski & Lavie, 2014), Rogue-L (Lin, 2004) and CIDEr (Vedantam et al., 2015) metrics. We report results on the MSR-VTT, VATEX and ActivityNet datasets.

Table 7: **Captioning performance on the MSR-VTT dataset**

	Captioning			
	BLUE4	METEOR	Rogue-L	CIDEr
VidTranslate (Korbar et al., 2020)	41.7	28.5	–	–
POS+VCT (Hou et al., 2019)	42.3	29.7	62.8	49.1
ORG (Zhang et al., 2020)	43.6	28.8	62.1	50.9
Ours, MSR-VTT only	39.7	28.3	60.5	46.5
Ours, HT100M + MSR-VTT	38.9	28.2	59.8	48.6

Table 8: **Captioning performance on the VATEX dataset**

	Captioning			
	Blue@4	METEOR	Rogue-L	CIDEr
Shared Enc-Dec (Wang et al., 2019)	28.4	21.7	47.0	45.1
ORG (Zhang et al., 2020)	32.1	22.2	48.9	49.7
Ours, VATEX only	**32.8**	**24.4**	**49.1**	**51.2**
Ours, HT100M + Vatex	32.5	24.1	48.9	50.5

Table 9: **Captioning performance on the ActivtyNet dataset**

	Captioning			
	Blue@4	METEOR	Rogue-L	CIDEr
DENSE (Krishna et al., 2017)	1.6	8.9	–	–
DVC-D-A (Li et al., 2018)	1.7	9.3	–	–
Bi-LSTM+TempoAttn (Zhou et al., 2018b)	2.1	10.0	–	–
Masked Transformer (Zhou et al., 2018b)	2.8	11.1	–	–
Ours, ActivityNet only	1.5	6.9	17.8	3.2
Ours, HT100M + ActivityNet	1.4	6.9	17.5	3.1

6.5 Zero-Shot Retrieval Experiments

We also evaluate our model in the zero-shot setting on MSR-VTT, Vatex, ActivityNet and MSVD, after pre-training on HT100M. While we are able to get reasonable results on MSR-VTT and MSVD, our results are not great on Vatex and Activity-Net due to significant domain gap.

Table 10: **Zero-shot Retrieval performance on VATEX, MSR-VTT, MSVD and ActivityNet.**

	Text →Video				Video →Text			
	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$
Zero-Shot								
ActivityNet	0.06	0.2	0.5	1907.0	0.0	0.2	0.3	2238.0
VATEX	0.07	0.4	0.7	682.0	0.07	0.4	0.9	697
MSVD	8.9	26.0	37.9	18.0	21.4	46.2	57.7	6.0
MSR-VTT	8.7	23.0	31.1	31.0	12.7	27.5	36.2	24.0

6.6 Action Recognition

Lastly, we evaluate our model on the video action recognition task on HMDB-51 (Kuehne et al., 2011) and UCF-101 (Soomro et al., 2012). For this, we use the R(2+1)D-34 (pretrained on IG65M) model as well as a ResNet-152 model (pretrained on Imagenet), as in our method. We extract a feature per second per video by concatenating the features from each model (2560-D), and obtain an average representation per video using either average pooling (2560-D) or our proposed transformer pooling head (1024-D) pre-trained on HT100M using cross-captioning objective. We then train a linear classifier for 1500 epochs for HMDB-51 (500 for UCF-101) on these features using Adam (Kingma & Ba, 2015) optimizer with learning rate of $1e^{-4}$ and weight decay $1e^{-4}$ with early stopping. We also drop the learning rate by 10 at epochs 200, 400 for UCF-101 and 1000, 1200 for HMDB-51. In Table 11, we show the results of training only a linear-layer on features extracted from our fixed backbone with or without a learned transformer-pooling head. We find that our transformer temporal pooling head provides significant benefits over the baseline of simply average pooling the features, demonstrating the effectiveness of building contextualized representations using our proposed transformer. In particular, we see improvements of over 7% on HMDB-51 and 34% on UCF-101 by replacing average pooling with our transformer pooling head to aggregate features. We observe that naive average pooling performs significantly worse than our transformer pooling under evaluation protocol. This is likely because 1) the average pooling collapses temporal information, making the linear layer based classification difficult 2) compared to the transformer pooling, it does not benefit from large-scale pretraining on a wide variety of action videos of HT100M. We further compare very favorably to the current state-of-the-art approaches. In particular, we outperform all other approaches, both supervised and self-supervised, except the recently introduced Omni (Duan et al., 2020) which was finetuned on both UCF-101 and HMDB-51, while we only trained a linear classifier on extracted features. However, it should be noted that it is very difficult to fairly compare all these different approaches because they may use different modalities (images, RGB video, optical flow, audio, ASR outputs), pretraining datasets (Kinetics-400, HT100M, IG65M, Imagenet), architectures (S3D, I3D, R(2+1)D, R3D), pre-training (supervised, self-supervised) and downstream training (frozen, finetuned) strategies.

Table 11: **Action recognition.** Results of training only a linear-layer, on features extracted from our fixed backbone with or without a learned transformer-pooling head. We compare to the state-of-art supervised and self-supervised pretrainig methods on the HMDB-51 and UCF-101 action recognition task, for different downstream training protocols ("FT?" stands for finetuned). We report average Top-1 accuracy across all 3 folds. Dataset abbreviations: **A**udio**S**et, **H**MDB**51**, **H**owTo100**M**, **I**nsta**g**ram**65M**, **IM**agenet-1000, **K**inetics**400**, **O**mni**S**ource Images + Videos, **S**ports**1M**, **U**CF**101**, **Y**ou**T**ube**8M**. Other abbreviations: **V**ideo modality, **F**low modality, **I**mage modality, **A**udio modality, **T**ransformer pooling, **Av**era**g**e pooling

Method	Mod	Dataset	Model	FT?	H51	U101
Self-Supervised Pre-training						
MIL-NCE (Miech et al., 2020)	V,T	HM	S3D-G	✗	53.1	82.7
MIL-NCE (Miech et al., 2020)	V,T	HM	S3D-G	✓	61.0	91.3
MMV (Alayrac et al., 2020)	V,T,A	HM+AS	TSM-50x2	✗	67.1	91.8
ELo (Piergiovanni et al., 2020)	V,F,A	YT8M	R(2+1)D-50x3	✓	67.4	93.8
XDC (Alwassel et al., 2020)	V,A	IG65M	R(2+1)D-18	✓	68.9	95.5
GDT (Patrick et al., 2020)	V,A	IG65M	R(2+1)D-18	✓	72.8	95.2
MMV (Alayrac et al., 2020)	V,T,A	HM+AS	TSM-50x2	✓	75.0	95.2
Supervised Pre-training						
P3D (Qiu et al., 2017)	V,I	S1M+IM	P3D	✓	–	88.6
TSN (Wang et al., 2018)	V,I	IM	TSN	✓	69.4	94.2
I3D (Carreira & Zisserman, 2017)	V,I	K400+IM	I3D	✓	74.8	95.6
R(2+1)D (Tran et al., 2018)	V	K400	R(2+1)D-34	✓	74.5	96.8
S3D-G (Xie et al., 2018)	V,I	K400+IM	S3D-G	✓	75.9	96.8
I3D (Carreira & Zisserman, 2017)	V,I	K400+IM	I3D	✓	77.1	96.7
R(2+1)D (Tran et al., 2018)	V	K400	R(2+1)D-34	✓	76.4	95.5
R(2+1)D (Tran et al., 2018)	V,F	K400	R(2+1)D-34x2	✓	78.7	97.3
Omni (Duan et al., 2020)	V,I	K400+OS	Slow-8x8-R101	✓	79.0	97.3
I3D (Carreira & Zisserman, 2017)	V,F,I	K400+IM	I3Dx2	✓	80.7	98.0
Omni (Duan et al., 2020)	V,F,I	K400+OS	Slow-8x8-R101x2	✓	**83.8**	**98.6**
Ours (Avg-pooling)	V,I	IG65M+IM	R(2+1)D-34+R152	✗	73.7	64.3
Ours (T-pooling)	V,I	HM+IG65M+IM	R(2+1)D-34+R152	✗	81.3	98.0

6.7 STATISTICAL SIGNIFICANCE

In Table 12, we show the results of finetuning our pretrained model for 3 times on the VATEX dataset. We find that the variance is quite low and our model consistently beats the state of the art.

Table 12: **Retrieval performance on the VATEX dataset**

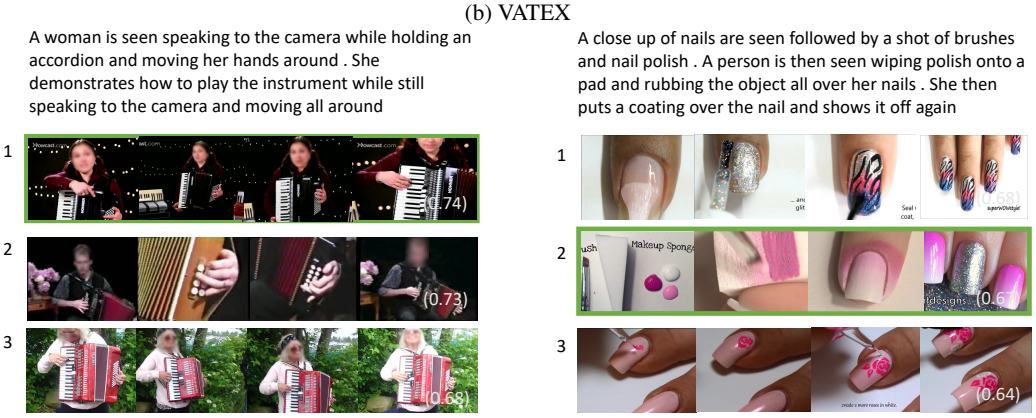
	Text →Video				**Video →Text**			
	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	Md$R\downarrow$	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	Md$R\downarrow$
Random Baseline	0.2	0.7	1.05	2000.5	0.02	0.1	1.02	2100.5
VSE (Kiros et al., 2014)	28.0	64.3	76.9	3.0	–	–	–	–
VSE++ (Faghri et al., 2018)	33.7	70.1	81.0	2.0	–	–	–	–
Dual (Dong et al., 2019)	31.1	67.4	78.9	3.0	–	–	–	–
HGR (Chen et al., 2020b)	35.1	73.5	83.5	2.0	–	–	–	–
Ours	**44.9**$_{\pm 0.2}$	**82.1**$_{\pm 0.2}$	**89.7**$_{\pm 0.2}$	**1.0**	**58.4**$_{\pm 0.1}$	**84.4**$_{\pm 0.2}$	**91.0**$_{\pm 0.3}$	**1.0**

6.8 ADDITIONAL QUALITATIVE RESULTS

We provide addition qualitative text-to-video retrieval results on MSR-VTT, VATEX, ActivityNet in Fig. 5. Given a text query, in most cases, our model successfully retrieves the correct videos marked in green.

(a) MSR-VTT

(b) VATEX



(c) ActivityNet

Fig. 5: Examples of top-3 Text→Video retrieval results and similarities on the MSR-VTT, VATEX, and ActivityNet testing set. Only one correct video (colored in green) for each text query on the top.

5

Multilingual Multimodal Pre-training for Zero-Shot Cross-Lingual Transfer of Vision-Language Models

This work was presented at the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL) 2021.

Multilingual Multimodal Pre-training for Zero-Shot Cross-Lingual Transfer of Vision-Language Models

Po-Yao Huang[13]*, **Mandela Patrick**[23]*, **Junjie Hu**[1],
Graham Neubig[1], **Florian Metze**[3], **Alexander Hauptmann**[1]
[1]School of Computer Science, Carnegie Mellon University
[2]Visual Geometry Group, University of Oxford
[3]Facebook AI
{poyaoh,junjieh,gneubig,alex}@cs.cmu.edu, {mandelapatrick,fmetze}@fb.com

Abstract

This paper studies zero-shot cross-lingual transfer of vision-language models. Specifically, we focus on multilingual text-to-video search and propose a Transformer-based model that learns contextual multilingual multimodal embeddings. Under a zero-shot setting, we empirically demonstrate that performance degrades significantly when we query the multilingual text-video model with non-English sentences. To address this problem, we introduce a multilingual multimodal pre-training strategy, and collect a new multilingual instructional video dataset (Multi-HowTo100M) for pre-training. Experiments on VTT show that our method significantly improves video search in non-English languages without additional annotations. Furthermore, when multilingual annotations are available, our method outperforms recent baselines by a large margin in multilingual text-to-video search on VTT and VATEX; as well as in multilingual text-to-image search on Multi30K. Our model and Multi-HowTo100M is available at http://github.com/berniebear/Multi-HT100M

1 Introduction

One of the key challenges at the intersection of computer vision (CV) and natural language processing (NLP) is building versatile vision-language models that not only work in English, but in all of the world's approximately 7,000 languages. Since collecting and annotating task-specific parallel multimodal data in all languages is impractical, a framework that makes vision-language models generalize across languages is highly desirable.

One technique that has shown promise to greatly improve the applicability of NLP models to new languages is *zero-shot cross-lingual transfer*, where models trained on a source language are applied as-is to a different language without any additional annotated training data (Täckström et al., 2012; Klementiev et al., 2012; Cotterell and Heigold, 2017; Chen et al., 2018; Neubig and Hu, 2018). In particular, recent techniques for cross-lingual transfer have demonstrated that by performing unsupervised learning of language or translation models on many languages, followed by downstream task fine-tuning using only English annotation, models can nonetheless generalize to a non-English language (Wu and Dredze, 2019a; Lample and Conneau, 2019; Huang et al., 2019a; Artetxe et al., 2020; Hu et al., 2020). This success is attributed to the fact that many languages share a considerable amount of underlying vocabulary or structure. At the vocabulary level, languages often have words that stem from the same origin, for instance, "desk" in English and "Tisch" in German both come from the Latin "discus". At the structural level, all languages have a recursive structure, and many share traits of morphology or word order.

For cross-lingual transfer of vision-language models, the visual information is clearly an essential element. To this end, we make an important yet under-explored step to incorporate visual-textual relationships for improving multilingual models (Devlin et al., 2019; Artetxe et al., 2020). While spoken languages could be different, all humans share similar vision systems, and many visual concepts can be understood universally (Sigurdsson et al., 2020; Zhang et al., 2020). For example, while  is termed "cat" for an English speaker and "chat" for a French speaker; they understand  similarly. We leverage this observation to learn to associate sentences in different languages with visual concepts for promoting cross-lingual transfer of vision-language models.

In this work, we focus on multilingual text-to-video search tasks and propose a Transformer-based video-text model to learn contextual mul-

*Equal contribution.

tilingual multimodal representations. Our vanilla model yields state-of-the-art performance in multilingual text→video search when trained with multilingual annotations. However, under the zero-shot setting, rather surprisingly, there is a significant performance gap between English and non-English queries (see §5.5 for details). To resolve this problem, motivated by recent advances in large-scale language model (Artetxe et al., 2020) and multimodal pre-training (Lu et al., 2019; Miech et al., 2019; Patrick et al., 2020), we propose a multilingual multimodal pre-training (MMP) strategy to exploit the weak supervision from large-scale multilingual text-video data. We construct the Multilingual-HowTo100M dataset, that extends the English HowTo100M (Miech et al., 2019) dataset to contain subtitles in 9 languages for 1.2 million instructional videos.

Our method has two important benefits. First, compared to pre-training on English-video data only, pre-training on multilingual text-video data exploits the additional supervision from a variety of languages, and therefore, enhances the search performance on an individual language. Second, by exploiting the visual data as an implicit "pivot" at scale, our methods learns better alignments in the multilingual multimodal embedding space (*e.g.*, "cat"-  -"chat"), which leads to improvement in zero-shot cross-lingual transfer (*e.g.*, from "cat"-  to "chat"- ) of vision-language models.

In our experiments on VTT (Xu et al., 2016) and VATEX (Wang et al., 2019), our method yields state-of-the-art English→video search performance. For zero-shot cross-lingual transfer, the proposed multilingual multimodal pre-training improves English-video pre-training by $2 \sim 2.5$ in average R@1 across 9 languages. Additionally, when trained with in-domain multilingual annotations as other baselines, our method outperforms them by a large margin in multilingual text→video search on VATEX and text→image search on Multi30K (Elliott et al., 2016).

To summarize, we make the following contributions: (1) We propose a transformer-based video-text model that learns contextual multilingual multimodal representations (§3.1). (2) We empirically demonstrate that vision-language models, unlike NLP models, have limited zero-shot cross-lingual transferrability. (§5.5). (3) We introduce the multilingual multimodal pre-training strategy and construct a new Multi-HowTo100M dataset (§4) for pre-training to improve zero-shot cross-lingual capability of vision-language models. (4) We demonstrate the effectiveness of our approach, by achieving state-of-the-art multilingual text→video search performance in both the zero-shot (§5.5) and fully supervised setup (§5.6).

2 Related Work

Cross-lingual representations. Early work on learning non-contextual cross-lingual representations used either parallel corpora (Gouws and Søgaard, 2015; Luong et al., 2015) or a bilingual dictionary to learn a transformation (Faruqui and Dyer, 2014; Mikolov et al., 2013). Later approaches reduced the amount of supervision using self-training (Artetxe et al., 2017). With the advances in monolingual transfer learning (McCann et al., 2017; Howard and Ruder, 2018; Peters et al., 2018; Devlin et al., 2019), multilingual extensions of pre-trained encoders have been proven effective in learning deep contextual cross-lingual representations (Eriguchi et al., 2017; Lample and Conneau, 2019; Wu and Dredze, 2019b; Siddhant et al., 2020; Pires et al., 2019; Pfeiffer et al., 2020). We extend prior work to incorporate visual context.

Video-text representations. The HowTo100M dataset (Miech et al., 2019) has attracted significant interest in leveraging multimodal pre-training for text→video search (Korbar et al., 2020), captioning (Iashin and Rahtu, 2020), and unsupervised translation via image-based (Surís et al., 2020; Huang et al., 2020b) and video-based (Sigurdsson et al., 2020) alignment. This work studies a challenging and unexplored task: Zero-shot cross-lingual transfer of vision-language models. Unlike prior image/video-text work that utilizes RNN (Dong et al., 2019; Chen et al., 2020a; Burns et al., 2020; Kim et al., 2020) and inter-modal contrastive objectives (Sigurdsson et al., 2020; Liu et al., 2019; Huang et al., 2019b; Patrick et al., 2021), we employ Transformers to learn contextual multilingual multimodal representations and uniquely models cross-lingual instances. Moreover, we build Multi-HowTo100M, the largest text-video dataset for multilingual multimodal pre-training.

Cross-lingual Transfer. Cross-lingual transfer has proven effective in many NLP tasks including dependency parsing (Schuster et al., 2019), named entity recognition (Rahimi et al., 2019), sentiment analysis (Barnes et al., 2019), document classification (Schwenk and Li, 2018), and question an-

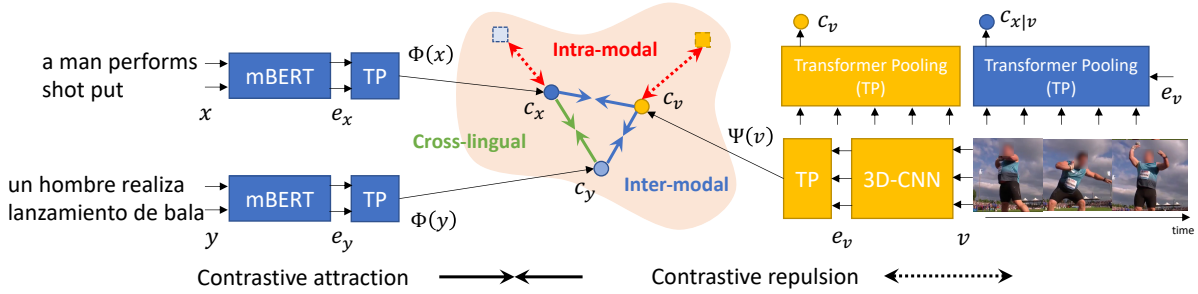


Figure 1: The proposed video-text model for learning contextual multilingual multimodal representations. We utilize *intra-modal*, *inter-modal*, and conditional *cross-lingual* contrastive objectives to align (x, v, y) where x and y are the captions or transcriptions in different languages of a video v. TP: Transformer pooling head.

swering (Lewis et al., 2020; Artetxe et al., 2020). Recently, XTREME (Hu et al., 2020) was proposed to evaluate the cross-lingual transfer capabilities of multilingual representations across a diverse set of NLP tasks and languages. However, a comprehensive evaluation of multilingual multimodal models on zero-shot cross-lingual transfer capabilities is still missing. To our best knowledge, we are the first work that investigates and improves zero-shot cross-lingual transfer of vision-language models.

3 Method

We consider the problem of learning multilingual multimodal representations from a corpus $\mathcal{C}$ of video-text pairs $\{(x_i, v_i)\}_{i=1}^{C}$, where v_i is a video clip and x_i is its corresponding text (caption or transcription) that is written in one of K languages. Our goal is to learn a shared multilingual text encoder $c_x = \Phi(x)$ and a video encoder $c_v = \Psi(v)$, both of which project the input to a shared D-dimensional embedding space $c_v, c_t \in \mathbb{R}^D$, where semantically similar instances (*i.e.*, paired (x_i, v_i)) are closer to each other than the dissimilar ones (*i.e.*, $(x_i, v_j), i \neq j$). In the following, we denote a batch of multilingual text-video samples as $\mathcal{B} = \{(x_i, v_i)\}_{i=1}^{B}\}$ where $\mathcal{B} \subset \mathcal{C}$.

3.1 Multilingual Multimodal Transformers

Figure 1 gives an overview of the proposed method. Our text encoder consists of a multilingual Transformer (*e.g.* multilingual BERT (Devlin et al., 2019)) and a text Transformer pooling head (explained below). Similarly, our video encoder consists of a 3D-CNN (*e.g.* R(2+1)D network (Tran et al., 2018)) and a video Transformer pooling head. We use these multilingual multimodal Transformers to encode text and video for alignment.

Unlike prior multilingual text-image models (Gella et al., 2017; Kim et al., 2020; Huang et al., 2019b) that utilize word embeddings and RNNs, our multilingual text encoder is built on a multilingual Transformer that generates contextual multilingual representations $e_x \in \mathbb{R}^{N \times D}$ to encode a sentence x containing N words. We employ an additional 2-layer Transformer which we will call a "Transformer pooling head (TP)" as it serves as a pooling function to selectively encode variable-length sentences and aligns them with the corresponding visual content. We use the first output token of the second Transformer layer as the final sentence representation. Precisely, we set $c_x = \text{Trans}_x^{(2)}(\text{query=key=value}=e_x)[0]$ where $\text{Trans}_x^{(2)}$ is a 2-layer stack of Transformers (Vaswani et al., 2017) with e_x as the (query,key,value) in the multi-head attention. Note that we use the same text encoder to encode sentences in all languages.

For encoding videos, our model uses pre-trained 3D-CNNs that encode spatial-temporal context in a video. For a M-second video v, we apply R(2+1)D (Tran et al., 2018) and S3D (Miech et al., 2020) networks to its frames, concatenate network outputs, and apply a linear layer to encode the visual input, $e_v \in \mathbb{R}^{M \times D}$, to our model. Similarly to the text part, we employ a two-layer Transformer as the pooling head to encode videos with different lengths into fixed-length representations. Formally, we set $c_v = \text{Trans}_v^{(2)}(\text{query=key=value}=e_v)[0]$. Since videos are typically long and have a high frame rate (*e.g.*, 30 fps), it is infeasible to update 3D-CNNs simultaneously and therefore, we use pre-extracted video features. Our model is parameterized by $\theta = \theta_{\text{mBERT}} \cup \theta_{\text{Trans}_x} \cup \theta_{\text{Trans}_v}$.

3.2 Multilingual Text-Video Alignment

For learning multimodal representations, the common practice is to minimize a contrastive objective to map the associated (video, text) embeddings

to be near to each other in a shared embedding space. The inter-modal max-margin triplet loss has been widely studied in video-text (Yu et al., 2018; Liu et al., 2019) and image-text (Kim et al., 2020; Burns et al., 2020; Huang et al., 2019b) research. In this work, we generalize and model all *inter-modal*, *intra-modal*, and *cross-lingual* instances with a noise contrastive estimation objective (NCE) (Gutmann and Hyvärinen, 2010; van den Oord et al., 2018; Chen et al., 2020b).

Inter-modal NCE. Let $\mathcal{X}$ and $\mathcal{V}$ denote the subsets of the sampled sentences in multiple languages and videos in $\mathcal{B}$, respectively. And let $s(a,b) = \frac{a^T b}{\|a\|\|b\|}$ be the cosine similarity measure. We use an (inter-modal) NCE objective defined as:

$$\mathcal{L}(\mathcal{X},\mathcal{V}) = -\frac{1}{B}\sum_{i=1}^{B}\log\ell^{\text{NCE}}(\Phi(x_i),\Psi(v_i)), \quad (1)$$

where

$$\ell^{\text{NCE}}(c_x,c_v) = \frac{e^{s(c_x,c_v)}}{e^{s(c_x,c_v)} + \sum_{(x',v')\sim\mathcal{N}} e^{s(c_{x'},c_{v'})}} \quad (2)$$

In inter-modal NCE, $\mathcal{L}^{\text{inter}} = \mathcal{L}(\mathcal{X},\mathcal{V})$, the noise $\mathcal{N}$ is a set of "negative" video-text pairs sampled to enforce the similarity of paired ones are high and and those do not are low. Following Miech et al. (2020), we set the negatives of (x_i, v_i) as other x_j and $v_j, j \neq i$ in $\mathcal{B}$.

Intuitively, inter-modal NCE draws paired (semantically similar) instances closer and pushes apart non-paired (dissimilar) instances. Note that we do not distinguish language types in $\mathcal{X}$ and the sentences in all possible languages will be drawn towards their corresponding videos in the shared multilingual text-video embedding space.

Intra-modal NCE. Beyond cross-modality matching, we leverage the intra-modal contrastive objective to learn and preserve the underlying structure within the video and text modality. For example, *Corgi* should be closer to *Husky* than *Balinese*. Prior image-text work (Gella et al., 2017; Huang et al., 2019c) utilizes a triplet loss to maintain such neighborhood relationships. Inspired by recent success in self-supervised image and video representation learning (Yalniz et al., 2019; Ghadiyaram et al., 2019), our model leverages intra-modal NCE that constrains the learned representations to be invariant against noise and to maintain the within-modality structure simultaneously. We minimize the following intra-modal NCE loss:

$$\mathcal{L}^{\text{intra}} = \mathcal{L}(\mathcal{X},\mathcal{X}^m) + \mathcal{L}(\mathcal{V},\mathcal{V}^m), \quad (3)$$

where $\mathcal{X}^m$ and $\mathcal{V}^m$ are the noised version of the original sentences and videos. For noising, we randomly mask 5% of the multilingual text tokens and video clips. We optimize our model by

$$\min_\theta \mathcal{L}^{\text{inter}} + \mathcal{L}^{\text{intra}} \quad (4)$$

3.3 When Visually-Pivoted Multilingual Annotations Are Available

In many multilingual multimodal datasets, there are sentences in different languages that describe a shared visual context. For example, 10 English and 10 Chinese descriptions are available for each video in VATEX. With these visually-pivoted (weakly paralleled) sentences (x, y), we further revise the contrastive objectives to leverage this additional supervisory signal. Given a visually-pivoted corpus $\mathcal{C}^p$ that contains all possible combination of visually-pivoted pairs $\{(x_i, v_i, y_i)\}_{i=0}^{C_p}$, we sample batches $\mathcal{B}^p = \{(x_i, v_i, y_i)\}_{i=1}^{\mathcal{B}^p}, \mathcal{B}^p \subset \mathcal{C}^p$ and revise the contrastive objective as:

$$\mathcal{L}^{\text{inter}} = \mathcal{L}(\mathcal{X},\mathcal{V}) + \mathcal{L}(\mathcal{Y},\mathcal{V}) \quad (5)$$

$$\mathcal{L}^{\text{intra}} = \mathcal{L}(\mathcal{X},\mathcal{X}^m) + \mathcal{L}(\mathcal{Y},\mathcal{Y}^m) + \mathcal{L}(\mathcal{V},\mathcal{V}^m) \quad (6)$$

Visual-pivoted Cross-lingual NCE. Inspired by Translation Language Modeling (TLM) in XLM (Lample and Conneau, 2019), we propose a multimodal TLM-like contrastive objective which promotes alignments of descriptions in different languages that describe the same video. We use the intuition that conditioned on a video, the descriptions (need not to be translation pairs) in different languages would likely be semantically similar. To this end, we set the cross-lingual NCE as:

$$\mathcal{L}^{\text{cross}} = \mathcal{L}(\mathcal{X}|\mathcal{V},\mathcal{Y}|\mathcal{V}) \quad (7)$$

For visually-pivoted sentences, as shown in Fig. 1, we generate their representations conditioned on the video they describe. We extend the *key* and *value* of multihead attention with the additional visual content e_v and generate new $c_{x|v}$ and $c_{y|v}$ for matching. Specifically, our model employs $c_{x|v} = \text{Trans}_x^{(2)}(\text{query}{=}e_x, \text{key}{=}\text{value}{=}e_x||e_v)[0]$. With the access to (visually-pivoted) multilingual annotations, we optimize our model by

$$\min_\theta \mathcal{L}^{\text{inter}} + \mathcal{L}^{\text{intra}} + \mathcal{L}^{\text{cross}} \quad (8)$$

Figure 2: Video clips and the corresponding multilingual subtitles in Multi-HowTo100M.

At the inference time, we simply apply $c_x = \Phi(x)$ and $c_v = \Psi(v)$ to encode multilingual text queries and videos. For text-to-video search, we sort videos according to their cosine similarity scores to the text query.

4 The Multilingual HowTo100M Dataset

As large-scale pre-training has been shown important in recent NLP and vision-language models, we construct the **Multilingual HowTo100M** dataset (Multi-HowTo100M) to facilitate research in multilingual multimodal learning. The original HowTo100M (Miech et al., 2019) dataset is a large-scale video collection of 1.2 million instructional videos (around 138 million clips/segments) on YouTube, along with their automatic speech recognition (ASR) transcriptions as the subtitles. For each video in HowTo100M, we crawl and collect the multilingual subtitles provided by YouTube, which either consist of user-generated subtitles or those generated by Google ASR and Translate in the absence of user-generated ones. Essentially, we collect video subtitles in 9 languages: English (*en*), German (*de*), French (*fr*), Russian (*ru*), Spanish (*es*), Czech (*cz*), Swahili (*sw*), Chinese (*zh*), Vietnamese (*vi*).

At the time of dataset collection (May 2020), there are 1.1 million videos available, each with subtitles in 7-9 languages. The video length ranges from 1 minute to more than 20 minutes. We utilize Multi-HowTo100M for multilingual multimodal pre-training to exploit the weak supervision from large-scale multilingual text-video data. In Fig. 2, we provide a visualization of few instances sampled in Multi-HowTo100M with the corresponding video frame, timestamp, and transcriptions in different languages. Please refer to Appendix for more details and dataset statistics.

5 Experiment

In this section, we first describe our experimental setup (§5.1-5.3). In §5.4, we conduct ablation studies to validate the effectiveness of proposed multilingual text-video model . With the best models at hand, we investigate their zero-shot cross-lingual transferability in §5.5, where we showcase that the proposed multilingual multimodal pre-training serves as the key facilitator. We then verify the superior text→video search performance of our method under the monolingual, multilingual, and cross-modality settings in §5.6.

5.1 Evaluation Datasets

MSR-VTT (VTT) (Xu et al., 2016) contains 10K videos, where each video is annotated with 20 captions. Additionally, we created pseudo-multilingual data by translating the English captions into 8 languages with off-the-shelf machine translation models.[1] We use the official training set (6.5K videos) and validation set (497 videos). We follow the protocol in Miech et al. (2019); Liu et al. (2019) which evaluates on text→video search with the 1K testing set defined by Yu et al. (2018).

VATEX (Wang et al., 2019) is a multilingual (Chinese and English) video-text dataset with 35K videos. Five (*en*,*zh*) translation pairs and five non-paired *en* and *zh* descriptions are available for each video. We use the official training split (26K videos) and follow the testing protocol in Chen et al. (2020a) to split the validation set equally into 1.5K validation and 1.5K testing videos.

Multi30K (Elliott et al., 2016) is a multilingual extension of Flickr30K (Young et al., 2014). For each image, there are two types of annotations available: (1) One parallel (English,German,French,Czech) translation pair and (2) five English and five Ger-

[1] https://marian-nmt.github.io/

man descriptions collected independently. The training, validation, and testing splits contain 29K, 1K, and 1K images respectively.

5.2 Implementation Details

For the video backbone, we use a 34-layer, R(2+1)-D (Tran et al., 2018) network pre-trained on IG65M (Ghadiyaram et al., 2019) and a S3D (Miech et al., 2020) network pre-trained on HowTo100M. We pre-extract video features and concatenate the two 3D-CNN outputs to form $e_x \in \mathbb{R}^{M\times 1024}$ as a video input.

For the text backbone, we use multilingual BERT (mBERT) (Devlin et al., 2019) or XLM-Roberta-large (XLM-R) (Artetxe et al., 2020), where the latter achieves near SoTA zero-shot cross-lingual transfer performance for NLP tasks. Following Hu et al. (2020), instead of using the top layer, we output the 12-th layer in XLM-R and mBERT. For vision-language tasks, we freeze layers below 9 as this setup empirically performs the best.

Our model employs a 2-layer Transformer with 4-head attention for the text and video transformer pooling (TP) modules. The embedding dimension D is set to 1024. We use the Adam (Kingma and Ba, 2015) optimizer and a 0.0002 learning rate to train our model for 16 (pre-training) and 10 (fine-tuning) epochs. The softmax temperature in all noise contrastive objectives is set to 0.1.

5.3 Experimental Setup

We use Multi-HowTo100M for multilingual multimodal pre-training (MMP). For each video, we randomly sample the start and end time to construct a video clip. For a video clip, we randomly sample one language type each time from 9 languages and use the consecutive ASR transcriptions that are closest in time to compose (text-video) pairs for training. For simplicity and speed purposes, we follow the training protocol of XLM-R to pre-train on a multilingual corpus *wihtout* using translation pairs, *i.e.*, we use multilingual text-video pairs (x, v) but no translation pairs from Multi-HowTo100M and utilize only inter- and intra-modal NCE (Eq. 1-3) for MMP.

We fine-tune our model on VTT, VATEX, and Multi30K to evaluate on text→video search tasks. In the zero-shot cross-lingual transfer experiments, we use only English-video data and fine-tune with Eq. 1-3. We then test the model with non-English queries. When annotations in additional languages are available (by humans in VATEX and Multi30K; by MT models (*i.e.*, *translate-train*) in VTT), we utilize all available multilingual annotations (*i.e.*, fully supervised) and iterate over all possible (x, v, y) pairs to train with Eq. 5-7 to demonstrate the strong performance target for evaluating zero-shot cross-lingual transfer on VTT and to compare fairly with other fully-supervised baselines in multilingual text→video search on VATEX and Multi30K. We report the standard recall at k (R@k) metrics (higher is better).

Text-B	Video-B	R@1↑	R@5↑	R@10↑
XLM-R	S3D	19.5	49.0	62.8
XLM-R	R(2+1)D	19.0	49.5	63.2
XLM-R	R+S	**21.0**	**50.6**	**63.6**
mBERT	R+S	19.9	49.8	62.5

Table 1: **Text and Video (B)ackbone comparison.**

T layers	V layers	R@1↑	R@5↑	R@10↑
1	1	20.0	50.3	63.2
2	1	20.1	50.5	63.8
2	2	**21.0**	**50.6**	63.6
2*	2*	20.7	50.5	63.3
4	4	20.8	50.4	**63.8**

Table 2: **Architecture comparison.** Number of multilingual multimodal transformer layers. *:Weight sharing between video and text transformers.

Objective	Inter	Intra	Cross	R@1↑	R@5↑	R@10↑
Triplet	✓			13.3	36.0	55.2
Triplet	✓	✓		20.9	49.3	63.0
NCE	✓			21.4	49.3	61.1
NCE	✓	✓		21.0	50.6	63.6
NCE*	✓	✓		21.3	50.7	63.5
NCE*	✓	✓	✓	**21.5**	**51.0**	**63.8**

Table 3: **Objective comparison.** *Training with additional machine translated *de*-video and *fr*-video pairs.

5.4 Comparison Experiments and Ablations

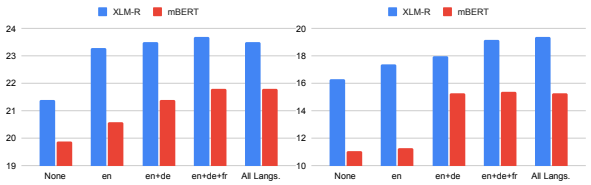
In this section, we ablate and compare different text/video encoders, Transformer model architectures, and learning objectives for English→video search on VTT.

Text and Video Encoders. Table 1 compares different text and video encoder backbones. For the visual encoders, while R(2+1)D outperforms S3D, the simple concatenation (*i.e.*, early-fusion) of their output features provides a $1.5 \sim 2.0$ improvement in R@1. For the text encoder, XLM-R significantly outperforms mBERT.

Transformer Pooling. Table 2 compares various configurations of the proposed Transformer pooling module. We observe that a simple 2-layer Transformer achieves the best performance. Weight

Model	*en*	*de*	*fr*	*cs*	*zh*	*ru*	*vi*	*sw*	*es*	Avg↑
mBERT	19.9	11.1	11.6	8.2	6.9	7.9	2.7	1.4	12.0	9.1
mBERT-MP	20.6	11.3	11.9	8.0	7.1	7.7	2.5	1.1	12.5	9.2
mBERT-MMP	21.8	15.0	15.8	11.2	8.4	11.0	3.7	3.4	15.1	11.7
XLM-R	21.0	16.3	17.4	16.0	14.9	15.4	7.7	5.7	17.3	14.7
XLM-R-MP	23.3	17.4	18.5	17.1	16.3	17.0	8.1	6.2	18.5	15.8
XLM-R-MMP	**23.8**	**19.4**	**20.7**	**19.3**	**18.2**	**19.1**	**8.2**	**8.4**	**20.4**	**17.5**
mBERT + translated VTT	19.6	18.2	18.0	16.9	16.2	16.5	8.4	13.0	18.5	16.1
mBERT-MMP + translated VTT	21.5	19.1	19.8	18.3	17.3	18.3	8.9	14.1	20.0	17.4
XLM-R + translated VTT	21.5	19.6	20.1	19.3	18.9	19.1	10.3	12.5	18.9	17.8
XLM-R-MMP + translated VTT	**23.1**	**21.1**	**21.8**	**20.7**	**20.0**	**20.5**	**10.9**	**14.4**	**21.9**	**19.4**

Table 4: **Recall@1 of multilingual text→video search on VTT.** Upper: Zero-shot cross-lingual transfer. Lower: Performance with synthesized pseudo-multilingual annotations for training. MMP: multilingual multimodal pre-training on Multi-HowTo100M. MP: Multimodal (English-Video) pre-training on HowTo100M.



(a) English→Video (b) Zero-shot German→Video

Figure 3: R@1 trends in languages used for multilingual multimodal pre-training. Left: English→video search. Right: Zero-shot German→video search.

sharing of the video and text Transformer slightly degrades the performance. Therefore, we choose to separate them.

Learning Objective. From Table 3, the intra-modal contrastive objective is important for both NCE and Triplet loss. In general, the NCE loss outperforms the Triplet loss. The proposed inter-modal and intra-modal NCE objective achieves the best performance. When captions in multiple languages are available, cross-lingual NCE additionally provides a consistent improvement.

5.5 VTT Zero-Shot Cross-Lingual Transfer

Table 4 shows the multilingual text→video search results on VTT. With the best English-video models at hand (with either mBERT or XLM-R as the text backbone), we first investigate how well these models transfer to other non-English languages under the zero-shot setting. We then analyze the benefit of the proposed multilingual multimodal pre-training.

The upper section shows the zero-shot results. Unlike cross-lingual transfer in NLP tasks, employing multilingual Transformers in vision-language tasks apparently does not generalize well across languages. For example, there is a significant drop in R@1 (19.9→11.1 (-44%) with mBERT, 21.0→16.3 (-24%) with XLM-R) when directly applying English-finetuned model to German→video search. For comparison, there is only a -10% degradation for XLM-R on $en \rightarrow de$ cross-lingual transfer in XNLI (Conneau et al., 2018). Multimodal (English-video) pre-training (MP) on HowTo100M only improves average R@1 (+0.1 or mBERT and +1.1 for XLM-R) compared to model-from-scratch. In contrast, our proposed multilingual multimodal pre-training (MMP) is shown to be the key facilitator for zero-shot cross-lingual transfer. MMP improves German→Video search (11.1→15.0, +35% for mBERT, and 16.3→19.4, +20% for XLM-R) and achieves 2.6 $\sim$ 2.8 improvement in average R@1. We attribute the effectiveness of MMP to learning improved alignments between multilingual textual and visual context in the shared embedding space, as relatively balanced improvements between English→video and non-English→video is observed with fine-tuning.

Fig. 3 demonstrates the trend of R@1 while incrementally incorporating additional languages for MMP. For XLM-R, the improvement in R@1 asymptotically converges when pre-training with more multilingual text-video pairs. On the other hand, for zero-shot German→video search, pre-training with more languages keeps improving the search performance, even though the additional language (*e.g.*, French) is different from the target language (*i.e.*, German).

The lower section of Table 4 shows the results of models fine-tuned with (synthesized) pseudo-multilingual annotations. It can be regarded as the *translate-train* scenario, which serves as a strong performance target for evaluating zero-shot cross-lingual transfer, as discussed in (Lample and Conneau, 2019; Hu et al., 2020). Both mBERT and XLM-R yield better performance across non-

Figure 4: Qualitative multilingual (*en*, *ru*, *vi*, *zh*) text→video search results on VTT.

English languages with the in-domain translated pseudo-multilingual annotations. However, for English→video search, a 0.7 degradation is observed compared to the zero-shot setting. It is likely due to the noise in the translated captions. Notably, there is still a performance gap between zero-shot and translate-train settings for models with mBERT. In contrast, the gap is much smaller for models with XLM-R. In the following sections, we refer `Ours-MMP` as our best model with XLM-R as the text backbone and compare it with other state-of-the-art methods.

Qualitative Results Fig. 4 shows the multilingual text→video search results with Ours-MMP (VTT:*en*-only) on VTT under the zero-shot setup. Note that only one shared English-finetuned model is used for text→video search in all languages. As demonstrated, the proposed model successfully retrieves the correct videos with English (*en*) and Russian (*ru*) queries. The other top-ranked videos also share similar visual appearance to the correct one. For zero-shot transferring of the English-finetuned model to distant languages such as Vietnamese (*vi*) and Chinese (*zh*), we observe that there is still limitation for our zero-shot models to understand abstract concepts (*e.g.*, "space project") and associate small objects (*e.g.*, "microphone") with the text queries in distant languages.

5.6 Comparison to Supervised State of the Art

English→Video Search on VTT. Table 5 shows the comparison of English→video models on VTT. For a fair comparison to other baselines, our model fine-tunes only with the original English annotations on VTT. The results show that our model outperforms other baselines by a large margin. Specifically, our model achieves 8.9 R@1 improvement over the original HowTo100M model (Miech et al., 2019) and other recent baselines with pre-training on HowTo100M. Using a smaller set of visual features and training on a smaller (6,513 vs 9,000) training set[2], our model also outperforms CE (Liu et al., 2019) with or without pre-training.

Model	R@1↑	R@5↑	R@10↑
JSFusion (Yu et al., 2018)	10.2	31.2	43.2
JPoSE (Wray et al., 2019)	14.3	38.1	53.0
VidTrans† (Korbar et al., 2020)	14.7	–	52.8
HT100M† (Miech et al., 2019)	14.9	40.2	52.8
Noise† (Amrani et al., 2020)	17.4	41.6	53.6
CE[2] (Liu et al., 2019)	20.9	48.8	62.4
Ours(VTT:*en*-only)	21.0	50.6	63.6
Ours-MMP (VTT:*en*-only)	**23.8**	**52.6**	**65.0**

Table 5: English→video search performance on VTT. †: Models with pre-training on HowTo100M.

	English to Video			Chinese to Video		
Model	R@1↑	R@5↑	R10↑	R@1↑	R@5↑	R@10↑
VSE (Kiros et al., 2014)	28.0	64.3	76.9	-	-	-
VSE++ (Faghri et al., 2018)	33.7	70.1	81.0	-	-	-
Dual (Dong et al., 2019)	31.1	67.4	78.9	-	-	-
HGR (Chen et al., 2020a)	35.1	73.5	83.5	-	-	-
Ours (VATEX:*en*-only)	43.5	79.8	88.1	23.9	55.1	67.8
Ours-MMP (VATEX:*en*-only)	**44.4**	80.5	88.7	29.7	63.2	75.5
Ours-MMP (VATEX:*en*, *zh*)	44.3	**80.7**	**88.9**	**40.5**	**76.4**	**85.9**

Table 6: Multilingual text→video search on VATEX.

Multilingual Text→Video Search on VATEX. Table 6 summarizes English→video and Chinese→video search performance on the VATEX dataset. Under the zero-shot setting where we train with only English-video pairs, our model already outperforms other baselines. However, a clear performance gap between English→video and Chinese→video search is observed, indicating that cross-lingual transfer to a distant language remains challenging even with XLM-R. With the proposed MMP, the gap is significantly closed by 5.8/8.1/7.7 in R@1/5/10. When in-domain human-annotated Chinese captions are available, the performance of our model can further be improved for both languages and our model yields new state-of-the-art performance.

[2]CE uses 9,000 videos (VTT training and part of exclusive testing set) for training, while other baselines and our model in Table 5 are trained on the official VTT training set which contains 6,513 videos.

Model	M30K # lang.	English to Image R@1↑	English to Image R@5↑	English to Image R10↑	German to Image R@1↑	German to Image R@5↑	German to Image R@10↑	Czech to Image R@1↑	Czech to Image R@5↑	Czech to Image R@10↑
OE (Vendrov et al., 2015)	2	25.8	56.5	67.8	21.0	48.5	60.4	-	-	-
VSE++ (Faghri et al., 2018)	2	39.6	69.1	79.8	31.3	62.2	70.9	-	-	-
Pivot (Gella et al., 2017)	2	26.2	56.4	68.4	22.5	49.3	61.7	-	-	-
FB-NMT (Huang et al., 2020a)	2	47.3	75.4	83.5	37.0	64.0	73.1	-	-	-
MULE (Kim et al., 2020)	4	42.2	72.2	81.8	35.1	64.6	75.3	37.5	64.6	74.8
SMALR (Burns et al., 2020)	10	41.8	72.4	82.1	36.9	65.4	75.4	36.7	68.0	78.2
MHA-D (Huang et al., 2019b)	2	50.1	78.1	85.7	40.3	70.1	79.0	-	-	-
Ours (M30K:en-only)	1	48.4	78.3	85.9	31.4	61.1	72.6	33.2	65.2	76.1
Ours-MMP (M30K:en-only)	1	50.0	79.2	86.8	33.8	63.3	74.7	37.9	68.8	78.2
Ours-MMP (M30K:en, de, cs, fr)	4	**51.6**	**80.1**	**87.3**	**45.1**	**75.6**	**85.0**	**46.6**	**75.9**	**83.4**

Table 7: Multilingual text→image search on Multi30K. MMP: Multilingual multimodal pre-training.

Cross-Modality Transfer to Multi30K: From Video-Text to Image-Text. To extend our study on zero-shot cross-lingual transfer for image-text tasks, we investigate the feasibility of transferring our video-text model across modalities. We replace the 3D-CNN in the original video-text model with a 2D-CNN to encode the image. In practice, following MHA-D (Huang et al., 2019b), we utilize the Faster-RCNN (Ren et al., 2015) pre-trained in Visual Genome (Krishna et al., 2016) to extract regional visual features. Essentially, an image is encoded as $e_v = \mathbb{R}^{M \times H}$ where $M = 36$ is the maximum number of visual objects in an image. For models with MMP, we initialize their weights with the model pre-trained on Multi-HowTo100M. To tackle the feature mismatch between 2D-CNN and 3D-CNN, we leverage a linear layer with a doubled learning rate to map 2D-CNN features to the same dimension as 3D-CNN features.

Table 7 shows the results on Multi30K. For zero-shot cross-lingual transfer, when trained from scratch (M30K:en-only), our model achieves comparable performance to MHA-D but lags in German→image search since it only uses English annotations. In Ours-MMP, pre-training improves all recall metrics even with modality gap. The average R@1 improvement is 3.2. A larger gain for (relatively) low-resource language such as Czech is observed. Without using any Czech annotations, our zero-shot model with MMP achieves comparable Czech→image search performance to SMALR (Burns et al., 2020), which uses 10 languages including Czech. However, when transferring across modalities and using only English annotations, there are performance gaps between English→Image and German/Czech→Image search, implying that transferring models across modalities is feasible but remains challenging. We consider zero-shot cross-modal cross-lingual transfer as our future work.

For a fair comparison with other baselines, when trained with annotations in all 4 languages provided by Multi30K, our model greatly outperforms all baselines by large margins in multilingual text→image search.

6 Conclusion

We have presented a multilingual multimodal pre-training (MMP) strategy, the Multi-HowTo100M dataset, and a Transformer-based text-video model for learning contextual multilingual multimodal representations. The results in this paper have convincingly demonstrated that MMP is an essential ingredient for zero-shot cross-lingual transfer of vision-language models. Meanwhile, there are many remaining challenges, such as resolving the performance gap between zero-shot and training with in-domain non-English annotations; as well as techniques to transfer varieties of vision-language models (*e.g.*, VQA (Goyal et al., 2017), TVQA (Lei et al., 2020)) or visually-enhanced NLP models such as unsupervised multimodal machine translation (Huang et al., 2020b). We believe the proposed methodology, and the corresponding resources we release, will be an important first step towards spurring more research in this direction.

Acknowledgments

This work is supported by the DARPA grants funded under the AIDA program (FA8750-18-2-0018) and the GAILA program (award HR00111990063) (P.Y.). This work is also supported by EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems [EP/L015897/1] (M.P.). The authors appreciate Prahal Arora, Shengxin Zha, Polina Kuznetsova, Xu Hu, and Geoffrey Zweig for their suggestions of this work. The authors would also like to thank the anonymous reviewers for their feedback.

References

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. 2016. Unsupervised learning from narrated instruction videos. In *Computer Vision and Pattern Recognition (CVPR)*.

Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. 2020. Noise estimation using density estimation for self-supervised multimodal learning. *arXiv preprint arXiv:2003.03186*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Vancouver, Canada. Association for Computational Linguistics.

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637, Online. Association for Computational Linguistics.

Jeremy Barnes, Lilja Øvrelid, and Erik Velldal. 2019. Sentiment analysis is not solved! assessing and probing sentiment classification. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 12–23, Florence, Italy. Association for Computational Linguistics.

Andrea Burns, Donghyun Kim, Derry Wijaya, Kate Saenko, and Bryan A. Plummer. 2020. Learning to scale multilingual representations for vision-language tasks. In *The European Conference on Computer Vision (ECCV)*.

David Chen and William Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA. Association for Computational Linguistics.

Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. 2020a. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. In *ICML*.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2018. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ryan Cotterell and Georg Heigold. 2017. Cross-lingual character-level neural morphological tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 748–759, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. 2019. Dual encoding for zero-example video retrieval. In *CVPR*.

Desmond Elliott, Stella Frank, Khalil Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*, pages 70–74. Association for Computational Linguistics.

Akiko Eriguchi, Yoshimasa Tsuruoka, and Kyunghyun Cho. 2017. Learning to parse and translate improves neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 72–78, Vancouver, Canada. Association for Computational Linguistics.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2018. Vse++: Improving visual-semantic embeddings with hard negatives. In *BMVC*.

Manaal Faruqui and Chris Dyer. 2014. Improving vector space word representations using multilingual correlation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 462–471, Gothenburg, Sweden. Association for Computational Linguistics.

Spandana Gella, Rico Sennrich, Frank Keller, and Mirella Lapata. 2017. Image pivoting for learning multilingual multimodal representations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2839–2845. Association for Computational Linguistics.

Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. 2019. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*.

Stephan Gouws and Anders Søgaard. 2015. Simple task-specific bilingual word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1386–1390, Denver, Colorado. Association for Computational Linguistics.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2017. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *CoRR*, abs/2003.11080.

Haoyang Huang, Yaobo Liang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, and Ming Zhou. 2019a. Unicoder: A universal language encoder by pre-training with multiple cross-lingual tasks. *arXiv preprint arXiv:1909.00964*.

Po-Yao Huang, Xiaojun Chang, and Alexander Hauptmann. 2019b. Multi-head attention with diversity for learning grounded multilingual multimodal representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1461–1467, Hong Kong, China. Association for Computational Linguistics.

Po-Yao Huang, Xiaojun Chang, Alexander Hauptmann, and Eduard Hovy. 2020a. Forward and backward multimodal nmt for improved monolingual and multilingual cross-modal retrieval. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, ICMR '20, page 53–62, New York, NY, USA. Association for Computing Machinery.

Po-Yao Huang, Junjie Hu, Xiaojun Chang, and Alexander Hauptmann. 2020b. Unsupervised multimodal neural machine translation with pseudo visual pivoting. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8226–8237, Online. Association for Computational Linguistics.

Po-Yao Huang, Guoliang Kang, Wenhe Liu, Xiaojun Chang, and Alexander G. Hauptmann. 2019c. Annotation efficient cross-modal retrieval with adversarial attentive alignment. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, page 1758–1767, New York, NY, USA. Association for Computing Machinery.

Po-Yao Huang, Vaibhav, Xiaojun Chang, and Alexander G. Hauptmann. 2019d. Improving what cross-modal retrieval models learn through object-oriented inter- and intra-modal attention networks. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, ICMR '19, page 244–252, New York, NY, USA. Association for Computing Machinery.

Vladimir Iashin and Esa Rahtu. 2020. Multi-modal dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 958–959.

Donghyun Kim, Kuniaki Saito, Kate Saenko, Stan Sclaroff, and Bryan A. Plummer. 2020. MULE: Multimodal Universal Language Embedding. In *AAAI Conference on Artificial Intelligence*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Ryan Kiros, Ruslan Salakhutdinov, and Richard S. Zemel. 2014. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539.

Alexandre Klementiev, Ivan Titov, and Binod Bhattarai. 2012. Inducing crosslingual distributed representations of words. In *Proceedings of COLING 2012*, pages 1459–1474.

Bruno Korbar, F. Petroni, Rohit Girdhar, and L. Torresani. 2020. Video understanding as machine translation. *ArXiv*, abs/2006.07203.

R. Krishna, Yuke Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, Stephanie Chen, Yannis Kalantidis, L. Li, D. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2016. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123:32–73.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 7059–7069. Curran Associates, Inc.

Jie Lei, Licheng Yu, Tamara Berg, and Mohit Bansal. 2020. TVQA+: Spatio-temporal grounding for video question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8211–8225, Online. Association for Computational Linguistics.

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. 2019. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 13–23.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nicholas Johnston, Andrew Rabinovich, and Kevin Murphy. 2015. What's cookin'? interpreting cooking videos using text, speech and vision. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 143–152, Denver, Colorado. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6294–6305. Curran Associates, Inc.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Takashi Miyazaki and Nobuyuki Shimizu. 2016. Cross-lingual image caption generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1780–1790. Association for Computational Linguistics.

Graham Neubig and Junjie Hu. 2018. Rapid adaptation of neural machine translation to new languages. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 875–880, Brussels, Belgium. Association for Computational Linguistics.

Aäron van den Oord, Y. Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748.

Mandela Patrick, Y. Asano, Ruth Fong, João F. Henriques, G. Zweig, and A. Vedaldi. 2020. Multi-modal self-supervision from generalized data transformations. *ArXiv*, abs/2003.04298.

Mandela Patrick, Po-Yao Huang, Yuki M. Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. 2021. Support-set bottlenecks for video-text representation learning. In *International Conference on Learning Representations (ICLR)*.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An Adapter-Based Framework for Multi-Task Cross-Lingual Transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7654–7673, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Afshin Rahimi, Yuan Li, and Trevor Cohn. 2019. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy. Association for Computational Linguistics.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In

Advances in neural information processing systems, pages 91–99.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Holger Schwenk and Xian Li. 2018. A corpus for multilingual document classification in eight languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Aditya Siddhant, Ankur Bapna, Yuan Cao, Orhan Firat, Mia Chen, Sneha Kudugunta, Naveen Arivazhagan, and Yonghui Wu. 2020. Leveraging monolingual data with self-supervision for multilingual neural machine translation. *arXiv preprint arXiv:2005.04816*.

Gunnar A. Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, João Carreira, Phil Blunsom, and Andrew Zisserman. 2020. Visual grounding in video for unsupervised word translation. In *CVPR*.

Dídac Surís, Dave Epstein, and Carl Vondrick. 2020. Globetrotter: Unsupervised multilingual translation from visual alignment. *arXiv preprint arXiv:2012.04631*.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 477–487, Montréal, Canada. Association for Computational Linguistics.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2015. Order-embeddings of images and language. *arXiv preprint arXiv:1511.06361*.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4581–4591.

Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. 2019. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*.

Shijie Wu and Mark Dredze. 2019a. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

Shijie Wu and Mark Dredze. 2019b. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A large video description dataset for bridging video and language. In *CVPR*.

I. Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. 2019. Billion-scale semi-supervised learning for image classification.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. 2014. Instructional videos for unsupervised harvesting and learning of action examples. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 825–828, New York, NY, USA. Association for Computing Machinery.

Youngjae Yu, Jongseok Kim, and Gunhee Kim. 2018. A joint sequence fusion model for video question answering and retrieval. In *ECCV*.

Zhuosheng Zhang, Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Zuchao Li, and Hai Zhao. 2020. Neural machine translation with universal visual representation. In *International Conference on Learning Representations*.

A Appendix Overview

The Appendix is organized as follows: First we provide details about the Multilingual HowTo100M (Multi-HowTo100M) dataset for multilingual multimodal pre-training (MMP) in §B. Then we provide additional implementation details and experiment setup in §C. Additional ablation studies regarding choices of Transformer architecture are discussed in §D. Then we present additional cross-dataset transfer experiments in §E.

B The Multilingual HowTo100M Dataset

In this section we provide the detailed statistics of the Multilingual HowTo100M (Multi-HowTo100M) dataset. We also provide a comparison to Sigurdsson et al. (2020) that also uses HowTo100M for unsupervised word translation.

The Multi-HowTo100M dataset is built upon the original English HowTo100M dataset (Miech et al., 2019) that contains 1.2 million instructional videos (138 million clips) on YouTube. We reuse the *raw* English subtitles in HowTo100M, where the subtitles in HowTo100M are either automatic speech recognition (ASR) transcriptions or user generated subtitles.

For Multi-HowTo100M, we use the same video collection as English HowTo100M. At the time of data collection (May 2020), there were 1.09 million videos accessible. We collect the subtitles provided by YouTube, which either consist of user-generated subtitles or those generated by Google ASR and Translate in the absence of user-generated ones. Essentially, we collect video subtitles in 9 languages: English (*en*), German (*de*), French (*fr*), Russian (*ru*), Spanish (*es*), Czech (*cz*), Swahili (*sw*), Chinese (*zh*), Vietnamese (*vi*). Table 8 summarizes the dataset statistics for each language. In most cases there are more than 1 billion tokens a language.

Fig. 5 further shows the number of tokens per video. There are typically lengthy narrations that contains several hundreds of tokens available in each instructional video. Fig. 6 shows the distribution of number of tokens in a subtitle. For each subtitle segment, which ranges from 0∼20 seconds, there are typically 15∼25 words. The most of the cases, subtitles are well aligned in time for non-English languages. Fig. 2 visualizes a few examples in Multi-HowTo100M.

A similar HowTo100M variant has been recently reported in MUVE (Sigurdsson et al., 2020) that is created for unsupervised word translation.

Language	videos	#subtitle	#tokens
English	1238911	138429877	1.18B
German	1092947	69317890	1.26B
French	1093070	69399097	1.33B
Czech	1092717	68911940	1.22B
Russian	1092802	69117193	1.25B
Chinese	1092915	68939488	0.94B
Swahili	1092302	68898800	1.22B
Vietnamese	1092603	68887868	1.13B
Spanish	1092649	70143503	1.16B

Table 8: Multi-HowTo100M statistics

Our Multi-HowTo100M differs from MUVE in the following perspectives: First, we collects 9 language for *all* videos in HowTo100M while MUVE only has 4 languages available (English, French, Japanese, and Korean) on HowTo100M. Also, MUVE divided HowTo100M into 4 non-overlapped sections for each language, there are no parallel pairs for each subtitle. While in Multi-HowTo100M, there are 7-9 languages for each subtitle. Essentially, There are more than 1 billion tokens in most languages in Multi-HowTo100M. To our best knowledge, our Multi-HowTo100M dataset is currently the largest multilingual text-video collection.

Beyond scale, instructional videos in Multi-HowTo100M are feasible pre-training resources for many downstream vision-language models. Demonstrators in instructional videos typically perform intentionally and explain the visual object or action explicitly. According to the inspection by (Miech et al., 2019), for around 51% of clips, at least one object or action mention in the caption can be visually seen. Prior work has shown that instructional videos are useful for event recognition (Yu et al., 2014), action localization model (Alayrac et al., 2016), cross-modal alignments (Malmaud et al., 2015). We expect the previous success in the intersection of natural language processing (NLP) and computer vision (CV) could be further translated into more languages to have a broaden impact.

The are great potentials of using our Multi-HowTo100M dataset in related research field such as multilingual multimodal representation learning (Huang et al., 2019b; Kim et al., 2020; Burns et al., 2020), multilingual multimodal translation (Huang et al., 2020b; Surís et al., 2020), multilingual image/video captioning (Miyazaki and Shimizu, 2016) ... etc. We expect the release of

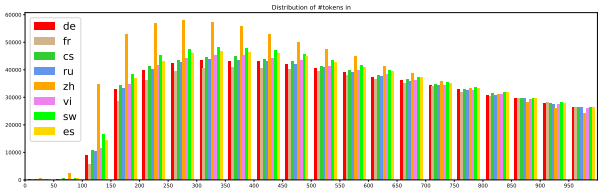


Figure 5: Distribution of #tokens/video in Multi-HowTo100M

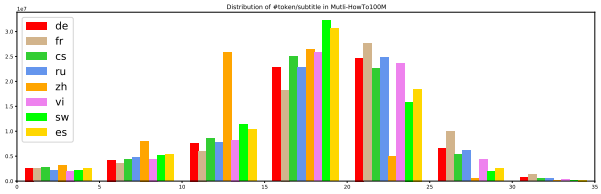


Figure 6: Distribution of #tokens/subtitle in Multi-HowTo100M

Multi-HowTo100M will be a first step towards spurring more research in these directions.

C Implementation and Experiment Details

Pre-Processing. For pre-possessing, we truncate the maximum length N of text to 192 for pre-training on Multi-HowTo100M. The maximum length is set to 96 for fine-tuning VTT (Xu et al., 2016), VATEX (Wang et al., 2019) and Multi30K (Elliott et al., 2016). The maximum video length M is set to 128 for pre-training on Multi-HowTo100M and 36 for all fine-tuning tasks.

Model Architecture. For the multilingual Transformers, either multilingual BERT (Devlin et al., 2019) or XLM-R-large (Artetxe et al., 2020), we use the pre-trained version provided by HuggingFace. [3] and use their corresponding tokenizers for tokenization. Detailed design choices regarding output layer and frozen layer is discussed in §D.

For the video backbone, we use a 34-layer, R(2+1)-D (Tran et al., 2018) network pre-trained on IG65M (Ghadiyaram et al., 2019) and a S3D (Miech et al., 2020) network pre-trained on HowTo100M (Miech et al., 2019). We apply a spatial-temporal average pooling over the last convolutional layer, resulting in a 512-dimensional vector for each 3D CNN network. We extract visual features at a rate of 1 feature per second. Since the 3D CNNs employs different size of input windows (*e.g.*, 8 frames for R(2+1)D and 16 for S3D), we re-sample videos to 30 fps and employs a window of size 8 or 30 that takes consecutive frames starting from the beginning of every second for encoding. We simply concatenate the two 3D-CNN outputs and use the 1024-dimension vector as the visual input stream to our model. Notably, instead of using 9 different types of visual features as in CE (Liu et al., 2019), we use only the above 2 features and achieve superior performance.

For the Transformer pooling head (TP) modules, we use a 2-layer Transformer with 4-head attention for each TP. The embedding dimension D is set to 1024. We do not use the positional embeddings in both text and video TP as we do not find them beneficial in our experiments. The softmax temperature in all NCE contrastive objectives is set to 0.1 as used in SimCLR (Chen et al., 2020b).

Note that unlike ViLBERT (Lu et al., 2019) or OAN (Huang et al., 2019d), our models does not employ cross-modality attention and keep the multi-head self-attention within the same modality. The main reason is to reduce the inference time complexity. For cross-modality attention, the complexity is $O(TV)$ to encode T text queries for V videos in a dataset before retrieval (since video and query representations depend on each other). It is clearly not scalable when the dataset contains millions of videos. To this end, our model keep self-attention within the same modality which results in a $O(T+V)$ complexity compared $O(TV)$ in prior work with cross-modality attention. In our preliminary experiments, we also incorporate cross-modality attention and achieved 0.3∼1.8 R@1 improvement. Considering the trade-off between performance and scalability, we choose the latter.

Training and Inference Details and Profiling. For the softmax temperature in NCE, we set to 0.1 as used in SimCLR (Chen et al., 2020b). We use the Adam (Kingma and Ba, 2015) optimizer with a initial learning rate $2 \cdot 10^{-4}$ and clip gradients greater than 0.2 during the training phase. Dropout rate is 0.3. Since the video length and token length is longer in the pre-training phase, we use a 64 batch size for pre-training. For fine-tuning, we use a batch size of 128.

Pre-training on the 1.2 million HowTo100M videos takes around 10 GPU hours (NVIDA V100) for 16 epochs. We speed up the pre-training process by distributing the workload over 8 GPUs on a single node of our server. We use 1 GPU for the fine-tuning or training from scratch experiments.

[3] https://github.com/huggingface/transformers

For the MSR-VTT split, it takes 12 GPU hours to train our model on 180K video-text pairs for 20 epochs. For VATEX, it takes 32 GPU hours to train on 260K video-text pairs for 30 epochs. For inference, the encoding speed is around 250-300 videos/sec and 200-250 text queries/sec. The overall text→video search speed on 1,000 video-text pairs (1,000 text queries over 1,000 videos) is around 6 seconds including video/text encoding and ranking their similarity scores.

Experiment Details. Our experiment consider three types of pre-training: (1) Multilingual multimodal pre-training (MMP), (2) Multimodal pre-training (MP), and (3) no pre-training (from scratch). For (1) and (2), we pre-train 16 epochs and use the model weight at 16-th epoch for fine-tuning experiments.

For multimodal pre-training, we pre-train on the original English HowTo100M dataset. We iterate over all videos in HowTo100M. For each video, we randomly sample the start and end time to construct a video clip. For each clip, we locate the nearest consecutive ASR transcriptions in time and use it as to construct the (video, text) pair for training.

For multilingual multimodal pre-training (MMP), we use Multi-HowTo100M for pre-training. For each video, we follow the same strategy as MP. For a clip, we sample one language type each time from 9 languages and use the consecutive ASR transcriptions that are closest in time to compose (video, text) pairs for training.

After pre-training, we fine-tune our model on VTT and VATEX to evaluate on text→video search tasks. In the zero-shot cross-lingual transfer experiments, we use only English-video data. We then directly test the model with non-English queries to report the zero-shot performance. When annotations in additional languages are available (by humans in VATEX and Multi30K; by MT models (*i.e. translate-train*) in VTT), we train our model with all available multilingual annotations (*i.e.* fully supervised) to compare fairly with other baselines in multilingual text→video search.

Since pre-trained model has a faster convergence rate, we fine-tune for 10 epochs and use the model with best validation performance (summation of R@1, R@5, R@10) for testing. For models without pre-training (*i.e.*, from-scratch), we train for 20 epochs under the same training protocol.

Output layer	Freeze lower	*en*	*de*
3	0	20.9	3.2
6	0	20.5	3.1
9	0	21.0	4.8
12	0	21.0	13.3
15	0	20.5	12.3
18	0	20.8	12.6
12	6	21.0	15.5
12	9	**21.0**	**16.3**
12	12	18.9	14.1

Table 9: Text→video R@1 of XLM-R output layers and layers to freeze on VTT

Output layer	Freeze lower	*en*	*de*
3	0	19.2	2.5
6	0	19.5	2.0
9	0	19.3	5.8
12	0	19.6	8.8
12	6	19.3	10.5
12	9	**19.9**	**11.1**
12	12	18.9	9.8

Table 10: Text→video R@1 of mBERT output layers and layers to freeze on VTT

D Additional Ablation Studies

As has been investigated in XTREME (Hu et al., 2020), choosing different output layers will affect the zero-shot transferability of multilingual Transformers in various NLP tasks. For text→video search tasks, we conduct a series of experiments to identify the desirable choices of hyper-parameters in the proposed multilingual multimodal Transformer that lead to best performance in English-to-video and (zero-shot) non-English-to-video search performance. Beyond our ablation studies in Sec. 5, in this part we highlight our trials in the choice of the output layer and the layers to be frozen in our multilingual Transformer backbone (*i.e.*, mBERT and XLM-R). There are 24 layers in XLM-R (large) and 12 layers in mBERT. We perform grid-search on VTT to identify the best choice of these two hyper-parameters.

Choice of Output Layers Table 9 and Table 10 compare different choices of output layer and layers to freeze in multilingual Transformers. Our results suggest that the best output layer for mBERT and XLM-R is the 12-th layer. Surprisingly, while output layer does not affect English→video search significantly, it greatly affects the zero-shot cross-lingual transfer performance of video-text models. For both XLM-R and mBERT, the performance degrade significantly if fine-tuning all layers.

text→video	English	Non-English
In-domain	✓	✓
Out-of-domain	✓	

Table 11: Coverage of our experiments

Choice of Layers to Freeze Similar to output layers, the choice of frozen layers greatly affects cross-lingual transferability. For both mBERT and XLM-R, it is desirable to freeze part of the lower layers and make the top-3 layers trainable for video-text models. We observe that when freezing all layers (*i.e.*, using the pre-extracted contextual multilingual embeddings) does not lead to satisfactory results. For mBERT, R@1 drops from 19.9 to 18.9 in English→video search and 11.1 to 9.8 in German→video search. For XLM-R, R@1 drops from 21.0 to 18.9 in English→video search and 16.3 to 14.1 in German→video search. These results imply that text-only contextual multilingual embeddings along are likely to be infeasible to be applied to vision-language tasks without proper fine-tuning.

An important observation is that the best English→video search performance corresponds to the best German→video performance. This trend implies that for model selection, the configuration for the best English→video model usually translates to the best configuration for (zero-shot) cross-lingual model. This shared trend justifies the English→video ablation studies in the original paper. Note that we utilize the best English→video for all (zero-shot) cross-lingual experiment in our experiment section.

For multilingual text→video search, the best configuration we found in our experiments is to output the 12-th layer and freeze the layers below 9 for both mBERT and XLM-R.

E Additional Experimental Results

The coverage of our text→video search experiments is summarized in Table 11. Our experiments cover the following scenarios:
In-domain, English: Table 5 (VTT) and Table 6 (VATEX) in the original paper.
In-domain, non-English: Table 4 (VTT, 9 languages) and Table 6 (VATEX, Chinese).
Out-of-domain, English: Additional (zero-shot) generalization results across datasets are in §E.1.
Out-of-domain, non-English: We consider this as our future work.

Model	R@1	R@5	R@10
VSE (Kiros et al., 2014)	10.1	29.4	41.5
VSE++ (Faghri et al., 2018)	14.4	35.7	46.9
Dual (Dong et al., 2019)	13.7	36.1	48.2
HGR (Chen et al., 2020a)	16.4	38.3	49.8
Ours-Full	**24.0**	**50.5**	**62.1**

Table 12: Zero-shot generalization on YouTube2Text with VTT-finetuned model.

E.1 Generalizability across English-Video Datasets

In this section. we provide additional experiment results regarding zero-shot generalization of the VTT-finetuned model on out-of-domain dataset. Specifically, we test on YouTube2Text (Chen and Dolan, 2011). The aim of this experiment is to test the cross-dataset generalizabilty of our model without using domain-specific training data.

Table 12 shows the comparison of English→video search results on the YouTube2Text testing set. Models in this table are only fine-tuned on VTT and use *no* YouTube2Text training data. As can be observed, our model with MMP generalizes well on YouTube2Text, outperforming HGR (Chen et al., 2020a) by 7.6 and DualEncoder (Dong et al., 2019) by 10.3 in R@1.

6

Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers

This work is currently under review and its preprint is public on arXiv.

Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers

Mandela Patrick[*]
Facebook AI
mandelapatrick@fb.com

Dylan Campbell[*]
University of Oxford
dylan@robots.ox.ac.uk

Yuki Asano[*]
University of Oxford
yuki@robots.ox.ac.uk

Ishan Misra
Facebook AI
imisra@fb.com

Florian Metze
Facebook AI
fmetze@fb.com

Christoph Feichtenhofer
Facebook AI
feichtenhofer@fb.com

Andrea Vedaldi
Facebook AI
vedaldi@fb.com

João F. Henriques
University of Oxford
joao@robots.ox.ac.uk

Abstract

In video transformers, the time dimension is often treated in the same way as the two spatial dimensions. However, in a scene where objects or the camera may move, a physical point imaged at one location in frame t may be entirely unrelated to what is found at that location in frame $t + k$. These temporal correspondences should be modeled to facilitate learning about dynamic scenes. To this end, we propose a new drop-in block for video transformers—*trajectory attention*—that aggregates information along implicitly determined motion paths. We additionally propose a new method to address the quadratic dependence of computation and memory on the input size, which is particularly important for high resolution or long videos. While these ideas are useful in a range of settings, we apply them to the specific task of video action recognition with a transformer model and obtain state-of-the-art results on the Kinetics, Something–Something V2, and Epic-Kitchens datasets. Code and models are available at: `https://github.com/facebookresearch/Motionformer`.

1 Introduction

Transformers [76] have become a popular architecture across NLP [32], vision [20] and speech [4]. The self-attention mechanism in the transformer works well for different types of data and across domains. However, its generic nature and its lack of inductive biases also mean that transformers typically require extremely large amounts of data for training [57, 8], or aggressive domain-specific augmentations [72]. This is particularly true for video data, for which transformers are also applicable [51], but where statistical inefficiencies are exacerbated. While videos carry rich temporal information, they can also contain redundant spatial information from neighboring frames. Vanilla self-attention applied to videos compares pairs of image patches extracted at all possible spatial locations and frames. This can lead it to focus on the redundant spatial information rather than the temporal information, as we show by comparing normalization strategies in our experiments.

We therefore contribute a variant of self-attention, called *trajectory attention*, which is better able to characterize the temporal information contained in videos. For the analysis of still images,

[*] Equal contribution.

Preprint. Under review.

spatial locality is perhaps the most important inductive bias, motivating the design of convolutional networks [41] and the use of spatial encodings in vision transformers [20]. This is a direct consequence of the local structure of the physical world: points that belong to the same 3D object tend to project to pixels that are close to each other in the image. By studying the correlation of nearby pixels, we can thus learn about the objects.

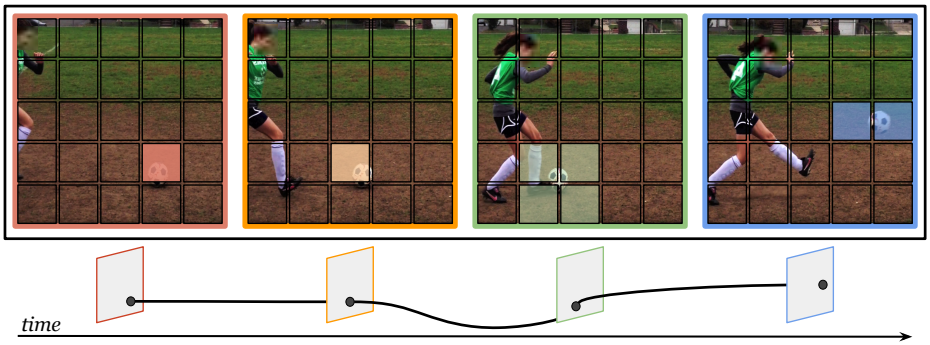


Figure 1: **Trajectory attention.** In this sequence of frames from the Kinetics-400 dataset, depicting the action 'kicking soccer ball', the ball does not remain stationary with respect to the camera, but instead moves to different locations in each frame. Trajectory attention aims to share information along the motion path of the ball, a more natural inductive bias for video data than pooling axially along the temporal dimension or over the entire space-time feature volume. This allows the network to aggregate information from multiple views of the ball, to reason about its motion characteristics, and to be less sensitive to camera motion.

Videos are similar, except that 3D points *move* over time, thus projecting on different parts of the image along certain 2D *trajectories*. Existing video transformer methods [7, 2, 51] disregard these trajectories, pooling information over the entire 3D space-time feature volume [2, 51], or pooling axially across the temporal dimension [7]. We contend that pooling along motion trajectories would provide a more natural inductive bias for video data, allowing the network to aggregate information from multiple views of the same object or region, to reason about how the object or region is moving (for example, the linear and angular velocities), and to be invariant to camera motion.

We leverage attention itself as a mechanism to find these trajectories. This is inspired by methods such as RAFT [71], which showed that excellent estimates of optical flow can be obtained from the correlation volume obtained by comparing local features across space and time. We observe that the joint attention mechanism for video transformers computes such a correlation volume as an intermediate result. However, subsequent processing collapses the volume without consideration for its particular structure. In this work, we seek instead to use the correlation volume to guide the network to pool information along motion paths.

We also note that visual transformers operate on image patches which, differently from individual pixels, cannot be assumed to correspond to individual 3D points and thus to move along simple 1D trajectories. For example, in Figure 1, depicting the action 'kicking soccer ball', the ball spans up to four patches, depending on the specific video frame. Furthermore, these patches contain a mix of foreground (the ball) and background objects, thus at least two distinct motions. Fortunately, we are not forced to select a single putative motion: the attention mechanism allows us to assemble a motion feature from all relevant 'ball regions'.

Inspired by Nyströmformer [85], we also propose a principled approximation to self-attention, *Orthoformer*. Our approximation sets state-of-the-art performance on the recent Long Range Arena (LRA) benchmark [70] for evaluating efficient attention approximations and generalizes beyond the video domain to long text and high resolution images, with lower FLOPS and memory requirements compared to alternatives, Nyströmformer and Performer [14]. Combining our approximation with trajectory attention allows us to significantly improve its computational and memory efficiency. With our contributions, we set state-of-the-art results on four video action recognition benchmarks.

2 Related Work

Video representations and 3D-CNNs. Hand-crafted features were originally used to convert video data into a representation amenable to analysis by a shallow linear model. Such representations include SIFT-3D [61], HOG3D [38], and IDT [77]. Since the breakthrough of AlexNet [39] on the ImageNet classification benchmark [59], which demonstrated the empirical benefits of deep neural networks to learn representations end-to-end, there have been many attempts to do the same for video. Architectures with 3D convolutions—3D-CNNs—were originally proposed to learn deep video representations [73]. Since then, improvements to this paradigm include the use of ImageNet-inflated weights [10], the space-time decomposition of 3D convolutions [55, 75, 84], channel-separated convolutions [74], non-local blocks [80], and attention layers [12].

Vision transformers. The transformer architecture [76], originally proposed for natural language processing, has recently gained traction in the computer vision domain. The vision transformer (ViT) [20] decomposes an image into a sequence of 16×16 words and uses a multi-layer transformer to perform image classification. To improve ViT's data efficiency, DeiT [72] used distillation from a strong teacher model and aggressive data augmentation. Transformers have also been used in a variety of vision image tasks, such as image representation learning [11, 83, 18, 60], image generation [52], object detection [47, 9], few-shot learning [19], and image–text representation learning [49, 63, 68, 43, 69]. and video-text [65, 64, 87, 26, 54, 1, 5], and video-audio [42, 53, 29] representation learning. While the use of transformer architectures for video is still in its infancy, concurrent works [7, 2, 51, 22] have already demonstrated that this is a highly promising direction. However, these approaches do not have a mechanism for reasoning about motion paths, treating time as just another dimension, unlike our approach.

Efficient attention. Due to the quadratic complexity of self-attention, there has been a significant amount of research on how to reduce its computational complexity with respect to time and memory use. Sparse attention mechanisms [13] were used to reduce self-attention complexity to $\mathcal{O}(n\sqrt{n})$, and locality-sensitivity hashing was used by Reformer [37] to further reduce this to $\mathcal{O}(n \log n)$. More recently, linear attention mechanisms have been introduced, namely Longformer [6], Linformer [79], Performer [14] and Nyströmformer [85]. The Long Range Arena benchmark [70] was recently introduced to compare these different attention mechanisms.

Temporal correspondences and optical flow. There are many approaches that aim to establish explicit correspondences between video frames as a way to reason about camera and object motion. For short-range correspondences across time, optical flow algorithms [30, 66, 71] are highly effective. In particular, RAFT [71] showed the effectiveness of an all-pairs inter-frame correlation volume as an encoding, which is essentially an attention map. All-pairs intra-frame correlations were subsequently shown to help resolve correspondence ambiguities [34]. For longer-range correspondences, object tracking by repeated detection [58] and data association can be used. In contrast to these approaches, our work does not explicitly establish temporal correspondences, but facilitates implicit correspondence learning via trajectory attention. Jabri et al. [31] estimate correspondences in a similar way, framing the problem as a contrastive random walk on a graph and apply explicit guidance via a cycle consistency loss. Incorporating such guidance into a video transformer is an interesting direction.

3 Trajectory Attention for Video Data

Our goal is to modify the attention mechanism in transformers to better capture the information contained in videos. Consider an input video $I \in \mathbb{R}^{T' \times 3 \times H \times W}$ consisting of T' frames of resolution $H \times W$. As in existing video transformer models [7, 2], we pre-process the video into a sequence of ST tokens $\mathbf{x}_{st} \in \mathbb{R}^D$, for a spatial resolution of S and a temporal resolution of T. We use a cuboid embedding [2, 22], where disjoint spatio-temporal cubes from the input volume are linearly projected to $\mathbb{R}^D$ (equivalent to a 3D convolution with downsampling). We also test an embedding of disjoint image patches [20]. A learnable positional encoding $\mathbf{e} \in \mathbb{R}^D$ is added to the video embeddings for spatial and temporal dimensions separately, resulting in the code $\mathbf{z}_{st} = \mathbf{x}_{st} + \mathbf{e}^s_s + \mathbf{e}^t_t$. Finally, a learnable classification token $\mathbf{z}_{\text{cls}}$ is added to the sequence of tokens, like in the BERT Transformer [32], to reason about the video as a whole. For clarity, we elide the classification token from our treatment in the sequel.

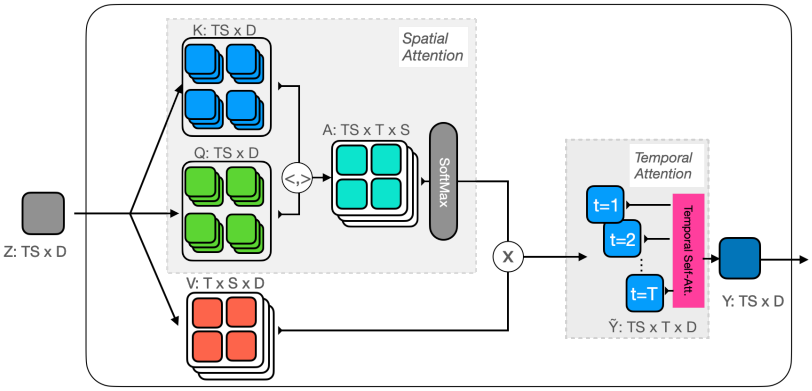


Figure 2: **Trajectory attention flowchart.** We divide the attention operation into two stages: the first forming a set of ST trajectory tokens for every space-time location st—a spatial attention operation between pairs of frames—and the second pooling along these trajectories with a 1D temporal attention operation. In this way, we accumulate information along the motion paths of objects in the video. The softmax operations are computed over the last dimension.

We now have a set of tokens that form the input to a sequence of transformer layers that, as in ViT [20], consist of Layer Norm (LN) operations [3], multi-head attention (MHA) [76], residual connections [28], and a feed-forward network (MLP):

$$\mathbf{y} = \text{MHA}(\text{LN}(\mathbf{z})) + \mathbf{z}; \quad \mathbf{z}' = \text{MLP}(\text{LN}(\mathbf{y})) + \mathbf{y}. \tag{1}$$

In the next section, we shall focus on a single head of the attention operation, and demonstrate how self-attention can realize a suitable inductive bias for video data. For clarity of exposition, we abuse the notation slightly, neglecting the layer norm operation and using the same dimensions for single-head attention as for multi-head attention.

3.1 Video self-attention

The self-attention operation begins by forming a set of query-key-value vectors $\mathbf{q}_{st}, \mathbf{k}_{st}, \mathbf{v}_{st} \in \mathbb{R}^D$, one for each space-time location st in the video. These are computed as linear projections of the input $\mathbf{z}_{st}$, that is, $\mathbf{q}_{st} = \mathbf{W}_q \mathbf{z}_{st}$, $\mathbf{k}_{st} = \mathbf{W}_k \mathbf{z}_{st}$, and $\mathbf{v}_{st} = \mathbf{W}_v \mathbf{z}_{st}$, for projection matrices $\mathbf{W}_i \in \mathbb{R}^{D \times D}$. A direct application of attention across space-time (called *joint space-time attention* [7, 2]) computes:

$$\mathbf{y}_{st} = \sum_{s't'} \mathbf{v}_{s't'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't'} \rangle}{\sum_{\bar{s}\bar{t}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}\bar{t}} \rangle}. \tag{2}$$

In this way, each query $\mathbf{q}_{st}$ is compared to all keys $\mathbf{k}_{s't'}$ using dot products, the results are normalized using the softmax operator, and the weights thus obtained are used to average the values corresponding to the keys. Compared to a standard transformer, we have omitted for brevity the softmax temperature parameter $D^{1/2}$ and instead assume that the queries and keys have been divided by $D^{1/4}$.

One issue with this formulation is that it has quadratic complexity in both space and time, i.e., $\mathcal{O}(S^2T^2)$. An alternative is to restrict attention to either space or time (called *divided space-time attention*):

$$\mathbf{y}_{st} = \sum_{s'} \mathbf{v}_{s't} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't} \rangle}{\sum_{\bar{s}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}t} \rangle} \text{ (space)}; \quad \mathbf{y}_{st} = \sum_{t'} \mathbf{v}_{st'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{st'} \rangle}{\sum_{\bar{t}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{s\bar{t}} \rangle} \text{ (time)}. \tag{3}$$

This reduces the complexity to $\mathcal{O}(S^2T)$ and $\mathcal{O}(ST^2)$, respectively, but only allows the model to analyse time and space independently. This is usually addressed by interleaving [7] or stacking [2] the two attention modules in a sequence.

Different to both of these approaches, we perform attention *along trajectories*, the probabilistic path of a token between frames.[2] For each space-time location st (the trajectory 'reference point') and corresponding query $\mathbf{q}_{st}$, we construct a set of trajectory tokens $\tilde{\mathbf{y}}_{stt'}$, representing the pooled

[2]Here, we refer to the trajectory as the motion between pairs of frames, rather than a multi-frame path.

information weighted by the trajectory probability. The trajectory extends for the duration of the video sequence and its tokens $\tilde{\mathbf{y}}_{stt'} \in \mathbb{R}^D$ at different times t' are given by:

$$\tilde{\mathbf{y}}_{stt'} = \sum_{s'} \mathbf{v}_{s't'} \cdot \frac{\exp\langle \mathbf{q}_{st}, \mathbf{k}_{s't'} \rangle}{\sum_{\bar{s}} \exp\langle \mathbf{q}_{st}, \mathbf{k}_{\bar{s}t'} \rangle}. \tag{4}$$

Note that the attention in this formula is applied spatially (index s) and independently for each frame. Intuitively, this pooling operation implicitly seeks the location of the trajectory at time t' by comparing the trajectory query $\mathbf{q}_{st}$ to the keys $\mathbf{k}_{s't'}$ at time t'.

Once the trajectories are computed, we further pool them across time to reason about intra-frame information/connections. To do so, the trajectory tokens are projected to a new set of queries, keys and values as usual:

$$\tilde{\mathbf{q}}_{st} = \tilde{\mathbf{W}}_q \, \tilde{\mathbf{y}}_{stt}, \quad \tilde{\mathbf{k}}_{stt'} = \tilde{\mathbf{W}}_k \, \tilde{\mathbf{y}}_{stt'}, \quad \tilde{\mathbf{v}}_{stt'} = \tilde{\mathbf{W}}_v \, \tilde{\mathbf{y}}_{stt'}. \tag{5}$$

Like $\mathbf{q}_{st}$ before, the updated reference query $\tilde{\mathbf{q}}_{st}$ corresponds to the trajectory reference point st and contains information spatially-pooled from across the reference frame t. This new query is used to pool across the new time (trajectory) dimension by applying 1D attention:

$$\mathbf{y}_{st} = \sum_{t'} \tilde{\mathbf{v}}_{stt'} \cdot \frac{\exp\langle \tilde{\mathbf{q}}_{st}, \tilde{\mathbf{k}}_{stt'} \rangle}{\sum_{\bar{t}} \exp\langle \tilde{\mathbf{q}}_{st}, \tilde{\mathbf{k}}_{st\bar{t}} \rangle}. \tag{6}$$

Like joint space-time attention, our approach has quadratic complexity in both space and time, $\mathcal{O}(S^2T^2)$, so has no computational advantage and is in fact slower than divided space-time attention. However, we demonstrate better accuracy than both joint and divided space-time attention mechanisms. We also provide fast approximations in Section 3.2. A flowchart of the full trajectory attention operation is shown in tensor form in Figure 2.

3.2 Approximating attention

To complement our trajectory attention, we also propose an approximation scheme to speed up calculations. This scheme is generic and applies to any attention-like pooling mechanism. We thus switch to a generic transformer-like notation to describe it. Namely, consider query-key-value matrices $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{D \times N}$ such that the query-key-value vectors are stored as columns $\mathbf{q}_i, \mathbf{k}_i, \mathbf{v}_i \in \mathbb{R}^D$ in these matrices.

In order to obtain an efficient decomposition of the attention operator, we will rewrite it using a probabilistic formulation. Let $A_{ij} \in \{0, 1\}$ be a categorical random variable indicating whether the jth input (with key vector $\mathbf{k}_j \in \mathbb{R}^D$) is assigned to the ith output (with query vector $\mathbf{q}_i \in \mathbb{R}^D$), with $\sum_j A_{ij} = \mathbf{1}$. The attention operator uses a parametric model of the probability of this event based on the multinomial logistic function, i.e., the softmax operator $\mathcal{S}(\cdot)$:[3]

$$P(A_{i:}) = \mathcal{S}(\mathbf{q}_i^\mathsf{T} \mathbf{K}), \tag{7}$$

where the subscript : denotes a full slice of the input tensor in that dimension. We now introduce the latent variables $U_{\ell j} \in \{0, 1\}$, which similarly indicate whether the jth input is assigned to the ℓth *prototype*, an auxiliary vector which we denote by $\mathbf{p}_\ell \in \mathbb{R}^D$. We can use the laws of total and conditional probability to obtain:

$$P(A_{ij}) = \sum\nolimits_\ell P(A_{ij} \mid U_{\ell j}) P(U_{\ell j}). \tag{8}$$

Note that the latent variables that we chose are independent of the inputs (keys). They use the same parametric model, but with parameters $\mathbf{P} \in \mathbb{R}^{D \times R}$ (the concatenated prototype vectors $\mathbf{p}_\ell$): $P(U) = \mathcal{S}(\mathbf{P}^\mathsf{T} \mathbf{K})$. Eq. 8 is *exact*, even under the parametric model for $P(U)$, though the corresponding true distribution $P(A \mid U)$ is intractable. We now *approximate* the conditional probability $P(A \mid U)$ with a similar parametric model:

$$\tilde{P}(A \mid U) = \mathcal{S}(\mathbf{Q}^\mathsf{T} \mathbf{P}), \tag{9}$$

where $\mathbf{Q} \in \mathbb{R}^{D \times N}$ concatenates all query vectors horizontally. Substituting equations 7–9 we write the full approximate attention $\tilde{\mathcal{A}}$, multiplied by an arbitrary matrix $\mathbf{V}$ (which in the case of a transformer contains the values of the key–value pairs stacked as rows):

$$\tilde{P}(A)\mathbf{V} = \mathcal{S}(\mathbf{Q}^\mathsf{T} \mathbf{P}) \left(\mathcal{S}(\mathbf{P}^\mathsf{T} \mathbf{K}) \mathbf{V} \right). \tag{10}$$

[3] I.e. $[\mathcal{S}(\mathbf{z})]_i = \exp(z_i/\sqrt{D}) / \sum_j \exp(z_j/\sqrt{D})$. For matrix inputs, the sum is over the columns.

Table 1: **Comparison of recent video transformer models.** We show the different design choices of recent video transformer models and how they compare to our proposed Motionformer model.

Model	Base Model	Attention	Pos. Encoding	Tokenization
TimeSformer [7]	ViT-B	Divided Space–Time	Separate	Square
ViViT [2]	ViT-L	Joint/Divided Space–Time	Joint	Cubic
Motionformer	ViT-B	Trajectory	Separate	Cubic

Computational efficiency. One important feature of the approximation in eq. 10 is that it can be computed in two steps. First the values $\mathbf{V}$ are multiplied by a prototypes-keys attention matrix $\mathcal{S}(\mathbf{P}^\mathsf{T}\mathbf{K}) \in \mathbb{R}^{R\times N}$, which can be much smaller than the full attention matrix $\mathcal{S}(\mathbf{Q}^\mathsf{T}\mathbf{K}) \in \mathbb{R}^{N\times N}$ (eq. 7), i.e., $R \ll N$. Finally, this product is multiplied by a queries-prototypes attention matrix $\mathcal{S}(\mathbf{Q}^\mathsf{T}\mathbf{P}) \in \mathbb{R}^{N\times R}$, which is also small. This allows us to sidestep the quadratic dependency of full attention over the input and output size ($\mathcal{O}(N^2)$), replacing it with linear complexity ($\mathcal{O}(N)$) as long as R is kept constant.

Prototype selection. The aim for prototype-based attention approximation schemes is to use as few prototypes as possible while reconstructing the attention operation as accurately as possible. As such, it behooves us to select prototypes efficiently. We have two priorities for the prototypes: to dynamically adjust to the query and key vectors so that their region of space is well-reconstructed, and to minimize redundancy. The latter is important because the relative probability of a query–key pair may be over-estimated if many prototypes are clustered near that query and key. To address these criteria, we incrementally build a set of prototypes from the set of queries and keys such that a new prototype is maximally orthogonal to the prototypes already selected, starting with a query or key at random. This greedy strategy is dynamic, since it selects prototypes from the current set of queries and keys, and has high entropy, since it preferences well-separated prototypes. Moreover, it balances speed and performance by using a greedy strategy, rather than finding a globally-optimal solution to the maximum entropy sampling problem [62], making it suitable for use in a transformer.

Naïvely applying prototype-based attention approximation techniques to video transformers would involve creating a unique set of prototypes for each frame in the video. However, additional memory savings can be realized by sharing prototypes across time. Since there is significant information redundancy between frames, video data is opportune for compression via temporally-shared prototypes.

Orthoformer algorithm. The proposed approximation algorithm is outlined in Algorithm 1. The attention matrix is approximated using intermediate prototypes, selected as the most orthogonal subset of the queries and keys, given a desired number of prototypes R. To avoid a linear dependence on the sequence length N, we first randomly subsample cR queries and keys, for a constant c, before selecting the most orthogonal subset, resulting in a complexity quadratic in the number of prototypes $\mathcal{O}(R^2)$. The algorithm then computes two attention matrices, much smaller than the original problem, and multiplies them with the values. The most related approach in the literature is Nyströmformer [85] attention, outlined in Algorithm 2. This approach involves a pseudoinverse to attenuate the effect of near-parallel prototypes, has more operations, and a greater memory footprint.

Algorithm 1 Orthoformer (proposed) attention

$\mathbf{P} \leftarrow \text{MostOrthogonalSubset}(\mathbf{Q}, \mathbf{K}, R)$
$\mathbf{\Omega}_1 = \mathcal{S}(\mathbf{Q}^\mathsf{T}\mathbf{P}/\sqrt{D})$
$\mathbf{\Omega}_2 = \mathcal{S}(\mathbf{P}^\mathsf{T}\mathbf{K}/\sqrt{D})$
$\mathbf{Y} = \mathbf{\Omega}_1(\mathbf{\Omega}_2\mathbf{V})$

Algorithm 2 Nyströmformer [85] attention

$\mathbf{P}_q, \mathbf{P}_k \leftarrow \text{SegmentMeans}(\mathbf{Q}, \mathbf{K}, R)$
$\mathbf{\Omega}_1 = \mathcal{S}(\mathbf{Q}^\mathsf{T}\mathbf{P}_k/\sqrt{D})$
$\mathbf{\Omega}_2^{-1} = \text{IterativeInverse}(\mathcal{S}(\mathbf{P}_q^\mathsf{T}\mathbf{P}_k/\sqrt{D}), N_{\text{iter}})$
$\mathbf{\Omega}_3 = \mathcal{S}(\mathbf{P}_q^\mathsf{T}\mathbf{K}/\sqrt{D})$
$\mathbf{Y} = \mathbf{\Omega}_1\left(\mathbf{\Omega}_2^{-1}\left(\mathbf{\Omega}_3\mathbf{V}\right)\right)$

3.3 The Motionformer model

Our full video transformer model builds on previous work, as shown in Table 1. In particular, we use the ViT image transformer model [20] as the base architecture, the separate space and time positional encodings of TimeSformer [7], and the cubic image tokenization strategy as in ViViT [2]. These design choices are ablated in Section 4. The crucial difference for our model is the trajectory attention mechanism, with which we demonstrate greater empirical performance than the other models.

Table 2: **Input encoding ablations:** Comparison of input tokenization and positional encoding design choices. We report GFLOPS and top-1 accuracy (%) on K-400 and SSv2.

(a) Cubic tokenization works best for trajectory attn.

Attention	Tokenization	GFlops	K-400	SSv2
Joint ST	Square (1×16^2)	179.7	79.4	63.0
	Cubic (2×16^2)	180.6	79.2	64.0
Trajectory	Square (1×16^2)	368.5	79.4	65.8
	Cubic (2×16^2)	369.5	**79.7**	**66.5**

(b) Trajectory attn. works well with both encodings.

Attention	Pos. Encoding	GFlops	K-400	SSv2
Joint ST	Joint ST	180.6	79.1	60.8
	Separate ST [22]	180.6	79.2	64.0
Trajectory	Joint ST	369.5	79.6	65.8
	Separate ST [22]	369.5	**79.7**	**66.5**

4 Experiments

Datasets. **Kinetics** [35] is a large-scale video classification dataset consisting of short clips collected from YouTube, licensed by Google under Creative Commons. As it is a dataset of human actions, it potentially contains personally identifiable information such as faces, names and license plates. **Something–Something V2** [27] is a video dataset containing more than 200,000 videos across 174 classes, with a greater emphasis on short temporal clips. In contrast to Kinetics, the background and objects remain consistent across different classes, and therefore models have to reason about fine-grained motion signals. We verified the importance of temporal reasoning on this dataset by showing that a single frame model gets significantly worse results, a decrease of 39% top-1 accuracy. In contrast, a drop of only 7% is seen on the Kinetics-400 dataset, showing that temporal information is much less relevant there. We obtained a research license for this data from `https://20bn.com`; the data was collected with consent. **Epic Kitchens-100** [16] is an egocentric video dataset capturing daily kitchen activities. The highest scoring verb and action pair predicted by the network constitutes an action, for which we report top-1 accuracy. The data is licensed under Creative Commons and was collected with consent by the Epic Kitchens teams.

Implementation details. We follow a standard training and augmentation pipeline [2], as detailed in the appendix. For ablations, our default Motionformer model is the Vision Transformer Base architecture [20] (ViT/B), pretrained on ImageNet-21K [17], patch-size $2\times16\times16$ with central frame initialization [2], separate space-time positional embedding and our trajectory attention. The base architecture has 12 layers, 12 attention heads, and an embedding dimension of 768. Our default Motionformer model operates on $16\times224\times224$ videos with temporal stride 4 i.e. temporal extent of 2s. For comparisons with state-of-the-art, we report results on two additional variants: Motionformer-HR, which has a high spatial resolution ($16\times336\times336$ videos with temporal stride 4 i.e. temporal extent of 2s), and Motionformer-L, which has a long temporal range ($32\times224\times224$ videos with temporal stride 3 i.e. temporal extent of 3s). Experiments with the large ViT architecture are deferred to the appendix.

4.1 Ablation studies

Input: tokenization. We consider the effect of different input tokenization approaches for both joint and trajectory attention on Kinetics-400 (K-400) and Something–Something V2 (SSv2) in Table 2b. For patch tokenization ($1\times16\times16$), we use inputs of size $8\times224\times224$, while for cubic [2, 22] tokenization ($2\times16\times16$), we use inputs of size $16\times224\times224$ to ensure that the model has the same number of input tokens over the same temporal range of 2 seconds. For both attention types, we see that cubic tokenization gives a 1% accuracy improvement over square tokenization on SSv2, a dataset for which temporal information is critical. Furthermore, our proposed trajectory attention using cubic tokenization outperforms joint space-time attention on both datasets.

Input: positional encoding. Here, we ablate using a joint or separate [22] (default) space-time positional encoding in Table 2b. Similar to the results for input tokenization, the choice of positional encoding is particularly important for the fine-grained motion dataset, SSv2. Since joint space-time attention treats tokens in the space-time volume equally, it benefits particularly from separating the positional encodings, allowing it to differentiate between space and time dimensions, with a 4% improvement on SSv2 over joint space-time encoding. Our proposed trajectory attention elicits a more modest improvement of 1% from using separated positional encodings on SSv2, and outperforms joint space-time attention in both settings on both datasets.

Table 3: **Orthoformer ablations:** We ablate various aspects of our Orthoformer approximation. E denotes exact attention and A denotes approximate attention. We report max CUDA memory consumption (GB) and top-1 accuracy (%) on K-400 and SSv2.

(a) Orthoformer is competitive with Nyström.

Attention	Approx.	Mem.	K-400	SSv2
Trajectory (E)	N/A	7.4	**79.7**	**66.5**
Trajectory (A)	Performer	5.1	72.9	52.7
	Nyströmformer	3.8	**77.5**	**64.0**
	Orthoformer	3.6	**77.5**	63.8

(b) Selecting orthogonal prototypes is the best strategy.

Attention	Selection	Mem.	K-400	SSv2
Trajectory (E)	N/A	7.4	**79.7**	**66.5**
Trajectory (A)	Seg-Means	3.6	75.8	60.3
	Random	3.6	76.5	62.5
	Orthogonal	3.6	**77.5**	**63.8**

(c) Approximation improves with more prototypes.

Attention	# Prototypes	Mem.	K-400	SSv2
Trajectory (E)	N/A	7.4	**79.7**	**66.5**
Trajectory (A)	16	3.1	73.9	59.2
	64	3.3	74.9	63.0
	128	3.6	**77.5**	**63.8**

(d) Temporal sharing is the best strategy.

Attention	Sharing	Mem.	K-400	SSv2
Trajectory (E)	N/A	7.4	**79.7**	**66.5**
Trajectory (A)	✗	16.5	77.3	61.5
	✓	3.6	**77.5**	**63.8**

Table 4: **Attention ablations:** We compare trajectory attention with alternatives and ablate its design choices. We report GFLOPS and top-1 accuracy (%) on K-400 and SSv2. Att_T: temporal attention, Avg_T: temporal averaging, Norm_{ST}: space-time normalization, Norm_S: spatial normalization.

Attention	Att_T	Avg_T	Norm_S	Norm_{ST}	GFLOPS	K-400	SSv2
Joint Space-Time	–	–	–	–	180.6	79.2	64.0
Divided Space-Time	–	–	–	–	185.8	78.5	64.2
	✗	✓	✓	✗	180.6	76.0	60.0
	✓	✗	✗	✓	369.5	77.2	60.9
Trajectory	✓	✗	✓	✗	369.5	**79.7**	**66.5**

Attention block: comparisons. We compare our proposed trajectory attention to joint space-time attention [2], and divided space-time attention [7] in Table 4. Our trajectory attention (bottom row) outperforms both alternatives on the K-400 and SSv2 datasets. While we see only modest improvements on the appearance cue-reliant K-400 dataset, our trajectory attention significantly outperforms ($+2\%$) the other approaches on the motion cue-reliant SSv2 dataset. This dataset requires fine-grained motion understanding, something explicitly singled out by previous video transformer works [2, 7] as a challenge for their models. In contrast, our trajectory attention excels on this dataset, indicating that its motion-based design is able to capture some of this information.

Attention block: trajectory attention design. We ablate two design choices for our trajectory attention: the per-frame softmax normalization and the 1D temporal attention. Unlike joint space-time attention, which normalizes the attention map over all tokens in space and time, trajectory attention normalizes independently per frame, allowing us to implicitly track the trajectories of query patches in time. In row 5 of Table 4, we ablate the benefits of this design choice. We observe a reduction of 2.5% on K-400 and 5.6% on SSv2 by normalizing over space and time (Norm_{ST}) compared with normalizing over space alone (Norm_S). In row 4, we show the benefit of using 1D temporal attention (Att_T) to aggregate temporal features, compared to average pooling (Avg_T). We observe reductions of 3.7% on K-400 and 6.5% on SSv2 when using average pooling instead of temporal attention applied to the motion trajectories, although it saves computing the additional query/key/value projections.

4.2 Orthoformer approximated attention

Approximation comparisons. In Table 3a, we compare our Orthoformer algorithm to alternative strategies: Nyströmformer [85] and Performer [14]. Our algorithm performs comparably with Nyströmformer with a reduced memory footprint. In Table 5, we also compare these attention mechanisms on the Long Range Arena benchmark [70] to show applicability to other tasks and data types. Orthoformer is able to effectively approximate self-attention, outperforming the state-of-the-art despite using far fewer prototypes (64) and so gaining significant computational and memory benefits.

Table 5: **Comparison to the state-of-the-art on Long Range Arena benchmark.** GFLOPS and CUDA maximum Memory (MB) are reported for the ListOps task. Note that our algorithm achieves the best overall results with far fewer prototypes (64) than the other methods.

Model	ListOps	Text	Retrieval	Image	Pathfinder	Avg↑	GFLOPS↓	Mem.↓
Exact [76]	36.69	63.09	78.22	31.47	66.35	55.16	1.21	4579
Performer-256 [14]	36.69	63.22	**78.98**	29.39	**66.55**	54.97	0.49	885
Nyströmformer-128 [85]	**36.90**	64.17	78.67	**36.16**	52.32	53.64	0.62	745
Orthoformer-64	33.87	**64.42**	78.36	33.26	66.41	**55.26**	**0.24**	**344**

Table 6: **Comparison to the state-of-the-art on video action recognition.** We report GFLOPS and top-1 (%) and top-5 (%) video action recognition accuracy on K-400/600, and SSv2. On Epic-Kitchens, we report top-1 (%) action (A), verb (V), and noun (N) accuracy.

(a) **Something–Something V2**

Model	Pretrain	Top-1	Top-5	GFLOPs ×views
SlowFast [25]	K-400	61.7	-	65.7×3×1
TSM [46]	K-400	63.4	88.5	62.4×3×2
STM [33]	IN-1K	64.2	89.8	66.5×3×10
MSNet [40]	IN-1K	64.7	89.4	67×1×1
TEA [45]	IN-1K	65.1	-	70×3×10
bLVNet [23]	IN-1K	65.2	90.3	128.6×3×10
VidTr-L [44]	IN-21K+K-400	60.2	-	351×3×10
Tformer-L [7]	IN-21K	62.5	-	1703×3×1
ViViT-L [2]	IN-21K+K-400	65.4	89.8	3992×4×3
MViT-B [22]	K-400	67.1	90.8	170×3×1
Mformer	IN-21K+K-400	66.5	90.1	369.5×3×1
Mformer-L	IN-21K+K-400	**68.1**	**91.2**	1185.1×3×1
Mformer-HR	IN-21K+K-400	67.1	90.6	958.8×3×1

(b) **Kinetics-400**

Method	Pretrain	Top-1	Top-5	GFLOPs×views
I3D [10]	IN-1K	72.1	89.3	108×N/A
R(2+1)D [75]	-	72.0	90.0	152×5×23
S3D-G [84]	IN-1K	74.7	93.4	142.8×N/A
X3D-XL [24]	-	79.1	93.9	48.4×3×10
SlowFast [25]	-	79.8	93.9	234×3×10
VTN [51]	IN-21K	78.6	93.7	4218×1×1
VidTr-L [44]	IN-21K	79.1	93.9	392×3×10
Tformer-L[7]	IN-21K	80.7	94.7	2380×3×1
MViT-B [22]	-	81.2	95.1	455×3×3
ViViT-L [2]	IN-21K	**81.3**	94.7	3992×3×4
Mformer	IN-21K	79.7	94.2	369.5×3×10
Mformer-L	IN-21K	80.2	94.8	1185.1×3×10
Mformer-HR	IN-21K	81.1	**95.2**	958.8×3×10

(c) **Epic-Kitchens**

Method	Pretrain	A	V	N
TSN [78]	IN-1K	33.2	60.2	46.0
TRN [86]	IN-1K	35.3	65.9	45.4
TBN [36]	IN-1K	36.7	66.0	47.2
TSM [46]	IN-1K	38.3	**67.9**	49.0
SlowFast [25]	K-400	38.5	65.6	50.0
ViViT-L [2]	IN-21K+K-400	44.0	66.4	56.8
Mformer	IN-21K+K-400	43.1	66.7	56.5
Mformer-L	IN-21K+K-400	44.1	67.1	57.6
Mformer-HR	IN-21K+K-400	**44.5**	67.0	**58.5**

(d) **Kinetics-600**

Model	Pretrain	Top-1	Top-5	GFLOPs ×views
AttnNAS [81]	-	79.8	94.4	-
LGD-3D [56]	IN-1K	81.5	95.6	-
SlowFast [25]	-	81.8	95.1	234×3×10
X3D-XL [24]	-	81.9	95.5	48.4×3×10
Tformer-HR [7]	IN-21K	82.4	96.0	1703×3×1
ViViT-L [2]	IN-21K	83.0	95.7	3992×3×4
MViT-B-24 [22]	-	**83.8**	**96.3**	236×1×5
Mformer	IN-21K	81.6	95.6	369.5×3×10
Mformer-L	IN-21K	82.2	96.0	1185.1×3×10
Mformer-HR	IN-21K	82.7	96.1	958.8×3×10

Prototype selection. A key part of our Orthoformer algorithm is the prototype selection procedure. In Table 3b, we ablate three prototype selection strategies: segment-means, random, and greedy most-orthogonal selection. Segment-means, the strategy used in Nyströmformer, performs poorly because it can generate multiple parallel prototypes, which will over-estimate the relative probability of query–key pairs near those redundant prototypes. In contrast, our proposed strategy of selecting the most orthogonal prototypes from the query and key set works the best across both datasets, because it explicitly minimises prototype redundancy with respect to direction.

Number of prototypes. In Table 3c, we show that Orthoformer improves monotonically as the number of prototypes is increased. In particular, we see an average performance improvement of 4% on both datasets as we increase the number of prototypes from 16 to 128.

Temporally-shared prototypes. In Table 3d, we demonstrate the memory savings and performance benefits of sharing prototypes across time. On SSv2, we observe a 2% improvement in performance and a $5\times$ decrease in memory usage. These gains may be attributed to the regularization effect of having prototypes leverage redundant information across frames.

4.3 Comparison to the state-of-the-art

In Table 6, we compare our method against the current state-of-the-art on four common benchmarking datasets: Kinetics-400, Kinetics-600, Something–Something v2 and Epic-Kitchens. We find that our method performs favorably against current methods, even when compared against much larger models such as ViViT-L. In particular, it achieves strong top-1 accuracy improvements of 1.0% and 2.3% for SSv2 and Epic-Kitchen Nouns, respectively. These datasets require greater motion reasoning than Kinetics and so are a more challenging benchmark for video action recognition.

5 Conclusion

We have presented a new general-purpose attention block for video data that aggregates information along implicitly determined motion trajectories, lending a realistic inductive bias to the model. We further address its quadratic dependence on the input size with a new attention approximation algorithm that significantly reduces the memory requirements, the largest bottleneck for transformer models. With these contributions, we obtain state-of-the-art results on several benchmark datasets. Nonetheless, our approach inherits many of the limitations of transformer models, including poor data efficiency and slow training. Specific to this work, trajectory attention has higher computational complexity than alternative attention operations used for video data. This is attenuated by the proposed approximation algorithm, with significantly reduced memory and computation requirements. However, its runtime is bottlenecked by prototype selection, which is not easily parallelized.

Potential negative societal impacts. One negative impact of this research is the significant environmental impact associated with training transformers, which are large and compute-expensive models. Compared to 3D-CNNs where the compute scales linearly with the sequence length, video transformers scale quadratically. To mitigate this, we proposed an approximation algorithm with linear complexity that greatly reduces the computational requirements. There is also potential for video action recognition models to be misused, such as for unauthorized surveillance.

Acknowledgments and Disclosure of Funding

We are grateful for support from the Rhodes Trust (M.P.), Qualcomm Innovation Fellowship (Y.A.), the Royal Academy of Engineering (DFR05420, J.H), and EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems [EP/L015897/1] (M.P. and Y.A.). M.P. funding was received under his Oxford affiliation. We thank Bernie Huang, Dong Guo, Rose Kanjirathinkal, Gedas Bertasius, Mike Zheng Shou, Mathilde Caron, Hugo Touvron, Benjamin Lefaudeux, Haoqi Fan, and Geoffrey Zweig from Facebook AI for their help, support, and discussion around this project. We also thank Max Bain and Tengda Han from VGG for fruitful discussions.

References

[1] Hassan Akbari, Linagzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text, 2021.

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[4] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *NeurIPS*, 2020.

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval, 2021.

[6] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. 2020.

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[10] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[11] Mark Chen, Alec Radford, Rewon Child, Jeff Wu, Heewoo Jun, Prafulla Dhariwal, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020.

[12] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. a^2-nets: Double attention networks. In *NeurIPS*, 2018.

[13] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *URL https://openai.com/blog/sparse-transformers*, 2019.

[14] Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. Rethinking attention with performers. In *ICLR*, 2021.

[15] E. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical data augmentation with no separate search. In *CVPRW*, 2020.

[16] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. In *ECCV*, 2020.

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[18] Karan Desai and Justin Johnson. VirTex: Learning Visual Representations from Textual Annotations. In *CVPR*, 2021.

[19] Carl Doersch, Ankush Gupta, and Andrew Zisserman. Crosstransformers: spatially-aware few-shot transfer. In *NeurIPS*, 2020.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[21] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. `https://github.com/facebookresearch/slowfast`, 2020.

[22] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers, 2021.

[23] Quanfu Fan, Chun-Fu (Ricarhd) Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More Is Less: Learning Efficient Video Representations by Temporal Aggregation Modules. In *NeurIPS*, 2019.

[24] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020.

[25] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019.

[26] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020.

[27] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, and et al. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[29] Po-Yao Huang, Mandela Patrick, Junjie Hu, Graham Neubig, Florian Metze, and Alexander Hauptmann. Multilingual multimodal pre-training for zero-shot cross-lingual transfer of vision-language models. In *NAACL*, 2021.

[30] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2016.

[31] Allan Jabri, Andrew Owens, and Alexei Efros. Space-time correspondence as a contrastive random walk. In *NeurIPS*, 2020.

[32] Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2018.

[33] Boyuan Jiang, Mengmeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *ICCV*, 2019.

[34] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation, 2021.

[35] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset, 2017.

[36] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019.

[37] Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020.

[38] Alexander Klaser, Marcin Marszałek, and Cordelia Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

[40] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *ECCV*, 2020.

[41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[42] Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. In *ICLR*, 2021.

[43] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language, 2019.

[44] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions, 2021.

[45] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, 2020.

[46] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. 2019.

[47] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

[48] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam, 2018.

[49] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019.

[50] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *ICLR*, 2018.

[51] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network, 2021.

[52] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *ICML*, 2018.

[53] Mandela Patrick, Yuki M. Asano, Bernie Huang, Ishan Misra, Florian Metze, Joao Henriques, and Andrea Vedaldi. Space-time crop & attend: Improving cross-modal video representation learning, 2021.

[54] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander G Hauptmann, Joao F. Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. In *ICLR*, 2021.

[55] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017.

[56] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. *CVPR*, 2019.

[57] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[58] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. Strike a pose: Tracking people by finding stylized poses. In *Proc. CVPR*, 2005.

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015.

[60] Mert Bulent Sariyildiz, Julien Perez, and Diane Larlus. Learning visual representations with caption annotations. In *ECCV*, 2020.

[61] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *ACM MM*, 2007.

[62] Michael C Shewry and Henry P Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[63] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020.

[64] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning, 2019.
[65] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019.
[66] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.
[67] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
[68] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.
[69] Hao Tan and Mohit Bansal. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *EMNLP*, 2020.
[70] Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena : A benchmark for efficient transformers. In *ICLR*, 2021.
[71] Zachary Teed and Jia Deng. RAFT: recurrent all-pairs field transforms for optical flow. In *Proc. ECCV*, 2020.
[72] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.
[73] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.
[74] Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In *ICCV*, 2019.
[75] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.
[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.
[77] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.
[78] Limin Wang, Yuanjun Xiong, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.
[79] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. In *NeurIPS*, 2020.
[80] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.
[81] Xiaofang Wang, Xuehan Xiong, Maxim Neumann, AJ Piergiovanni, Michael S Ryoo, Anelia Angelova, Kris M Kitani, and Wei Hua. Attentionnas: Spatiotemporal attention cell search for video classification. In *ECCV*, 2020.
[82] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019.
[83] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.
[84] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018.
[85] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, 2021.
[86] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018.
[87] Linchao Zhu and Yi Yang. Actbert: Learning global-local video-text representations. In *CVPR*, 2020.

6 Appendix

6.1 Further experimental analysis and results

6.1.1 Does trajectory attention make better use of motion cues?

In the main paper (and below in Section 6.1.2), we provide evidence that action classification on the Something–Something V2 (SSv2) dataset [27] is more reliant on motion cues than the Kinetics dataset [35], where appearance cues dominate and a single-frame model achieves high accuracy. Improved performance on SSv2 is one way to infer that our model makes better use of temporal information, however, here we consider another way. We artificially adjust the speed of the video clips by changing the temporal stride of the input. A larger stride simulates faster motions, with adjacent frames being more different. If our trajectory attention is able to make better use of the temporal information in the video than the other attention mechanisms, we expect the margin of improvement to increase as the temporal stride increases. As shown in Figure 3, this is indeed what we observe, with the lines diverging as temporal stride increases, especially for the motion cue-reliant SSv2 dataset. Since the same number of frames are used as input in all cases, the larger the stride, the more of the video clip is seen by the model. This provides additional confirmation that seeing a small part of a Kinetics video is usually enough to classify it accurately, as shown on the bottom left, where the absolute accuracy is reported.

6.1.2 How important are motion cues for classifying videos from the Kinetics-400 and Something–Something V2 datasets?

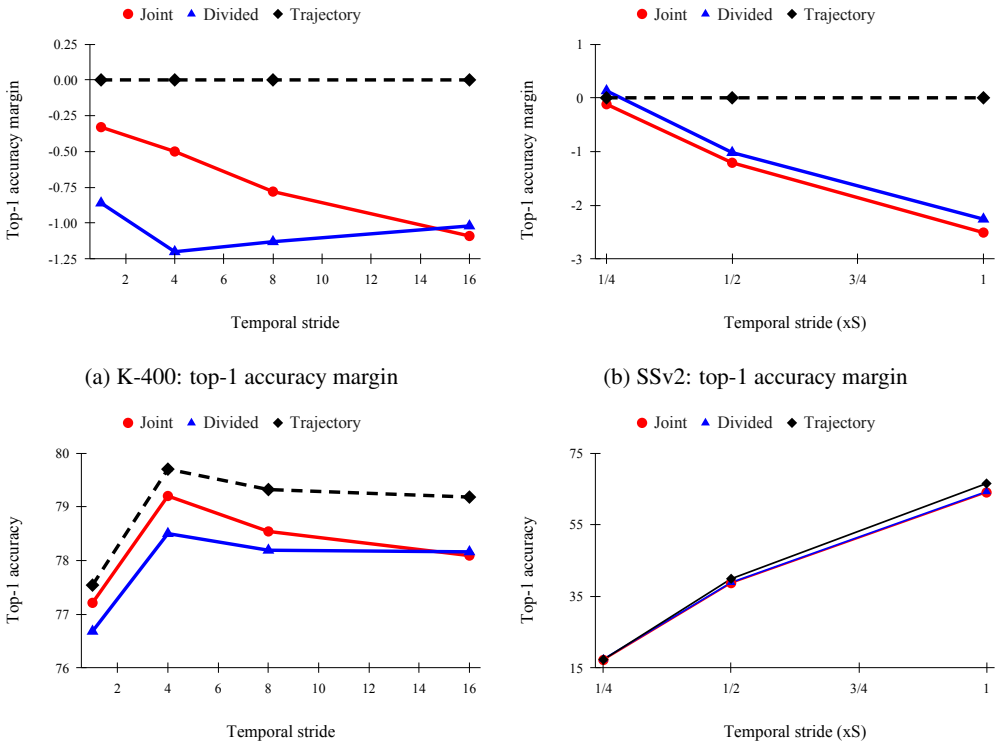
To determine the relative importance of motion cues compared to appearance cues for classifying videos on two of the major video action recognition datasets (Kinetics-400 and Something–Something V2), we trained a single frame vision transformer model and compare the results to a multi-frame model that can reason about motion. The single frame was sampled from the video at random. Table 7 shows that single-frame action classifiers can do almost as well as video action classifiers on the Kinetics-400 dataset, implying that the motion information is much less relevant. In contrast, classifying videos from the Something-Something V2 dataset clearly requires this motion information. Therefore, to excel on the SSv2 dataset, a model must reason about motion information. Our model, which introduces an inductive bias that favors pooling along motion trajectories, is able to do this and sees corresponding performance gains.

Table 7: **Importance of motion cues for the K-400 and SSv2 datasets.** A classifier for the K-400 dataset performs well when all motion information is removed (1 frame model), while a classifier for the SSv2 dataset performs very poorly. Therefore, SSv2 is a better dataset for evaluating *video* action classification, where the combination of appearance and motion is critical.

Dataset	Top-1 accuracy (1 frame)	Top-1 accuracy (8 frames)	Δ
Kinetics-400	73.2	79.7	6.5
Something–Something V2	27.1	66.5	**39.4**

6.1.3 Can we train larger models using approximated trajectory attention?

The Orthoformer attention approximation algorithm allows us to train larger models and higher resolution inputs for a given GPU memory budget. Here, we verify that this is the case, by training a large vision transformer model (ViT-L/16) [20] with a higher resolution input (336×336 pixels) on the Kinetics-400 dataset, using the Orthoformer approximation with 196 temporally-shared prototypes and the same schedule as the base model. We use a fixed patch size (in pixels) for all models, and so the number of input tokens to the transformer scales with the square of the image resolution. As shown in Table 8, this model achieves a competitive accuracy without fine-tuning the training schedule, hyperparameters or data augmentation strategy. We expect that fine-tuning these on a validation set would greatly improve the model's performance, based on results from contemporary work [2]. Obviously such a parameter sweep is more time-consuming for these large models than the base model, however these preliminary results are indicative that higher accuracies are attainable if these parameters were to be optimized.



(a) K-400: top-1 accuracy margin

(b) SSv2: top-1 accuracy margin

(c) K-400: top-1 accuracy

(d) SSv2: top-1 accuracy

Figure 3: **Does trajectory attention make better use of motion cues?** Performance of transformer models with joint space-time attention, divided space-time attention, and trajectory attention, as the temporal stride increases, on the Kinetics-400 dataset (left) and the Something–Something V2 dataset (right). Top: top-1 accuracy margin relative to trajectory attention (difference of accuracy and trajectory accuracy). Bottom: absolute top-1 accuracy shown for reference. If our trajectory attention is able to make better use of the temporal information in the video than the other attention mechanisms, we expect the accuracy margin between the methods to increase as the temporal stride increases. This is indeed the observed behaviour, especially for the motion cue-reliant SSv2 dataset. A larger stride simulates greater motion between input frames, which trajectory attention is better able to model and reason about. Note that the larger the stride, the more of the video clip is seen by the model; for all plots, the rightmost side of the axis corresponds to the entire video clip. Note also that the strides for SSv2 are written as multiples of S, the stride needed to evenly sample the entire video clip.

6.1.4 Trajectory attention maps

In Figure 4, we show qualitative results of the intermediate attention maps of our trajectory attention operation. The learned attention maps appear to implicitly track the query points across time, a strategy that is easier to learn with the inductive bias instilled by trajectory attention.

6.2 Implementation details

Preprocessing. During training, we randomly sample clips of size $16\times224\times224$ at a rate of $1/4$ from 30 FPS videos, thereby giving an effective temporal resolution of just over 2 seconds. We normalize the inputs with mean and standard deviation 0.5, rescaling in the range $[-1, 1]$. We use standard video augmentations such as random scale jittering, random horizontal flips and color jittering. For smaller datasets such as Something–Something V2 and Epic-Kitchens, we additionally apply rand-augment [15]. During testing, we uniformly sample 10 clips per video and apply a 3 crop evaluation [25].

Table 8: **Can we train larger models using approximated trajectory attention?** We report top-1 and top-5 accuracy (%) on the Kinetics-400 dataset of two variants of our Motionformer model: Motionformer-B and Motionformer-H. The former uses the base model with exact (E) trajectory attention, while the latter uses a much larger model (ViT-L) and a higher resolution input (336×336 pixels) with approximate (A) trajectory attention, i.e., using Orthoformer. The larger model has better performance, despite no optimization of the training schedule, hyperparameters, and data augmentation schedule. The larger model also has far more parameters than the base model, and so unavoidably requires more GPU memory. Furthermore, for a fixed patch size (in pixels), the memory requirements for exact attention scale with the square of the input resolution. We reduce this to a linear relationship with the Orthoformer approximation, which allows us to fit the model on the GPU.

Model	Base model	Params	Attention	Max memory (GB)	Top-1	Top-5
Mformer-B	ViT-B/224	109.1M	Trajectory (E)	**7.3**	79.7	94.2
Mformer-H	ViT-L/336	381.9M	Trajectory (A)	22.2	**80.0**	**94.5**

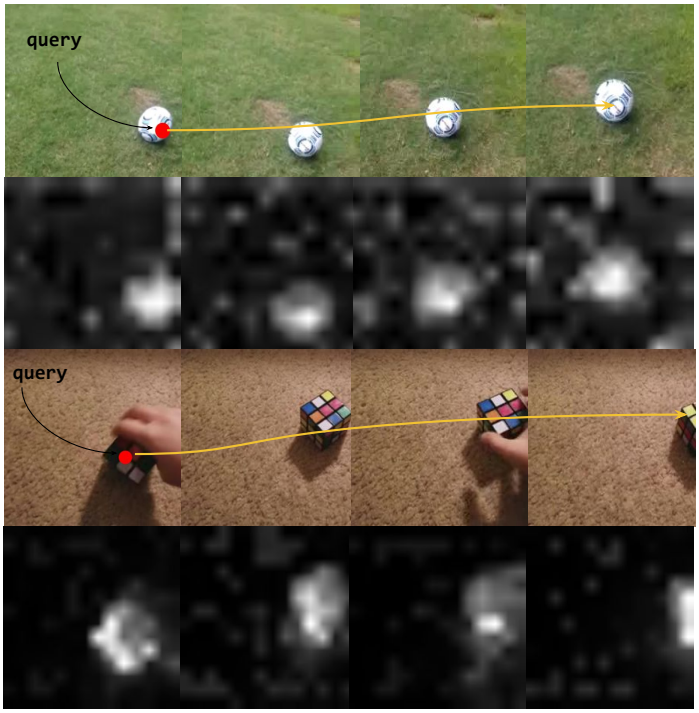


Figure 4: **Trajectory attention maps.** In this sequence of frames from Kinetics-400 (row 1) and Something-Something V2 (row 3), we show the attention maps at each frame given an initial query point (red point). We see that the model learns to implicitly track along motion paths (yellow arrow) using our trajectory attention module.

Training. For all datasets, we use the AdamW [48] optimizer with weight decay 5×10^{-2}, a batch size per GPU of 4, label smoothing [67] with alpha 0.2 and mixed precision training [50]. For Kinetics-400/600 and Something-Something V2, we train for 35 epochs, with an initial learning rate of 10^{-4}, which we decay by 10 at epochs 20, 30. As Epic-Kitchens is a smaller dataset, we use a longer schedule and train for 50 epochs with decay at 30 and 40.

Long Range Arena benchmark details. For the Long-Range Arena benchmark [70], we used the training, validation, and testing code and parameters from the Nyströmformer Github repository. The Performer [14] implementation was ported over to PyTorch from the official Github repo, and the Nyströmformer [85] implementation was used directly from its Github repository.

Computing resources. Ablation experiments were run on a GPU cluster using 4 nodes (32 GPUs) with an average training time of 12 hours. Experiments for comparing with state-of-the-art models used 8 nodes (64 GPUs), with an average training time of 7 hours.

Libraries. For our code implementation, we used the `timm` [82] library for our base vision transformer implementation, and the `PySlowFast` [21] library for training, data processing, and the evaluation pipeline.

Part II

Interpreting Representations

7

Understanding deep networks via extremal perturbations and smooth masks

This work was presented at the International Conference on Computer Vision (ICCV) 2019 as an oral.

Understanding Deep Networks via Extremal Perturbations and Smooth Masks

Ruth Fong[†*]
University of Oxford

Mandela Patrick[†]
University of Oxford

Andrea Vedaldi
Facebook AI Research

Abstract

The problem of attribution is concerned with identifying the parts of an input that are responsible for a model's output. An important family of attribution methods is based on measuring the effect of perturbations applied to the input. In this paper, we discuss some of the shortcomings of existing approaches to perturbation analysis and address them by introducing the concept of extremal perturbations, which are theoretically grounded and interpretable. We also introduce a number of technical innovations to compute extremal perturbations, including a new area constraint and a parametric family of smooth perturbations, which allow us to remove all tunable hyper-parameters from the optimization problem. We analyze the effect of perturbations as a function of their area, demonstrating excellent sensitivity to the spatial properties of the deep neural network under stimulation. We also extend perturbation analysis to the intermediate layers of a network. This application allows us to identify the salient channels necessary for classification, which, when visualized using feature inversion, can be used to elucidate model behavior. Lastly, we introduce TorchRay[1], an interpretability library built on PyTorch.

1. Introduction

Deep networks often have excellent prediction accuracy, but the basis of their predictions is usually difficult to understand. *Attribution* aims at characterising the response of neural networks by finding which parts of the network's input are the most responsible for determining its output. Most attribution methods are based on backtracking the network's activations from the output back to the input, usually via a modification of the backpropagation algorithm [23, 31, 26, 32, 22, 3]. When applied to computer vision models, these methods result in *saliency maps* that highlight important regions in the input image.

However, most attribution methods do not start from a definition of what makes an input region important for the neural network. Instead, most saliency maps are validated *a-posteriori* by either showing that they correlate with the image content (e.g., by highlighting relevant object categories), or that they find image regions that, if perturbed, have a large effect on the network's output (see Sec. 2).

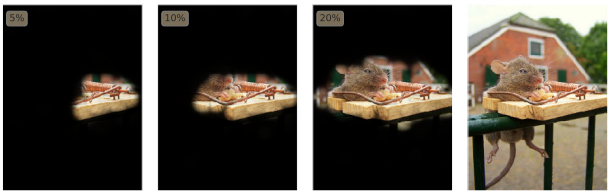


Figure 1: **Extremal perturbations** are regions of an image that, for a given area (boxed), maximally affect the activation of a certain neuron in a neural network (i.e., "mousetrap" class score). As the area of the perturbation is increased, the method reveals more of the image, in order of decreasing importance. For clarity, we black out the masked regions; in practice, the network sees blurred regions.

Some attribution methods, on the other hand, directly perform an analysis of the effect of *perturbing* the network's input on its output [31, 20, 7, 5]. This usually amounts to selectively deleting (or preserving) parts of the input and observing the effect of that change to the model's output. The advantage is that the meaning of such an analysis is clear from the outset. However, this is not as straightforward as it may seem on a first glance. First, since it is not possible to visualise *all* possible perturbations, one must find *representative* ones. Since larger perturbations will have, on average, a larger effect on the network, one is usually interested in small perturbations with a large effect (or large perturbations with a small effect). Second, Fong and Vedaldi [7] show that searching for perturbations with a large effect on the network's output usually results in *pathological* perturbations that trigger adversarial effects in the network. Characterizing instead the *typical* behavior of the model requires restricting the search to more representative perturbations via regularization terms. This results in an optimization problem that trades off maximizing the effect of the perturbation with its smoothness and size. In practice, this trade off is difficult to control numerically and somewhat obscures the meaning of the analysis.

In this paper, we make three contributions. First, instead

[*]Work done as a contractor at FAIR. [†] denotes equal contributions.
[1] github.com/facebookresearch/TorchRay

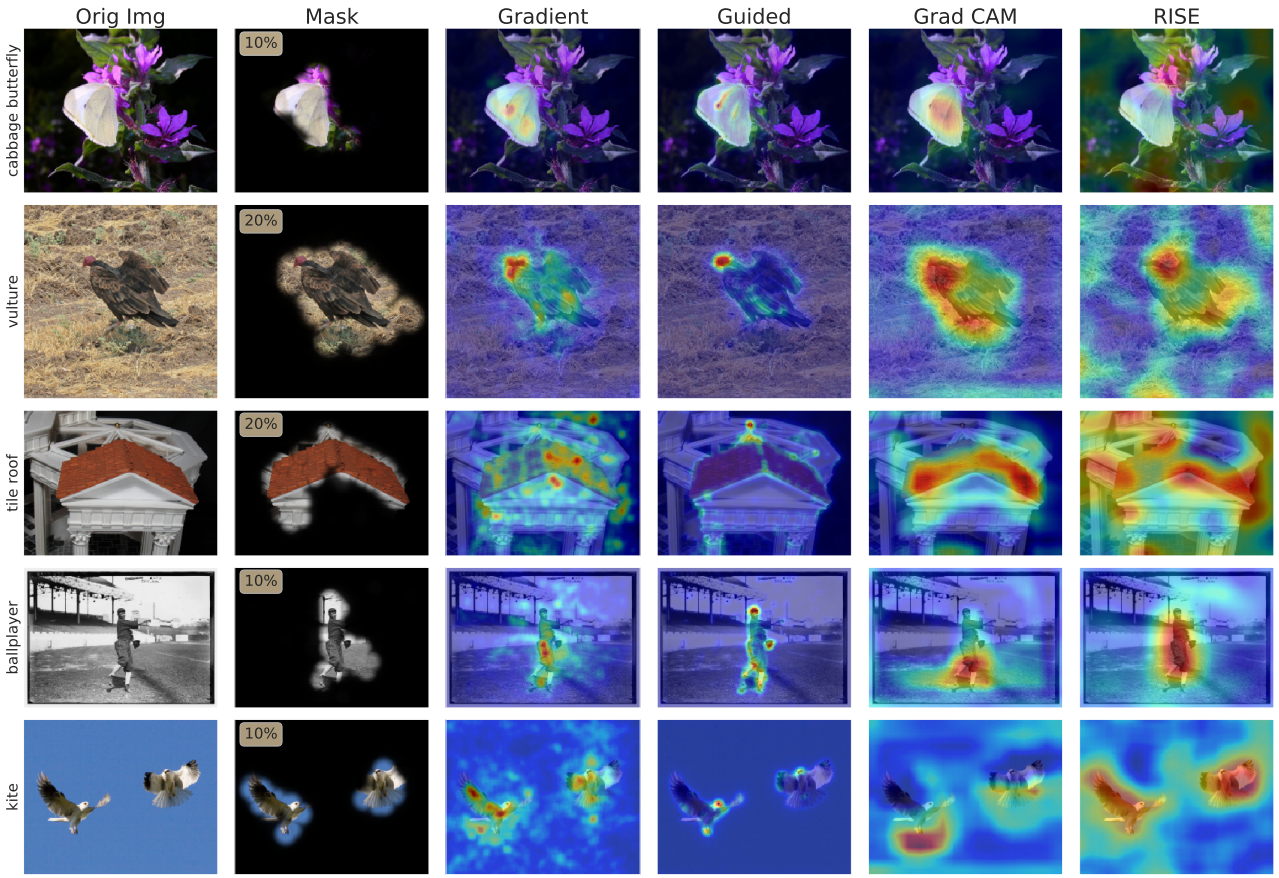


Figure 2: **Comparison with other attribution methods.** We compare our extremal perturbations (optimal area a^* in box) to several popular attribution methods: gradient [23], guided backpropagation [26], Grad-CAM [22], and RISE [20].

of mixing several effects in a single energy term to optimize as in Fong and Vedaldi [7], we introduce the concept of *extremal perturbations*. A perturbation is extremal if it has maximal effect on the network's output among all perturbations of a given, fixed area. Furthermore, the perturbations are regularised by choosing them within family with a minimum guaranteed level of smoothness. In this way, the optimisation is carried over the perturbation effect only, without having to balance several energy terms as done in [7]. Lastly, by sweeping the area parameter, we can study the perturbation's effect w.r.t. its size.

The second contribution is technical and is to provide a concrete algorithm to calculate the extremal perturbations. First, in the optimisation we must *constrain* the perturbation size to be equal to a target value. To this end, we introduce a new ranking-based *area loss* that can enforce these type of constraints in a stable and efficient manner. This loss, which we believe can be beneficial beyond our perturbation analysis, can be interpreted as a hard constraint, similar to a logarithmic barrier, differing from the soft penalty on the area in Fong and Vedaldi [7]. Second, we construct a parametric family of perturbations with a minimum guarantee amount of smoothness. For this, we use the *(smooth)-max-convolution operator* and a *perturbation pyramid*.

As a final contribution, we extend the framework of perturbation analysis to the intermediate activations of a deep neural network rather than its input. This allows us to explore how perturbations can be used beyond spatial, input-level attribution, to channel, intermediate-layer attribution. When combined with existing visualization techniques such as feature inversion [13, 19, 16, 28], we demonstrate how intermediate-layer perturbations can help us understand which channels are salient for classification.

2. Related work

Backpropagation-based methods. Many attribution techniques leverage backpropagation to track information from the network's output back to its input, or an intermediate layer. Since they are based on simple modifications of the backpropagation algorithm, they only require a single forward and backward pass through the model, and are thus efficient. [23]'s gradient method, which uses unmodified backprop, visualizes the derivative of the network's output

w.r.t. the input image. Other works (e.g., DeCovNet [31], Guided Backprop [26], and SmoothGrad [25]) reduce the noise in the gradient signal by tweaking the backprop rules of certain layers. Other methods generate visualizations by either combining gradients, network weights and/or activations at a specific layer (e.g., CAM [33] and Grad-CAM [22]) or further modify the backpropn rules to have a probabilistic or local approximation interpretation (e.g., LRP [3] and Excitation Backprop [32]).

Several papers have shown that some (but not all) backpropagation-based methods produce the same saliency map regardless of the output neuron being analysed [14], or even regardless of network parameters [2]. Thus, such methods may capture average network properties but may not be able to characterise individual outputs or intermediate activations, or in some cases the model parameters.

Perturbation-based methods. Another family of approaches perturbs the inputs to a model and observes resultant changes to the outputs. Occlusion [31] and RISE [20] occlude an image using regular or random occlusions patterns, respectively, and weigh the changes in the output by the occluding patterns. Meaningful perturbations [7] optimize a spatial perturbation mask that maximally affects a model's output. Real-time saliency [5] builds on [7] and learns to predict such a perturbation mask with a second neural network. Other works have leveraged perturbations at the input [24, 30] and intermediate layers [29] to perform weakly and fully supervised localization.

Approximation-based methods. Black-box models can be analyzed by approximating them (locally) with simpler, more interpretable models. The gradient method of [23] and, more explicitly, LIME [21], do so using linear models. Approximations using decision trees or other models are also possible, although less applicable to visual inputs.

Visualizations of intermediate activations. To characterize a filter's behavior, Zeiler and Fergus [31] show dataset examples from the training set that maximally activate that filter. Similarly, activation maximization [23] learns an input image that maximally activates a filter. Feature inversion [13] learns an image that reconstructs a network's intermediate activations while leveraging a natural image prior for visual clarity. Subsequent works tackled the problem of improving the natural image prior for feature inversion and/or activation maximization [28, 19, 16, 18, 17]. Recently, some methods have measured the performance of single [4, 34] and combinations of [11, 8] filter activations on probe tasks like classification and segmentation to identify which filter(s) encode what concepts.

One difficulty in undertaking channel attribution is that, unlike spatial attribution, where a salient image region is naturally interpretable to humans, simply identifying "important channels" is insufficient as they are not naturally interpretable. To address this, we combine the aforementioned visualization techniques with channel attribution.

3. Method

We first summarize the perturbation analysis of [7] and then introduce our extremal perturbations framework.

3.1. Perturbation analysis

Let $\boldsymbol{x} : \Omega \to \mathbb{R}^3$ be a colour image, where $\Omega = \{0, \dots, H-1\} \times \{0, \dots, W-1\}$ is a discrete lattice, and let Φ be a model, such as a convolutional neural network, that maps the image to a scalar output value $\Phi(\boldsymbol{x}) \in \mathbb{R}$. The latter could be an output activation, corresponding to a class prediction score, in a model trained for image classification, or an intermediate activation.

In the following, we investigate which parts of the input $\boldsymbol{x}$ strongly excite the model, causing the response $\Phi(\boldsymbol{x})$ to be large. In particular, we would like to find a *mask* $\boldsymbol{m}$ assigning to each pixel $u \in \Omega$ a value $\boldsymbol{m}(u) \in \{0, 1\}$, where $\boldsymbol{m}(u) = 1$ means that the pixel strongly contributes to the output and $\boldsymbol{m}(u) = 0$ that it does not.

In order to assess the importance of a pixel, we use the mask to induce a local perturbation of the image, denoted $\hat{\boldsymbol{x}} = \boldsymbol{m} \otimes \boldsymbol{x}$. The details of the perturbation model are discussed below, but for now it suffices to say that pixels for which $\boldsymbol{m}(u) = 1$ are preserved, whereas the others are blurred away. The goal is then to find a small subset of pixels that, when preserved, are sufficient to retain a large value of the output $\Phi(\boldsymbol{m} \otimes \boldsymbol{x})$.

Fong and Vedaldi [7] propose to identify such salient pixels by solving an optimization problem of the type:

$$\boldsymbol{m}_{\lambda,\beta} = \operatorname*{argmax}_{\boldsymbol{m}} \Phi(\boldsymbol{m} \otimes \boldsymbol{x}) - \lambda \|\boldsymbol{m}\|_1 - \beta \mathcal{S}(\boldsymbol{m}). \quad (1)$$

The first term encourages the network's response to be large. The second encourages the mask to select a small part of the input image, blurring as many pixels as possible. The third further regularises the smoothness of the mask by penalising irregular shapes.

The problem with this formulation is that the meaning of the trade-off established by optimizing eq. (1) is unclear as the three terms, model response, mask area and mask regularity, are not commensurate. In particular, choosing different λ and β values in eq. (1) will result in different masks without a clear way of comparing them.

3.2. Extremal perturbations

In order to remove the balancing issues with eq. (1), we propose to constrain the area of the mask to a fixed value (as a fraction $a|\Omega|$ of the input image area). Furthermore, we control the smoothness of the mask by choosing it in a fixed set $\mathcal{M}$ of sufficiently smooth functions. Then, we find

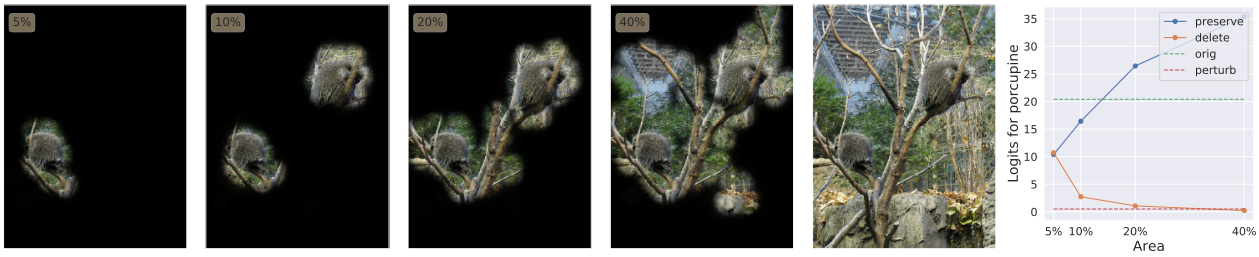


Figure 3: **Extremal perturbations and monotonic effects.** Left: "porcupine" masks computed for several areas a (a in box). Right: $\Phi(\boldsymbol{m}_a \otimes \boldsymbol{x})$ (preservation; blue) and $\Phi((1-\boldsymbol{m}_a)\otimes \boldsymbol{x})$ (deletion; orange) plotted as a function of a. At $a \approx 15\%$ the preserved region scores *higher* than preserving the entire image (green). At $a \approx 20\%$, perturbing the complementary region scores *similarly* to fully perturbing the entire image (red).

the mask of that size that maximizes the model's output:

$$\boldsymbol{m}_a = \operatorname*{argmax}_{\boldsymbol{m}:\ \|\boldsymbol{m}\|_1 = a|\Omega|,\ \boldsymbol{m}\in\mathcal{M}} \Phi(\boldsymbol{m}\otimes\boldsymbol{x}). \tag{2}$$

Note that the resulting mask is a function of the chosen area a only. With this, we can define the concept of *extremal perturbation* as follows. Consider a lower bound Φ_0 on the model's output (for example we may set $\Phi_0 = \tau\Phi(\boldsymbol{x})$ to be a fraction τ of the model's output on the unperturbed image). Then, we search for the *smallest mask* that achieves at least this output level. This amounts to sweeping the area parameter a in eq. (2) to find

$$a^* = \min\{a : \Phi(\boldsymbol{m}_a \otimes \boldsymbol{x}) \geq \Phi_0\}. \tag{3}$$

The mask $\boldsymbol{m}_{a^*}$ is extremal because preserving a smaller portion of the input image is not sufficient to excite the network's response above Φ_0. This is illustrated in fig. 3.

Interpretation. An extremal perturbation is a single mask $\boldsymbol{m}_{a^*}$ that results in a large model response, in the sense that $\Phi(\boldsymbol{m}_{a^*}\otimes\boldsymbol{x}) \geq \Phi_0$. However, due to extremality, we *also* know that any smaller mask does not result in an equally large response: $\forall \boldsymbol{m}:\ \|\boldsymbol{m}\|_1 < \|\boldsymbol{m}_{a^*}\|_1 \ \Rightarrow\ \Phi(\boldsymbol{m}\otimes\boldsymbol{x}) < \Phi_0$. Hence, a single extremal mask is informative because it characterises a *whole family* of input perturbations.

This connects extremal perturbations to methods like [21, 7], which explain a network by finding a succinct and interpretable description of its input-output mapping. For example, the gradient [23] and LIME [21] approximate the network locally around an input $\boldsymbol{x}$ using the Taylor expansion $\Phi(\boldsymbol{x}') \approx \langle \nabla\Phi(\boldsymbol{x}), \boldsymbol{x}'-\boldsymbol{x}\rangle + \Phi(\boldsymbol{x})$; their explanation is the gradient $\nabla\Phi(\boldsymbol{x})$ and their perturbations span a neighbourhood of $\boldsymbol{x}$.

Preservation vs deletion. Formulation (2) is analogous to what [7] calls the "preservation game" as the goal is to find a mask that preserves (maximises) the model's response. We also consider their "deletion game" obtaining by optimising $\Phi((1-\boldsymbol{m})\otimes\boldsymbol{x})$ in eq. (2), so that the goal is to suppress the response when looking outside the mask, and the hybrid [5], obtained by optimising $\Phi(\boldsymbol{m}\otimes\boldsymbol{x}) - \Phi((1-\boldsymbol{m})\otimes\boldsymbol{x})$, where the goal is to simultaneously preserve the response inside the mask and suppress it outside

3.3. Area constraint

Enforcing the area constraint in eq. (2) is non-trivial; here, we present an effective approach to do so (other approaches like [10] do not encourage binary masks). First, since we would like to optimize eq. (2) using a gradient-based method, we relax the mask to span the full range $[0, 1]$. Then, a possible approach would be to count how many values $\boldsymbol{m}(u)$ are sufficiently close to the value 1 and penalize masks for which this count deviates from the target value $a|\Omega|$. However, this approach requires soft-counting, with a corresponding tunable parameter for binning.

In order to avoid such difficulties, we propose instead to *vectorize and sort* in non-decreasing order the values of the mask $\boldsymbol{m}$, resulting in a vector $\operatorname{vecsort}(\boldsymbol{m}) \in [0,1]^{|\Omega|}$. If the mask $\boldsymbol{m}$ satisfies the area constraint exactly, then the output of $\operatorname{vecsort}(\boldsymbol{m})$ is a vector $\mathbf{r}_a \in [0,1]^{|\Omega|}$ consisting of $(1-a)|\Omega|$ zeros followed by $a|\Omega|$ ones. This is captured by the regularization term: $R_a(\boldsymbol{m}) = \|\operatorname{vecsort}(\boldsymbol{m}) - \mathbf{r}_a\|^2$. We can then rewrite eq. (2) as

$$\boldsymbol{m}_a = \operatorname*{argmax}_{\boldsymbol{m}\in\mathcal{M}} \Phi(\boldsymbol{m}\otimes\boldsymbol{x}) - \lambda R_a(\boldsymbol{m}). \tag{4}$$

Note that we have reintroduced a weighting factor λ in the formulation, so on a glance we have lost the advantage of formulation (2) over the one of eq. (1). In fact, this is not the case: during optimization we simply set λ to be as large as numerics allow it as we expect the area constraint to be (nearly) exactly satisfied; similarly to a logarithmic barrier, λ then has little effect on which mask $\boldsymbol{m}_a$ is found.

3.4. Perturbation operator

In this section we define the perturbation operator $\boldsymbol{m}\otimes\boldsymbol{x}$. To do so, consider a *local perturbation operator* $\pi(\boldsymbol{x}; u, \sigma) \in \mathbb{R}^3$ that applies a perturbation of intensity $\sigma \geq 0$ to pixel $u \in \Omega$. We assume that the lowest intensity $\sigma = 0$ corresponds to no perturbation, i.e. $\pi(\boldsymbol{x}; u, 0) =$

$\boldsymbol{x}(u)$. Here we use as perturbations the Gaussian blur[2]

$$\pi_g(\boldsymbol{x}; u, \sigma) = \frac{\sum_{v\in\Omega} g_\sigma(u-v)\boldsymbol{x}(v)}{\sum_{v\in\Omega} g_\sigma(u-v)}, \quad g_\sigma(u) = e^{-\frac{\|u\|^2}{2\sigma^2}}.$$

The mask $\boldsymbol{m}$ then doses the perturbation spatially: $(\boldsymbol{m}\otimes\boldsymbol{x})(u) = \pi(\boldsymbol{x}; u, \sigma_{\max}\cdot(1-\boldsymbol{m}(u)))$ where $\sigma_{\max}$ is the maximum perturbation intensity.[3]

3.5. Smooth masks

Next, we define the space of smooth masks $\mathcal{M}$. For this, we consider an auxiliary mask $\bar{\boldsymbol{m}} : \Omega \to [0,1]$. Given that the range of $\bar{\boldsymbol{m}}$ is bounded, we can obtain a smooth mask $\boldsymbol{m}$ by convolving $\bar{\boldsymbol{m}}$ by a Gaussian or similar kernel $\boldsymbol{k} : \Omega \to \mathbb{R}_+$[4] via the typical *convolution operator*:

$$\boldsymbol{m}(u) = Z^{-1}\sum_{v\in\Omega} \boldsymbol{k}(u-v)\bar{\boldsymbol{m}}(v) \tag{5}$$

where Z normalizes the kernel to sum to one. However, this has the issue that setting $\bar{\boldsymbol{m}}(u) = 1$ does not necessarily result in $\boldsymbol{m}(u) = 1$ after filtering, and we would like our final mask to be (close to) binary.

To address this issue, we consider the *max-convolution operator*:

$$\boldsymbol{m}(u) = \max_{v\in\Omega} \boldsymbol{k}(u-v)\bar{\boldsymbol{m}}(v). \tag{6}$$

This solves the issue above while at the same time guaranteeing that the smoothed mask does not change faster than the smoothing kernel, as shown in the following lemma (proof in supp. mat.).

Lemma 1. *Consider functions $\bar{\boldsymbol{m}}, \boldsymbol{k} : \Omega \to [0,1]$ and let $\boldsymbol{m}$ be defined as in eq. (6). If $\boldsymbol{k}(0) = 1$, then $\bar{\boldsymbol{m}}(u) \le \boldsymbol{m}(u) \le 1$ for all $u \in \Omega$; in particular, if $\bar{\boldsymbol{m}}(u) = 1$, then $\boldsymbol{m}(u) = 1$ as well. Furthermore, if $\boldsymbol{k}$ is Lipschitz continuous with constant K, then $\boldsymbol{m}$ is also Lipschitz continuous with a constant at most as large as K.*

The $\max$ operator in eq. (6) yields sparse gradients. Thus, to facilitate optimization, we introduce the *smooth max* operator[5], smax, to replace the $\max$ operator. For a function $f(u), u \in \Omega$ and temperature $T > 0$:

$$\operatorname*{smax}_{u\in\Omega;T} f(u) = \frac{\sum_{u\in\Omega} f(u)\exp f(u)/T}{\sum_{u\in\Omega}\exp f(u)/T} \tag{7}$$

[2] Another choice is the fade-to-black perturbation which, for $0 \le \sigma \le 1$, is given by $\pi_f(\boldsymbol{x}; u, \sigma) = (1-\sigma)\cdot\boldsymbol{x}(u)$.

[3] For efficiency, this is implemented by generating a *perturbation pyramid* $\pi(\boldsymbol{x}; \cdot, \sigma_{\max}\cdot l/L)$, $l = 0, \dots, L$ that contains $L+1$ progressively more perturbed versions of the image. Then $\boldsymbol{m}\otimes\boldsymbol{x}$ can be computed via bilinear interpolation by using $(u, \boldsymbol{m}(u))$ as an indices in the pyramid.

[4] It is easy to show that in this case the derivative of the smoothed mask $\||\nabla(\boldsymbol{k} * \bar{\boldsymbol{m}})\|\| \le \|\nabla\boldsymbol{k}\|$ is always less than the one of the kernel.

[5] Not to be confused with the softmax with temperature, as in [9].

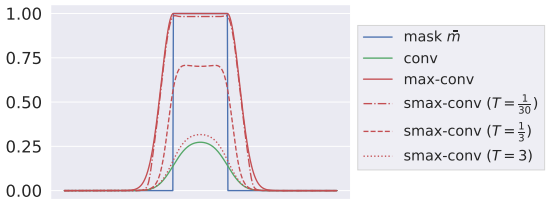


Figure 4: **Convolution operators for smooth masks.** Gaussian smoothing a mask (blue) with the typical convolution operator yields a dampened, smooth mask (green). Our max-convolution operator mitigates this effect while still smoothing (red solid). Our smax operator, which yields more distributed gradients than $\max$, varies between the other two convolution operators (red dotted).

The smax operator smoothly varies from behaving like the mean operator in eq. (5) as $T \to \infty$ to behaving like the $\max$ operator as $T \to 0$ (see fig. 4). This operator is used instead of $\max$ in eq. (6).

Implementation details. In practice, we use a smaller parameterization mask $\bar{\boldsymbol{m}}$ defined on a lattice $\bar{\Omega} = \{0, \dots, \bar{H}-1\}\times\{0, \dots, \bar{W}-1\}$, where the full-resolution mask $\boldsymbol{m}$ has dimensions $H = \rho\bar{H}$ and $W = \rho\bar{W}$. We then modify (6) to perform upsampling in the same way as the standard convolution transpose operator.

4. Experiments

Implementation details. Unless otherwise noted, all visualizations use the ImageNet validation set, the VGG16 network and the preservation formulation (Sec. 3.2). Specifically, $\Phi(\boldsymbol{x})$ is the classification score (before softmax) that network associates to the ground-truth class in the image. Masks are computed for areas $a \in \{0.05, 0.1, 0.2, 0.4, 0.6, 0.8\}$. To determine the optimal area a^* of the extremal perturbations (3), we set the threshold $\Phi_0 = \Phi(\boldsymbol{x})$ (which is the score on the unperturbed image).

Masks are optimised using SGD, initializing them with all ones (everything preserved). SGD uses momentum 0.9 and 1600 iterations. λ is set to 300 and doubled at 1/3 and 2/3 of the iterations and, in eq. (7), $1/T \approx 20$. Before upsampling, the kernel $\boldsymbol{k}(u) = k(\|u\|)$ is a radial basis function with profile $k(z) = \exp\left(\max\{0, z-1\}^2/4\right)$, chosen so that neighbour disks are centrally flat and then decay smoothly.

4.1. Qualitative comparison

Figure 2 shows a qualitative comparison between our method and others. We see that our criterion of $\Phi_0 = \Phi(x)$ chooses fairly well-localized masks in most cases. Masks tend to cover objects tightly, are sharp, and clearly identify a region of interest in the image. Figure 5 shows what the network considered to be most discriminative ($a = 5\%$; e.g.,

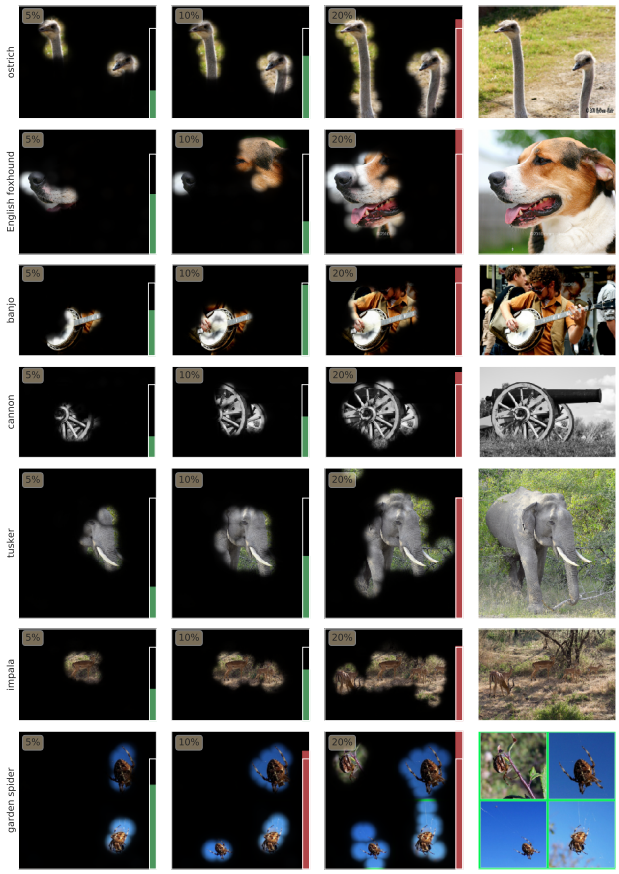


Figure 5: **Area growth.** Although each mask is learned independently, these plots highlight what the network considers to be most discriminative and complete. The bar graph visualizes $\Phi(\boldsymbol{m}_a \odot \boldsymbol{x})$ as a normalized fraction of $\Phi_0 = \Phi(\boldsymbol{x})$ (and saturates after exceeding Φ_0 by 25%).

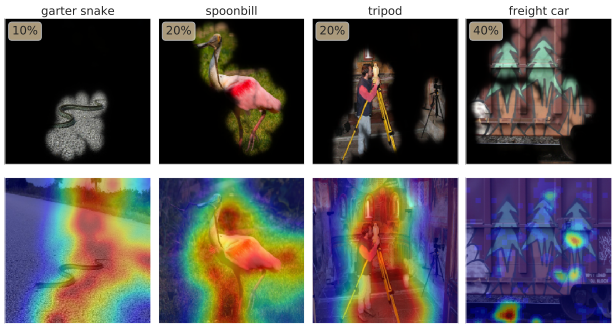


Figure 6: **Comparison with [7].** Our extremal perturbations (top) vs. masks from Fong and Vedaldi [7] (bottom).

banjo fret board, elephant tusk) and complete ($a = 20\%$) as the area increases. We notice that evidence from several objects accumulates monotonically (e.g., impala and spider) and that foreground (e.g., ostrich) or discriminative parts (e.g., dog's nose) are usually sufficient.

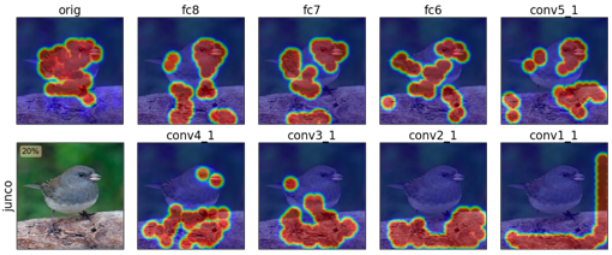


Figure 7: **Sanity check [2].** Model weights are progressively randomized from fc8 to conv1_1 in VGG16, demonstrating our method's sensitivity to model weights.

	VOC07 Test (All/Diff)		*COCO14 Val (All/Diff)*	
Method	*VGG16*	*ResNet50*	*VGG16*	*ResNet50*
Cntr.	69.6/42.4	69.6/42.4	27.8/19.5	27.8/19.5
Grad	76.3/56.9	72.3/56.8	37.7/31.4	35.0/29.4
DConv	67.5/44.2	68.6/44.7	30.7/23.0	30.0/21.9
Guid.	75.9/53.0	77.2/59.4	39.1/31.4	42.1/35.3
MWP	77.1/56.6	84.4/70.8	39.8/32.8	49.6/43.9
cMWP	79.9/66.5	**90.7**/82.1	49.7/44.3	**58.5/53.6**
RISE*	86.9/75.1	86.4/78.8	50.8/45.3	54.7/50.0
GCAM	86.6/74.0	90.4/**82.3**	**54.2/49.0**	57.3/52.3
Ours*	**88.0/76.1**	88.9/78.7	51.5/45.9	56.5/51.5

Table 1: **Pointing game.** Mean accuracy on the pointing game over the full data splits and a subset of difficult images (defined in [32]). Results from PyTorch re-implementation using TorchRay package (* denotes average over 3 runs).

In fig. 6, we compare our masks to those of Fong and Vedaldi [7]. The stability offered by controlling the area of the perturbation is obvious in these examples. Lastly, we visualize a sanity check proposed in Adebayo et al. [2] in fig. 7 (we use the "hybrid" formulation). Unlike other backprop-based methods, our visualizations become significantly different upon weight randomization (see supp. mat. for more qualitative examples).

4.2. Pointing game

A common approach to evaluate attribution methods is to correlate their output with semantic annotations in images. Here we consider in particular the pointing game of Zhang et al. [32]. For this, an attribution method is used to compute a saliency map for each of the object classes present in the image. One scores a hit if the maximum point in the saliency map is contained within the object; The overall accuracy is the number of hits over number of hits plus misses.

Table 1 shows results for this metric and compares our method against the most relevant work in the literature on PASCAL VOC [6] (using the 2007 test set of 4952 images) and COCO [12] (using the 2014 validation set of $\approx 50k$ im-

ages). We see that our method is competitive with VGG16 and ResNet50 networks. In contrast, Fong and Vedaldi's [7] was not competitive in this benchmark (although they reported results using GoogLeNet).

Implementation details. Since our masks are binary, there is no well defined maximum point. To apply our method to the pointing game, we thus run it for areas $\{0.025, 0.05, 0.1, 0.2\}$ for PASCAL and $\{0.018, 0.025, 0.05, 0.1\}$ for COCO (due to the smaller objects in this dataset). The binary masks are summed and a Gaussian filter with standard deviation equal to 9% of the shorter side of the image applied to the result to convert it to a saliency map. We use the original Caffe models of [32] converted to PyTorch and use the preservation formulation of our method.

4.3. Monotonicity of visual evidence

Eq. (2) implements the "preservation game" and searches for regions of a given area that *maximally activate* the networks' output. When this output is the confidence score for a class, we hypothesise that hiding evidence from the network would only make the confidence lower, i.e., we would expect the effect of maximal perturbations to be ordered consistently with their size:

$$a_1 \leq a_2 \;\Rightarrow\; \Phi(\boldsymbol{m}_{a_1} \otimes \boldsymbol{x}) \leq \Phi(\boldsymbol{m}_{a_2} \otimes \boldsymbol{x}) \qquad (8)$$

However, this may not always be the case. In order to quantify the frequency of this effect, we test whether eq. (8) holds for all $a_1, a_2 < a^*$, where a^* is the optimal area of the extremal perturbation (eq. (3), where $\Phi_0 = \Phi(\boldsymbol{x})$). Empirically, we found that this holds for 98.45% of ImageNet validation images, which indicates that evidence is in most cases integrated monotonically by the network.

More generally, our perturbations allow us to sort and investigate how information is integrated by the model in order of importance. This is shown in several examples in fig. 5 where, as the area of the mask is progressively increased, different parts of the objects are prioritised.

5. Attribution at intermediate layers

Lastly, we extend extremal perturbations to the *direct* study of the intermediate layers in neural networks. This allows us to highlight a novel use case of our area loss and introduce a new technique for understanding which channels are salient for classification.

As an illustration, we consider in particular channel-wise perturbations. Let $\Phi_l(x) \in \mathbb{R}^{K_l \times H_l \times W_l}$ be the intermediate representation computed by a neural network Φ up to layer l and let $\Phi_{l+} : \mathbb{R}^{K_l \times H_l \times W_l} \to \mathbb{R}$ represent the rest of model, from layer l to the last layer. We then re-formulate the preservation game from eq. (4) as:

$$\boldsymbol{m}_a = \operatorname*{argmax}_{\boldsymbol{m}} \Phi_{l+}(\boldsymbol{m} \otimes \Phi_l(\boldsymbol{x})) - \lambda R_a(\boldsymbol{m}). \qquad (9)$$

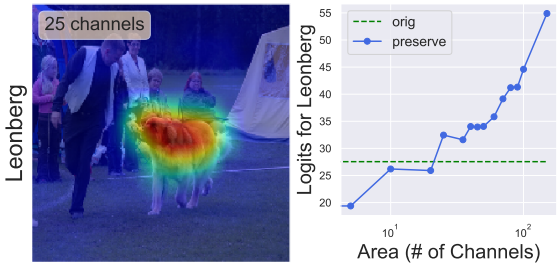


Figure 8: **Attribution at intermediate layers.** Left: This is visualization (eq. (11)) of the optimal channel attribution mask $\boldsymbol{m}_{a^*}$, where $a^* = 25$ channels, as defined in eq. (10). Right: This plot shows that class score monotonically increases as the area (as the number of channels) increases.

Here, the mask $\boldsymbol{m} \in [0,1]^{K_l}$ is a vector with one element per channel which element-wise multiplies with the activations $\Phi_l(\boldsymbol{x})$, broadcasting values along the spatial dimensions. Then, the extremal perturbation $\boldsymbol{m}_{a^*}$ is selected by choosing the optimal area

$$a^* = \min\{a : \Phi_{l+}(\boldsymbol{m}_a \otimes \Phi_l(\boldsymbol{x})) \geq \Phi_0\}. \qquad (10)$$

We assume that the output Φ_{l+} is the pre-softmax score for a certain image class and we set the $\Phi_0 = \Phi(\boldsymbol{x})$ to be the model's predicted value on the unperturbed input (fig. 8).

Implementation details. In these experiments, we use GoogLeNet [27] and focus on layer l =`inception4d`, where $H_l = 14, W_l = 14, K_l = 528$. We optimize eq. (9) for 300 iterations with a learning rate of 10^{-2}. The parameter λ linearly increases from 0 to 1500 during the first 150 iterations, after which $\lambda = 1500$ stays constant. We generate channel-wise perturbation masks for areas $a \in \{1, 5, 10, 20, 25, 35, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, 528\}$, where a denotes the number of channels preserved.

The saliency heatmaps in fig. 8 and fig. 9 for channel-wise attribution are generated by summing over the channel dimension the element-wise product of the channel attribution mask and activation tensor at layer l:

$$\boldsymbol{v} = \sum_{k \in K} \boldsymbol{m}^k_{a^*} \otimes \Phi^k_l(\boldsymbol{x}) \qquad (11)$$

5.1. Visualizing per-instance channel attribution

Unlike per-instance input-level spatial attribution, which can be visualized using a heatmap, per-instance intermediate channel attribution is more difficult to visualize because simply identifying important channels is not necessarily human-interpretable. To address this problem, we use feature inversion [15, 19] to find an image that maximises the dot product of the channel attribution vector and the activation tensor (see [19] for more details):

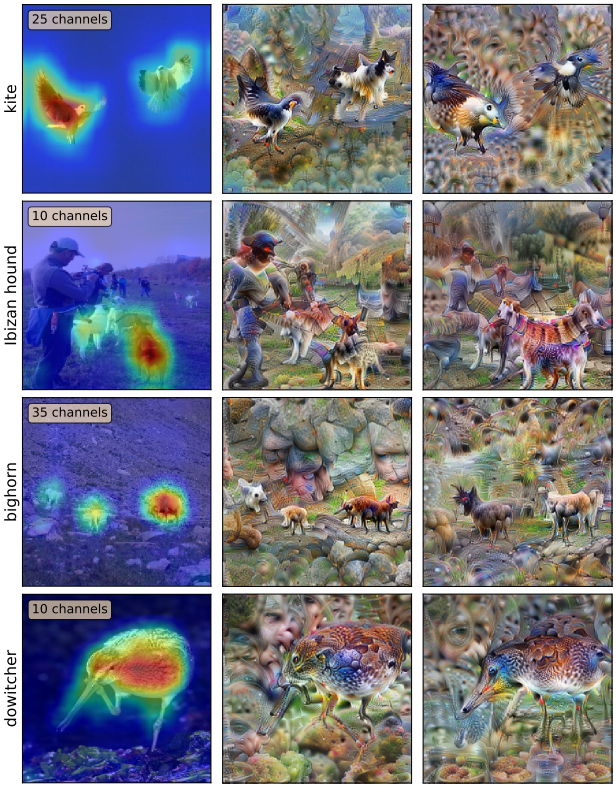


Figure 9: **Per-instance channel attribution visualization.** Left: input image overlaid with channel saliency map (eq. (11)). Middle: feature inversion of original activation tensor. Right: feature inversion of activation tensor perturbed by optimal channel mask $\boldsymbol{m}_{a^*}$. By comparing the difference in feature inversions between un-perturbed (middle) and perturbed activations (right), we can identify the salient features that our method highlights.

$$I^* = \underset{I}{\operatorname{argmax}} \{(\boldsymbol{m}_{a^*} \otimes \Phi_l(\boldsymbol{x})) \cdot \Phi_l(I)\} \tag{12}$$

where $\boldsymbol{m}_a^*$ is optimal channel attribution mask at layer l for input image $\boldsymbol{x}$ and $\Phi_l(I)$ is the activation tensor at layer l for image I, the image we are learning.

This inverted image allows us to identify the parts of the input image that are salient for a particular image to be correctly classified by a model. We can compare the feature inversions of activation tensors perturbed with channel mask (right column in fig. 9) to the inversions of original, unperturbed activation tensors (middle column) to get a clear idea of the most discriminative features of an image.

Since the masks are roughly binary, multiplying $\boldsymbol{m}_a^*$ with the activation tensor $\Phi_l(\boldsymbol{x})$ in eq. (12) zeroes out non-salient channels. Thus, the differences in the feature inversions of original and perturbed activations in fig. 9 highlight regions encoded by salient channels identified in our attribution masks (i.e., the channels that are not zeroed out in eq. (12)).

Figure 10: **Discovery of salient, class-specific channels.** By analyzing $\bar{\boldsymbol{m}}_c$, the average over all $\boldsymbol{m}_{a^*}$ for class c (see Sec. 5.2), we automatically find salient, class-specific channels like these. First column: channel feature inversions; all others: dataset examples.

5.2. Visualizing per-class channel attribution

We can also use channel attribution to identify important, class-specific channels. In contrast to other methods, which explicitly aim to find class-specific channels and/or directions at a global level [8, 11, 34], we are able to similarly do so "for free" using only our per-instance channel attribution masks. After estimating an optimal masks $\boldsymbol{m}_{a^*}$ for all ImageNet validation images, we then create a per-class attribution mask $\bar{\boldsymbol{m}}_c \in [0,1]^K$ by averaging the optimal masks of all images in a given class c. Then, we can identify the most important channel for a given class as follows: $k_c^* = \operatorname{argmax}_{k \in K} \bar{\boldsymbol{m}}_c^k$. In fig. 10, we visualize two such channels via feature inversions. Qualitatively, these feature inversions of channels k_c^* are highly class-specific.

6. Conclusion

We have introduced the framework of extremal perturbation analysis, which avoids some of the issues of prior work that use perturbations to analyse neural networks. We have also presented a few technical contributions to compute such extremal perturbation. Among those, the rank-order area constraint can have several other applications in machine learning beyond the computation of extremal perturbations. We have extended the perturbations framework to perturbing intermediate activations and used this to explore a number of properties of the representation captured by a model. In particular, we have visualized, likely for the first time, the difference between perturbed and unperturbed activations using a representation inversion technique. Lastly, we released TorchRay [1], a PyTorch interpretability library in which we've re-implemented popular methods and benchmarks to encourage reproducible research.

Acknowledgements. We are grateful for support from the Open Philanthropy Project (R.F.), the Rhodes Trust (M.P.), and ESPRC EP/L015897/1 (CDT in Autonomous Intelligent Machines and Systems) (M.P). We also thank Jianming Zhang and Samuel Albanie for help on re-implementing the Pointing Game [32] in PyTorch.

References

[1] Torchray. github.com/facebookresearch/TorchRay, 2019. 8

[2] Julius Adebayo, Justin Gilmer, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Proc. NeurIPS*, 2018. 3, 6

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015. 1, 3

[4] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proc. CVPR*, 2017. 3

[5] Piotr Dabkowski and Yarin Gal. Real time image saliency for black box classifiers. In *Proc. NeurIPS*, 2017. 1, 3, 4

[6] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, Jan. 2015. 6

[7] Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proc. ICCV*, 2017. 1, 2, 3, 4, 6, 7

[8] Ruth Fong and Andrea Vedaldi. Net2Vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Proc. CVPR*, 2018. 3, 8

[9] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv*, 2015. 5

[10] Hoel Kervadec, Jose Dolz, Meng Tang, Eric Granger, Yuri Boykov, and Ismail Ben Ayed. Constrained-cnn losses for weakly supervised segmentation. *Medical image analysis*, 54:88–99, 2019. 4

[11] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *Proc. ICML*, 2017. 3, 8

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proc. ECCV*, 2014. 6

[13] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proc. CVPR*, 2015. 2, 3

[14] Aravindh Mahendran and Andrea Vedaldi. Salient deconvolutional networks. In *Proc. ECCV*, 2016. 3

[15] Aravindh Mahendran and Andrea Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *IJCV*, 2016. 7

[16] Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 3(7):e12, 2018. 2, 3

[17] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. CVPR*, 2017. 3

[18] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Proc. NeurIPS*, 2016. 3

[19] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. 2, 3, 7

[20] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *Proc. BMVC*, 2018. 1, 2, 3

[21] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" explaining the predictions of any classifier. In *Proc. KDD*, 2016. 3, 4

[22] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *Proc. ICCV*, 2017. 1, 2, 3

[23] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Proc. ICLR workshop*, 2014. 1, 2, 3, 4

[24] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *Proc. ICCV*, 2017. 3

[25] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv*, 2017. 3

[26] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. 2015. 1, 2, 3

[27] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 7

[28] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proc. CVPR*, 2018. 2, 3

[29] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proc. CVPR*, 2017. 3

[30] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proc. CVPR*, 2017. 3

[31] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proc. ECCV*, 2014. 1, 3

[32] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018. 1, 3, 6, 7, 8

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proc. CVPR*, 2016. 3

[34] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation. *arXiv*, 2018. 3, 8

A. Implementation details

A.1. Generating smooth masks

We implement the equation:

$$\hat{m}(u) = \operatorname{pool}_i g(u - u_i) m_i$$

Here $i = 0, \dots, N-1$ are samples of the input mask, $u = 0, \dots, W-1$ samples ouf the output mask, and u_i is the mapping between input and output samples, $u_i = ai + b$. We assume that the kernel k has a "radius" σ, in the sense that only samples $|u - u_i| \leq \sigma$ matter.

In order to compute this expansion fast, we use the unpool operator. In order to do so, unpool is applied to m with window $K = 2R + 1$ and padding P. This results in the signal

$$m'_{k,i} = m_{i+k-P}, \quad 0 \leq i \leq W' - 1, \quad 0 \leq k \leq K - 1, \qquad W' = N - K + 2P + 1.$$

We then use nearest-neighbour upsampling in order to bring this signal in line with the resolution of the output:

$$m''_{k,u} = m_{\lfloor \frac{u}{s} \rfloor + k - P}, \quad 0 \leq u \leq W'' - 1, \quad 0 \leq k \leq K - 1.$$

Here the upsampling factor is given by $s = W''/W'$. In PyTorch, we specify upsampling via the input size W' and the output size W'', so we need to choose W'' appropriately.

To conclude, we do so as follows. We choose a σ (kernel width in image pixels) and and s (kernel step in pixels). We also choose a margin $b \geq 0$ to avoid border effects and set $a = s$. With this, we see that computing $\hat{m}(u)$ requires samples:

$$u - \sigma \leq u_i \leq u + \sigma \quad \Leftrightarrow \quad \frac{u}{s} - \frac{\sigma + b}{s} \leq i \leq \frac{u}{s} + \frac{\sigma - b}{s}.$$

On the other hand, at location u in $m''_{k,u}$ we have pooled samples m_i for which:

$$\left\lfloor \frac{u}{s} \right\rfloor - P \leq i \leq \left\lfloor \frac{u}{s} \right\rfloor + K - 1 - P.$$

Hence we require

$$\left\lfloor \frac{u}{s} \right\rfloor - P \leq \frac{u}{s} - \frac{\sigma + b}{s} \quad \Rightarrow \quad P \geq \frac{\sigma + b}{s} + \left\lfloor \frac{u}{s} \right\rfloor - \frac{u}{s}.$$

Conservatively, we take:

$$P = 1 + \left\lceil \frac{\sigma + b}{s} \right\rceil$$

The other bound is:

$$\frac{u}{s} + \frac{\sigma - b}{s} \leq \left\lfloor \frac{u}{s} \right\rfloor + K - 1 - P.$$

Hence:

$$K \geq \frac{u}{s} - \left\lfloor \frac{u}{s} \right\rfloor + \frac{\sigma - b}{s} + P + 1$$

Hence, conservatively we take:

$$K = 3 + \left\lceil \frac{\sigma + b}{s} \right\rceil + \left\lceil \frac{\sigma - b}{s} \right\rceil.$$

Since $K = 2R + 1$ and $b \approx \sigma$, we set

$$R = 1 + \left\lceil \frac{\sigma}{s} \right\rceil.$$

In this way, we obtain a pooled mask:

$$\bar{m}(u) = \operatorname*{pool}_i g(u - u_i)\, m_i = \operatorname*{pool}_{0 \leq k \leq K-1} g_{k,u} m''_{k,u},$$

where

$$g_{k,u} = g(u - \bar{u}(u,k)), \quad \bar{u}(u,k) = \left\lfloor \frac{u}{s} \right\rfloor + k - P.$$

Hence, the steps are: given the input mask parameters m_i, use unpooling to obtain $m'_{k,i}$ and then upsampling to obtain $m''_{k,u}$. Then use the equation above to pool using a precomputed weights $g_{k,u}$.

Generally, the input to the mask generator are: s, σ and the desired mask $\hat{m}(u)$ width W. So far, we have obtained a mask $\bar{m}(u)$ with with W'', where $W'' = sW'$ is chosen to obtain the correct scaling factor and $W' = N - K + 2P + 1$. As a rule of thumb, we set $N = \lceil W/s \rceil$ in order to spread the N samples at regular interval over the full stretch W. We then set R, K, P, W' and W'' according to the formulas above. Once $\bar{m}(u)$ is obtained, we take a W-sized crop shifted by b pixels to obtain the final mask $\hat{m}(u)$.

B. Supplementary Materials

The full supplementary materials for this paper can be found at ruthcfong.github.io/files/fong19_extremal_supps.pdf.

8

Labelling unlabelled videos from scratch with multi-modal self-supervision

This work was presented at the conference of Advances in Neural Information Processing Systems (NeurIPS) 2020.

Labelling unlabelled videos from scratch with multi-modal self-supervision

Yuki M. Asano[1*] **Mandela Patrick**[1,2*] **Christian Rupprecht**[1] **Andrea Vedaldi**[1,2]
[1] Visual Geometry Group, University of Oxford
yuki@robots.ox.ac.uk
[2] Facebook AI Research
mandelapatrick@fb.com

Abstract

A large part of the current success of deep learning lies in the effectiveness of data – more precisely: labelled data. Yet, labelling a dataset with human annotation continues to carry high costs, especially for videos. While in the image domain, recent methods have allowed to generate meaningful (pseudo-) labels for unlabelled datasets without supervision, this development is missing for the video domain where learning feature representations is the current focus. In this work, we a) show that unsupervised labelling of a video dataset does not come for free from strong feature encoders and b) propose a novel clustering method that allows pseudo-labelling of a video dataset without any human annotations, by leveraging the natural correspondence between the audio and visual modalities. An extensive analysis shows that the resulting clusters have high semantic overlap to ground truth human labels. We further introduce the first benchmarking results on unsupervised labelling of common video datasets Kinetics, Kinetics-Sound, VGG-Sound and AVE[2].

1 Introduction

One of the key tasks in machine learning is to convert continuous perceptual data such as images and videos into a symbolic representation, assigning discrete labels to it. This task is generally formulated as *clustering* [31]. For images, recent contributions such as [6, 13, 37, 72] have obtained good results by combining clustering and representation learning. However, progress has been more limited for videos, which pose unique challenges and opportunities. Compared to images, videos are much more expensive to annotate; at the same time, they contain more information, including a temporal dimension and two modalities, aural and visual, which can be exploited for better clustering. In this paper, we are thus interested in developing methods to *cluster video datasets without manual supervision*, potentially reducing the cost and amount of manual labelling required for video data.

Just as for most tasks in machine learning, clustering can be greatly facilitated by extracting a suitable *representation* of the data. However, representations are usually learned by means of manually supplied labels, which we wish to avoid. Inspired by [79], we note that a solution is to consider one of the recent state-of-the-art self-supervised representation learning methods and apply an off-the-shelf clustering algorithm *post-hoc*. With this, we show that we can obtain very strong baselines for clustering videos.

[*] Joint first authors
[2] Code will be made available at https://github.com/facebookresearch/selavi

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Still, this begs the question of whether even better performance could be obtained by *simultaneously learning to cluster and represent* video data. Our main contribution is to answer this question affirmatively and thus to show that *good clusters do not come for free from good representations*.

In order to do so, we consider the recent method SeLa [6], which learns clusters and representations for still images by solving an optimal transport problem, and substantially improve it to work with multi-modal data. We do this in three ways. First, we relax the assumption made in [6] that clusters are equally probable; this is not the case for semantic video labels, which tend to have a highly-skewed distribution [1, 29, 41], and extend the algorithm accordingly. Second, we account for the multi-modal nature of video data, by formulating the extraction of audio and visual information from a video as a form of data augmentation, thus learning a clustering function which is invariant to such augmentations. For this to work well, we also propose a new initialization scheme that synchronizes the different modalities before clustering begins. This encourages clusters to be more abstract and thus 'semantic' and learns a redundant clustering function which can be computed robustly from either modality (this is useful when a modality is unreliable, because of noise or compression). Third, since clustering is inherently ambiguous, we propose to learn multiple clustering functions in parallel, while keeping them orthogonal, in order to cover a wider space of valid solutions.

With these technical improvements, our method for Self-Labelling Videos (SeLaVi) substantially outperforms the post-hoc approach [79], SeLa [6] applied to video frames, as well as a recent multi-modal clustering-based representation learning method, XDC [2]. We evaluate our method by testing how well the automatically learned clusters match manually annotated labels in four different video datasets: VGG-Sound [17], AVE [68], Kinetics [41] and Kinetics-Sound [3]. We show that our proposed model results in substantially better clustering performance than alternatives. For example, our method can perfectly group 32% of the videos in the VGG-Sound dataset and 55% in the AVE dataset without using any labels during training. Furthermore, we show that, while some clusters do not align with the ground truth classes, they are generally semantically meaningful (e.g. they contain similar background music) and provide an interactive cluster visualization[3].

In a nutshell, our key contributions are: **(i)** establishing video clustering benchmark results on four datasets for which labels need to be obtained in an unsupervised manner; **(ii)** developing and assessing several strong clustering baselines using state-of-the-art methods for video representation learning, and **(iii)** developing a new algorithm tailored to clustering multi-modal data resulting in state-of-the-art highly semantic labels.

2 Related work

Unsupervised labelling for images. Early approaches to clustering images include agglomerative clustering [9] and partially ordered sets of hand-crafted features [10], while more recent methods combine feature learning with clustering. First, there are methods which propose to implicitly learn a clustering function by maximizing mutual information between the image and nuisance transformations [35, 37]. Second, there are methods which use explicit clustering combined with representation learning [6, 13, 14, 16, 48, 77, 80]. Lastly, there are methods which build on strong feature representations and, at a second stage, utilize these to obtain clusters [47, 72, 79].

Representation learning from videos. There is a growing literature on representation learning from videos. Many of these methods are *uni-modal*, leveraging works from the image domain [5, 8, 18, 26, 27, 56, 57, 69, 75, 81], such as predicting rotations [39] and 3D jigsaw puzzles [42]. Other works leverage temporal information explicitly and predict future features [30], the order of frames [46, 53] and clips [78], the direction of time [74] or the framerate [11, 19]. However, videos usually contain multiple modalities, such as audio, speech and optical flow. *Multi-modal* learning, originally proposed by de Sa [22], has seen a resurgence with the goal of learning strong feature representations that can be used for finetuning on downstream tasks. Most works leverage audio-visual semantic correspondence [3, 7, 59, 60] or the synchronized timing of content [43, 58] between the audio and visual streams. Some works use this information to obtain within-clip sound localisation [4, 34, 58, 63, 64, 82] as well as audio-separation [15, 25]. Other methods use a modality distillation framework to learn video encoders from other modalities [59, 61]. In [61], a loss function is meta-learned by computing common self-supervised losses and distilling these and clustering is

[3] `https://www.robots.ox.ac.uk/~vgg/research/selavi`

Figure 1: **Our model** views modalities as different *augmentations* and produces a multi-modal clustering of video datasets from scratch that can closely match human annotated labels.

used as an evaluation metric for meta-learning. New methods have started to learn even stronger representations by using ASR generated text from videos as another modality [49, 52, 55, 66, 67].

Clustering videos. Perhaps the simplest way of combining representation learning and clustering in videos is to apply *post-hoc* a clustering algorithm after pretraining a representation. In ClusterFit [79], the authors show that running a simple k-means algorithm on the features from a pretrained network on the pretraining dataset yields small but consistent gains for representation learning when these clusters are used as labels and the networks are retrained. While in [79], the authors found the optimal number of clusters to consistently be at least one order of magnitude higher than the number of ground-truth labels, we investigate applying this method on various pretrained methods as baselines for our task of labelling an unlabelled video dataset. Specifically, we apply k-means on state-of-the-art single modality models such as DPC [30] and MIL-NCE [52], as well the multi-modal model XDC [2], which itself uses k-means on the audio and visual streams to learn representations. However, they do this as a pretext task for representation learning and obtain separate clusters for audio and video. In contrast, our goal is multi-modally labelling an unlabelled dataset, and we find that our method works significantly better at this task.

3 Method

Given a dataset $D = \{\boldsymbol{x}_i\}_{i\in\{1,\dots,N\}}$ of multi-modal data $\boldsymbol{x}_i$, our goal is to learn a labelling function $y(\boldsymbol{x}) \in \{1,\dots,K\}$ without access to any ground-truth label annotations. There are two requirements that the labelling function must satisfy. First, the labels should capture, as well as possible, the *semantic content* of the data, in the sense of reproducing the labels that a human annotator would intuitively associate to the videos. As part of this, we wish to account for the fact that semantic classes are not all equally probable, and tend instead to follow a Zipf distribution [1, 41]. We then evaluate the quality of the discovered labels by matching them to the ones provided by human annotators, using datasets where ground-truth labels are known. The second requirement is that the labelling method should not overly rely on a single modality. Instead, we wish to treat each modality *as equally informative* for clustering. In this way, we can learn a more robust clustering function, which can work from either modality. Furthermore, correlating of modalities has been shown to be a proxy to learn better abstractions [4, 43, 58, 60].

While our method can work with any number of data modalities (vision, audio, depth, textual transcripts, ...), we illustrate it under the assumption of video data $\boldsymbol{x} = (a, v)$, comprising an audio stream a and a visual stream v. The following two sections describe our method in detail and show how it meets our requirements.

3.1 Non-degenerate clustering via optimal transport

In this section, we briefly summarize the formulation of [6] to interpret clustering as an optimal transport problem. SeLa [6] is a method that learns representations via clustering images. The labelling function can be expressed as the composition $y(\Psi(\boldsymbol{x}))$, where $\boldsymbol{z} = \Psi(\boldsymbol{x})$ is a data representation (i.e. a feature extractor implemented by a deep neural network), and $y(\boldsymbol{z}) \in \{1,\dots,K\}$ operates on top of the features rather than the raw data.

Any traditional clustering algorithm, such as k-means or Gaussian mixture models, defines an energy function $E(y)$ that, minimized, gives the best data clustering function y. When the representation is accounted for, the energy $E(y, \Psi)$ is a function of both y and Ψ, and we may be naïvely tempted to

optimize over both. However, this is well known to yield unbalanced solutions, which necessitates ad-hoc techniques such as non-uniform sampling or re-initialization of unused clusters [13, 14]. Theoretically, in fact, for most choices of E, the energy is trivially minimized by the representation Ψ that maps all data to a constant.

Asano et al. [6] address this issue by constraining the marginal probability distributions of the clusters to be uniform, and show that this reduces to an optimal transport problem. The algorithm then reduces to alternating the fast Sinkhorn-Knopp algorithm [21] for clustering, and standard neural network training for representation learning. To do this, one introduces the cross-entropy loss $E(q, p)$, between the labels given as one-hot vectors in q (i.e. $q(y(\boldsymbol{x})) = 1 \ \forall \boldsymbol{x}$) and the softmax outputs p of a network Ψ:

$$E(p, q) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{y=1}^{K} q(y|\boldsymbol{x}_i) \log p(y|\boldsymbol{x}_i), \quad p(y|\boldsymbol{x}_i) = \operatorname{softmax} \Psi(\boldsymbol{x}_i), \tag{1}$$

where K is the number of clusters. This energy is optimized under the constraint that the marginal cluster probability $\sum_{i=1}^{N} \frac{1}{N} p(y|\boldsymbol{x}_i) = \frac{1}{K}$ is constant (meaning all clusters are a-priori equally likely). Note that minimizing E with respect to p is the same as training the deep network Ψ using the standard cross-entropy loss.

Next, we show that minimizing $E(p, q)$ w.r.t. the label assignments q results in an optimal transport problem. Let $P_{yi} = p(y|\boldsymbol{x}_i)\frac{1}{N}$ be the $K \times N$ matrix of joint probabilities estimated by the model and $Q_{yi} = q(y|\boldsymbol{x}_i)\frac{1}{N}$ be $K \times N$ matrix of assigned joint probabilities. Matrix Q is relaxed to be an element of a transportation polytope

$$U(r, c) \coloneqq \{Q \in \mathbb{R}_{+}^{K \times N} \mid Q\mathbb{1} = r, \ Q^{\top}\mathbb{1} = c\}, \quad r = \mathbb{1}/K, \quad c = \mathbb{1}/N. \tag{2}$$

where $\mathbb{1}$ are vectors of ones, and r and c the marginal projections of matrix Q onto its clusters and data indices, respectively. Finally, optimizing $E(P, Q)$ w.r.t. to $Q \in U(r, c)$ is a linear optimal transport problem, for which [21] provides a fast, matrix-vector multiplication based solution.

3.2 Clustering with arbitrary prior distributions

A shortcoming of the algorithm just described is the assumption that all clusters are equally probable. This avoids converging to degenerate cases but is too constraining in practice since real datasets follow highly skewed distributions [1, 41], and even in datasets that are collected to be uniform, they are not completely so [17, 41, 68]. Furthermore, knowledge of the data distribution, for example long-tailedness, can be used as additional information (e.g. as in [61] for meta-learning) that can improve the clustering by allocating the right number of data points to each cluster. Next, we describe a mechanism to change this distribution arbitrarily.

In the algorithm above, changing the label prior amounts to choosing a different cluster marginal r in the polytope $U(r, c)$. The difficulty is that r is only known up to an arbitrary permutation of the clusters, as we do not know a-priori which clusters are more frequent and which ones less so. To understand how this issue can be addressed, we need to explicitly write out the energy optimised by the Sinkhorn-Knopp (SK) algorithm [21] to solve the optimal transport problem. This energy is:

$$\min_{Q \in U(r,c)} \langle Q, -\log P \rangle + \frac{1}{\lambda} \operatorname{KL}(Q \| rc^{\top}), \tag{3}$$

where λ is a fixed parameter. Let $r' = Rr$ where R is a permutation matrix matching clusters to marginals. We then seek to optimize the same quantity w.r.t. R, obtaining the optimal permutation as $R^* = \operatorname{argmin}_R E(R)$ where

$$E(R) = \langle Q, -\log P \rangle + \frac{1}{\lambda} \operatorname{KL}(Q \| Rrc^{\top}) = \text{const} + \sum_{y} -q(y) \ [R \log r]_y. \tag{4}$$

While there is a combinatorial number of permutation matrices, we show that minimizing Eq. (4) can be done by first sorting classes y in order of increasing $q(y)$, so that $y > y' \Rightarrow q(y) > q(y')$, and

then finding the permutation that R that also sorts $[R \log r]_y$ in increasing order.[4] We conclude that R cannot be optimal unless it sorts all pairs. After this step, the SK algorithm can be applied using the optimal permutation R^*, without any significant cost (as solving for R is equivalent to sorting $\mathcal{O}(K \log K)$ with $K \ll N$). The advantage is that it allows to choose any marginal distribution, even highly unbalanced ones which are likely to be a better match for real world image and video classes than a uniform distribution.

3.3 Multi-modal single labelling

Next, we tackle our second requirement of extracting as much information as possible from multi-modal data. In principle, all we require to use the clustering formulation Eq. (1) with multi-modal data $\boldsymbol{x} = (a, v)$ is to design a corresponding multi-modal representation $\Psi(\boldsymbol{x}) = \Psi(a, v)$. However, we argue for *multi-modal single labelling* instead. By this, we mean that we wish to cluster data one modality at a time, but in a way that is modality agnostic. Formally, we introduce *modality splicing transformations* [60] $t_a(\boldsymbol{x}) = a$ and $t_v(\boldsymbol{x}) = v$ and use these as *data augmentations*. Recall that augmentations are random transformations t such as rotating an image or distorting an audio track that one believes should leave the label/cluster invariant. We thus require our activations used for clustering to be an average over augmentations by replacing matrix $\log P$ with

$$[\log P]_{yi} = \mathbb{E}_t[\log \operatorname{softmax}_y \Psi(t\boldsymbol{x}_i)]. \tag{6}$$

If we consider splicing as part of the augmentations, we can learn clusters that are invariant to standard augmentations *as well as* the choice of modality. In practice, to account for modality splicing, we define and learn a pair $\Psi = (\Psi_a, \Psi_v)$ of representations, one per modality, resulting in the same clusters $(\Psi_a(t_a(\boldsymbol{x})) \approx \Psi_v(t_v(\boldsymbol{x})))$. This is illustrated in Figure 1.

Initialization and alignment. Since networks Ψ_a and Ψ_v are randomly initialized, at the beginning of training their output layers are *de-synchronized*. This means that there is no reason to believe that $\Psi_a(t_a(\boldsymbol{x}))) \approx \Psi_v(t_v(\boldsymbol{x}))$ simply because the *order* of the labels in the two networks is arbitrary. Nevertheless, in many self-supervised learning formulations, one exploits the fact that even randomly initialized networks capture a useful data prior [71], which is useful to bootstrap learning.

In order to enjoy a similar benefit in our formulation, we propose to *synchronise* the two output layers of Ψ_a and Ψ_b before training the model. Formally, we wish to find the permutation matrix R that, applied to the last layer of one of the two encoders maximizes the agreement with the other (still leaving all the weights to their initial random values). For this, let W_a and W_v be the last layer weight matrices of the two networks,[5] such as $\Psi_a(a) = W_a \bar{\Psi}_a(a)$ and $\Psi_v(v) = W_v \bar{\Psi}_v(v)$. We find the optimal permutation R by solving the optimisation problem:

$$\min_R \sum_{i=1}^{N} \left| \operatorname{softmax}(R W_a \bar{\Psi}_a(t_a(\boldsymbol{x}_i))) - \operatorname{softmax}(W_v \bar{\Psi}_v(t_v(\boldsymbol{x}_i))) \right|, \tag{7}$$

In order to compare softmax distributions, we choose $|\cdot|$ as the 1-norm, similar to [33]. We optimize Eq. (7) with a greedy algorithm: starting with a feasible solution and switching random pairs when they reduce the cost function [20], as these are quick to compute and we do not require the global minimum. Further details are given in Appendix A.3. With this permutation, the weight matrix of the last layer of one network can be resorted to match the other.

Decorrelated clustering heads. Conceptually, there is no single 'correct' way of clustering a dataset: for example, we may cluster videos of animals by their species, or whether they are taken indoor or outdoor. In order to alleviate this potential issue, inspired by [6, 37], we simply learn

[4] To see why this is optimal, and ignoring ties for simplicity, let R be any permutation and construct a permutation $\bar{R}$ by applying R and then by further swapping two labels $y > y'$. We can relate the energy of R and $\bar{R}$ as:

$$\begin{aligned} E(R) &= E(\bar{R}) + q(y)[\bar{R} \log r]_y + q(y')[\bar{R} \log r]_{y'} - q(y)[\bar{R} \log r]_{y'} - q(y')[\bar{R} \log r]_y \\ &= E(\bar{R}) + (q(y) - q(y'))\,([\bar{R} \log r]_y - [\bar{R} \log r]_{y'}). \end{aligned} \tag{5}$$

Since the first factor is positive by assumption, this equation shows that the modified permutation $\bar{R}$ has a lower energy than R if, and only if, $[\bar{R} \log r]_y > [\bar{R} \log r]_{y'}$, which means that $\bar{R}$ sorts the pair in increasing order.

[5] We assume that the linear layer biases are incorporated in the weight matrices.

multiple labelling functions y, using multiple classification heads for the network. We improve this scheme as follows. In each round of clustering, we generate two random augmentations of the data. Then, the applications of SK to half of the heads (at random) see the first version, and the other half the second version, thus increasing the variance of the resulting clusters. This increases the cost of the algorithm by only a small amount — as more time is used for training instead of clustering.

4 Experiments

The experiments are divided into three parts. First, in Section 4.1, we analyze the need for using both modalities when clustering and investigate the effect of our individual technical contributions via ablations and comparison to other approaches. Second, in Section 4.2, we demonstrate how our method achieves its stated goal of labelling a video dataset without human supervision. Third, in Section 4.3, we show that a side effect of our method is to learn effective audio-visual representations that can be used for downstream tasks *e.g.*video action retrieval, establishing a new state of the art.

Datasets. While the goal and target application of this work is to group unlabelled video datasets, for analysis purposes only, we use datasets that contain human annotated labels. The datasets range from small- to large-scale: The first is the recently released **VGG-Sound** [17], which contains around 200k videos obtained in the wild from YouTube with low labelling noise and covering 309 categories of general classes. The second dataset is **Kinetics-400** [41], which contains around 230k videos covering 400 human action categories. Third, we test our results on **Kinetics-Sound** proposed in [3], formed by filtering the Kinetics dataset to 34 classes that are potentially manifested visually and audibly, leading to 22k videos. Lastly, we use the small-scale **AVE Dataset** [68], originally proposed for audio-visual event localization and containing only around 4k videos. Among these, only VGG-Sound and Kinetics-400 are large enough for learning strong representations from scratch. We therefore train on these datasets and unsupervisedly finetune the VGG-Sound model on Kinetics-Sound and AVE.

Training details. Our visual encoder is a R(2+1)D-18 [70] network and our audio encoder is a ResNet [32] with 9 layers. For optimization, we use SGD for 200 epochs with weight decay of 10^{-5} and momentum of 0.9, further implementation details are provided in Appendix A.2.

Table 1: **Architectures and pretraining datasets.** We use state-of-the-art representation learning methods and combine pretrained representations with k-means as baselines in the Tables 5a to 5d.

Method	Input shape	Architecture	Pretrain dataset
Supervised	$32 \times 3 \times 112 \times 112$	R(2+1)D-18	Kinetics-400
DPC [30]	$40 \times 3 \times 224 \times 224$	R3D-34	Kinetics-400
MIL-NCE [52]	$32 \times 3 \times 224 \times 224$	S3D	HowTo100M
XDC [2]	$32 \times 3 \times 224 \times 224$	R(2+1)D-18	Kinetics-400

Baselines. To compare our method on this novel task of clustering these datasets, we obtained various pretrained video representations (DPC [30], MIL-NCE [52]and XDC [2]), both supervised[6] and self-supervised (see Table 1 for details). For comparison, following [79], we run k-means on the global-average-pooled features, setting k to the same number of clusters as our method to ensure a fair comparison. For the k-means algorithm, we use the GPU accelerated version from the Faiss library [40].

Evaluation. We adopt standard metrics from the self-supervised and unsupervised learning literature: the *normalized mutual information* (NMI), the *adjusted rand index* (ARI) and *accuracy* (Acc) after matching of the self-supervised labels to the ground truth ones (for this we use the Kuhn–Munkres/Hungarian algorithm [45]). We also report the *mean entropy* and the *mean maximal purity* per cluster, defined in Appendix A.4, to analyze the qualities of the clusters. For comparability and interpretability, we evaluate the results using the ground truth number of clusters – which usually is unknown – but we find our results to be stable w.r.t. other number of clusters (see Appendix).

[6]The R(2+1)D-18 model from PyTorch trained on Kinetics-400 [41] from `https://github.com/pytorch/vision/blob/master/torchvision/models/video/resnet.py`.

Table 2: **The value of multi-modal understanding** is observed for obtaining a strong set of labels for VGG-Sound. Our method combines both modalities effectively to yield accurracies beyond a single modality and other methods.

Method	🔊	📹	NMI	ARI	Acc.	⟨H⟩	$\langle \mathbf{p}_{\max} \rangle$
Random	✗	✓	10.2	4.0	2.2	4.9	3.5
Supervised	✗	✓	46.5	15.6	24.3	2.9	30.8
DPC	✗	✓	15.4	0.7	3.2	4.7	4.9
MIL-NCE	✗	✓	48.5	12.5	22.0	2.6	32.9
XDC	✗	✓	16.7	1.0	3.9	4.5	6.4
	✓	✗	14.0	0.8	2.9	4.6	4.4
	✓	✓	18.1	1.2	4.5	4.41	7.4
SeLaVi	✗	✓	52.8	19.7	30.1	2.6	35.6
	✓	✗	47.5	15.2	26.5	2.8	32.9
	✓	✓	**55.9**	**21.6**	**31.0**	**2.5**	**36.3**

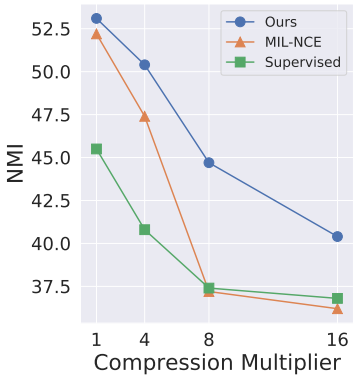


Figure 2: **Effective use of multi-modality** is found for our method when the visual input is compressed and decompressed.

Table 3: **Ablation** of multi-modality, Modality Alignment and Gaussian marginals. Decorrelated Heads. Models are evaluated at 75 epochs on the VGG-Sound dataset.

Method	🖼	(🔊 📹)	MA?	G.?	DH?	Acc	ARI	NMI
(a) SeLa	✓	✗	–	–	–	6.4	2.3	20.6
(b) Concat	✗	✓	–	✗	✗	7.6	3.2	24.7
(c) SeLaVi	✗	✓	✗	✗	✗	24.6	15.6	48.8
(d) SeLaVi	✗	✓	✗	✓	✓	26.6	18.5	50.9
(e) SeLaVi	✗	✓	✓	✗	✓	26.2	17.3	51.5
(f) SeLaVi	✗	✓	✓	✓	✗	23.9	14.7	49.9
(g) **SeLaVi**	✗	✓	✓	✓	✓	26.6	17.7	51.1

Table 4: **Retrieval** via various number of nearest neighbors.

	HMDB			UCF		
Recall @	1	5	20	1	5	20
3D-Puzzle [42]	–	–	–	19.7	28.5	40.0
OPN [46]	–	–	–	19.9	28.7	40.6
ST Order [12]	–	–	–	25.7	36.2	49.2
ClipOrder [78]	7.6	22.9	48.8	14.1	30.3	51.1
SpeedNet [11]	–	–	–	13.0	28.1	49.5
VCP [51]	7.6	24.4	53.6	18.6	33.6	53.5
VSP [19]	10.3	26.6	54.6	24.6	41.9	76.9
SeLaVi	**24.8**	**47.6**	**75.5**	**52.0**	**68.6**	**84.5**

4.1 Technical Analysis

Multi-modality. In order to shed light on the nature of labelling a multi-modal dataset, we provide a detailed study of the use and combination of modalities in Table 2. While visual-only methods such as the Kinetics-400 supervisedly pretrained model, MIL-NCE or DPC cannot work when only audio-stream (🔊) is provided, we show the results for XDC and our method when a single or both modalities are provided. In particular, we find that even when only the visual-stream (📹) is present at test-time, our method (57% NMI) already outperforms methods solely developed for representation learning, even surpassing the 100M videos with transcripts trained MIL-NCE (49% NMI). When only the audio-stream is used, our method's performance drops only slightly, congruent with the balanced informativeness of both modalities in our method. Finally, when both modalities are used, we find that our method profits from both, such that the combined performance is significantly higher than the maximal performance of each single modality alone.

Degraded modality. In Fig. 2, we analyze how well our method fares when the quality of one modality is reduced. For this, we compress the video-stream by down and upsampling of the video resolution by factors 1, 4, 8 and 16 (details are provided in Appendix A.5). Even though our method has not been trained with compressed videos, we find that its performance degrades more gracefully than the baselines indicating it has learned to rely on both modalities.

Ablation. Next, we ablate the key parameters of our method and show how they each contribute to the overall clustering quality in Table 3. First, we show a baseline model in Table 3(a), when naïvely applying the publically available source-code for SeLa [6] on video frames (🖼), this yields a NMI of 20%. Compared to this frame-only method, in row (b), we find the results for concatenating the features of both modalities (followed by a single clustering head) to only lead to a small improvement upon the frame-only method with a NMI of 25%. Our method is shown in row (c), where we

Table 5: **Unsupervised labelling of datasets.** We compare labels from our method to labels that are obtained with k-means on the representations from a supervised and various unsupervised methods on four datasets.

(a) **VGG-Sound.**

Method	**NMI**	**ARI**	**Acc.**	$\langle \mathbf{H} \rangle$	$\langle \mathbf{p}_{\max} \rangle$
Random	10.2	4.0	2.2	4.9	3.5
Supervised	46.5	15.6	24.3	2.9	30.8
DPC	15.4	0.7	3.2	4.7	4.9
XDC	18.1	1.2	4.5	4.41	7.4
MIL-NCE	48.5	12.5	22.0	2.6	32.9
SeLaVi	**55.9**	**21.6**	**31.0**	**2.5**	**36.3**

(b) **AVE.**

Method	**NMI**	**ARI**	**Acc.**	$\langle \mathbf{H} \rangle$	$\langle \mathbf{p}_{\max} \rangle$
Random	9.2	1.3	9.3	2.9	12.6
Supervised	58.4	34.8	50.5	1.1	60.6
DPC	18.4	5.0	15.1	2.7	17.5
XDC	17.1	6.0	16.4	2.6	19.1
MIL-NCE	56.3	30.3	42.6	**1.2**	57.1
SeLaVi	**66.2**	**47.4**	**57.9**	**1.1**	**59.3**

(c) **Kinetics.**

Method	**NMI**	**ARI**	**Acc.**	$\langle \mathbf{H} \rangle$	$\langle \mathbf{p}_{\max} \rangle$
Random	11.1	0.2	1.8	5.1	3.3
Supervised	70.5	43.4	54.9	1.6	62.2
DPC	16.1	0.6	2.7	4.9	3.9
XDC	17.2	0.8	3.4	4.7	6.2
MIL-NCE	**48.9**	**12.5**	**23.5**	**2.7**	**33.7**
SeLaVi	27.1	3.4	7.8	4.8	9.4

(d) **Kinetics-Sound.**

Method	**NMI**	**ARI**	**Acc.**	$\langle \mathbf{H} \rangle$	$\langle \mathbf{p}_{\max} \rangle$
Random	2.8	0.5	5.9	3.3	8.3
Supervised	81.7	66.3	75.0	0.5	85.4
DPC	8.8	2.2	9.6	3.1	13.6
XDC	7.5	1.9	9.4	3.1	13.6
MIL-NCE	**47.5**	24.0	37.8	**1.5**	**51.0**
SeLaVi	**47.5**	**28.7**	**41.2**	1.8	45.5

find a substantial improvement with a NMI of 52%, i.e. a relative gain of more than 100%. While part of the gain comes from multi-modality, especially compared to row (b), the largest gain comes from the ability of our method in exploiting the natural correspondence provided in the multi-modal data. Finally, by ablating the technical improvements in rows (d)-(f) we find the strongest gain to be coming decorrelated heads, followed by the audio-visual modality alignment (MA) procedure, and that each improvement indeed benefits the model. To analyze the gains obtained by using multiple heads, we have also computed the average NMI between all pairs of heads as $(77.8 \pm 4\%)$. This means that the different heads do learn fairly different clusterings (as NMI takes permutations into account) whilst being at a similar distance to the 'ground-truth' $(53.1 \pm 0.1\%)$.

4.2 Unsupervised labelling audio-visual data

Table 5 shows the quality of the labels obtained automatically by our algorithm. We find that for the datasets VGG-Sound, Kinetics-Sound, and AVE, our method achieves state-of-the-art clustering performance with high accuracies of $55.9\%, 41.2\%, 57.9\%$, even surpassing the one of the strongest video feature encoder at present, the manually-supervised R(2+1)D-18 network. This result echoes the findings in the image domain [72] where plain k-means on representations is found to be less effective compared to learning clusters. For Kinetics-400, we find that the clusters obtained from our method are not well aligned to the human annotated labels. This difference can explained by the fact that Kinetics is strongly focused on visual (human) actions and thus the audio is given almost no weighting in the annotation. We encourage exploring our interactive material, where our method finds clusters grouped by similar background music, wind or screaming crowds. We stress that such a grouping is *ipso facto* not wrong, only not aligned to this set of ground truth labels.

4.3 Labelling helps representation learning

Finally, we show how the visual feature representations unsupervisedly obtained from our method perform on downstream tasks. While not the goal of this paper, we test our representation on a standardized video action retrieval task in Table 4 and also provide results on video action classification in Table A.3, and refer to the Appendix for implementation details. We find that in obtaining strong labels, our method simultaneously learns robust, visual representations that can be used for other tasks without any finetuning and significantly improve the state of the art by over 100% for Recall @1 on UCF-101 and HMDB-51.

5 Conclusion

In this work, we have established strong baselines for the problem of unsupervised labelling of several popular video datasets; introduced a simultaneous clustering and representation learning approach for multi-modal data that outperforms all other methods on these benchmarks; and analysed the importance of multi-modality for this task in detail. We have further found that strong representations are not a sufficient criterion for obtaining good clustering results, yet, the strongest feature representations remain those obtained by supervised, *i.e*. well-clustered training. We thus expect the field of multi-modal clustering to be rapidly adopted by the research community who can build upon the presented method and baselines.

Broader Impact

We propose a method for clustering videos automatically. As such, we see two main areas of potential broader impact on the community and society as a whole.

Few-label harmful content detection. Our method clusters a video dataset into multiple sets of similar videos, as evidenced by the audio- and visual-stream and produces consistent, homogenous groupings. In practice, unsupervised clustering is especially useful for reducing the amount of data that human annotators have to label, since for highly consistent clusters only a single label needs to be manually obtained which can be propagated to the rest of the videos in the cluster. Using such an approach for the purpose of detecting harmful online content is especially promising. In addition, label-propagation might further lead to a beneficial reduction of type I errors (saying a video is safe when it is not). Furthermore, the multi-modality of our method allows it to potentially detect harmful content that is only manifested in one modality such as static background videos of harmful audio. Multi-modal harmful content detection has also been a subject of a recent data challenge that emphasizes insufficiency of using a single modality[7]. Lastly, the generality of our method allows it to also scale beyond these two modalities and in the future also include textual transcripts. Given the importance of this topic, it is also important to acknowledge, while less of a direct consequence, potential biases that can be carried by the dataset. Indeed models trained using our method will inherit the biases present in the dataset, which could be known but also unknown, potentially leading to propagation of biases without a clear way to analyze them, such as via labels. However, given the numerous pitfalls and failures when deploying computer vision systems to the real world, we believe that the positive impact of foundational research on public datasets, such as is presented in this paper, far outweighs these risks lying further downstream.

Overestimating clustering quality. The main benefit of our approach is to reduce the cost of grouping large collections of video data in a 'meaningful' way. It is difficult to think of an application where such a capability would lead directly to misuse. In part, this is due to the fact that better clustering results can generally be obtained by using some manual labels, so even where clustering videos could be misused, this probably would not be the method of choice. Perhaps the most direct risk is that a user of the algorithm might overestimate its capabilities. Clustering images is sometimes done in critical applications (e.g. medical science [36, 38, 50]). Our method clusters data based on basic statistical properties and the inductive prior of convolutional neural networks, without being able to tap into the deep understanding that a human expert would have of such domain expertise. Hence, the clusters determined by our method may not necessarily match the clusters an expert would make in a particular domain. Further, as the method is unsupervised, it may learn to exploit biases present in the data that might not be desired by the users. While we believe it has potential to be broadly applied after being finetuned to a specific domain, at present, our method is a better fit for applications such as indexing personal video collections where clustering 'errors' can be tolerated.

Acknowledgments and Disclosure of Funding

We are grateful for support from the Rhodes Trust (M.P.), Qualcomm Innovation Fellowship (Y.A.) and EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines & Systems

[7]Hateful memes challenge: `https://www.drivendata.org/competitions/64/hateful-memes/`

[EP/L015897/1] (M.P. and Y.A.). M.P. funding was received under his Oxford affiliation. C.R. is supported by ERC IDIU-638009. We thank Weidi Xie and Xu Ji from VGG for fruitful discussions.

Erratum In our initial version there was a bug in our code and we have since updated our repository and updated results in Tables 2,3 and 5 in this paper.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *arXiv preprint arXiv:1911.12667*, 2019.

[3] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proc. ICCV*, 2017.

[4] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.

[5] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *ICLR*, 2020.

[6] Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2020.

[7] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. In *NeurIPS*, 2016.

[8] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views, 2019.

[9] Miguel A Bautista, Artsiom Sanakoyeu, Ekaterina Tikhoncheva, and Bjorn Ommer. Cliquecnn: Deep unsupervised exemplar learning. In *NeurIPS*, pages 3846–3854, 2016.

[10] Miguel A Bautista, Artsiom Sanakoyeu, and Bjorn Ommer. Deep unsupervised similarity learning using partially ordered sets. In *CVPR*, pages 7130–7139, 2017.

[11] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos, 2020.

[12] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 770–786, 2018.

[13] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

[14] Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019.

[15] Anna Llagostera Casanovas, Gianluca Monaci, Pierre Vandergheynst, and Rémi Gribonval. Blind audiovisual source separation based on sparse redundant representations. *IEEE Transactions on Multimedia*, 12(5):358–371, 2010.

[16] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *ICCV*, pages 5879–5887, 2017.

[17] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. *ICASSP)*, May 2020.

[18] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. *ArXiv*, abs/2002.05709, 2020.

[19] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised spatio-temporal representation learning using variable playback speed prediction. *arXiv preprint arXiv:2003.02692*, 2020.

[20] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

[21] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NeurIPS*, pages 2292–2300, 2013.

[22] Virginia R. de Sa. Learning classification with unlabeled data. In J. D. Cowan, G. Tesauro, and J. Alspector, editors, *NeurIPS*, pages 112–119. Morgan-Kaufmann, 1994.

[23] Thomas Feo and Mauricio Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 03 1995. doi: 10.1007/BF01096763.

[24] Paola Festa and Mauricio GC Resende. An annotated bibliography of grasp. *Operations Research Letters*, 8:67–71, 2004.

[25] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, pages 35–53, 2018.

[26] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *ICLR*, 2018.

[27] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord. Learning representations by predicting bags of visual words, 2020.

[28] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[29] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, pages 6047–6056, 2018.

[30] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV*, 2019.

[31] John A Hartigan. Direct clustering of a data matrix. *Journal of the american statistical association*, 67 (337):123–129, 1972.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[33] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[34] Di Hu, Feiping Nie, and Xuelong Li. Deep multimodal clustering for unsupervised audiovisual learning. In *CVPR*, pages 9248–9257, 2019.

[35] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *ICML*, pages 1558–1567, 2017.

[36] Dimitris K Iakovidis, Spiros V Georgakopoulos, Michael Vasilakakis, Anastasios Koulaouzidis, and Vassilis P Plagianakos. Detecting and locating gastrointestinal anomalies using deep learning and iterative cluster unification. *IEEE transactions on medical imaging*, 37(10):2196–2210, 2018.

[37] Xu Ji, João F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation, 2018.

[38] Yizhang Jiang, Kaifa Zhao, Kaijian Xia, Jing Xue, Leyuan Zhou, Yang Ding, and Pengjiang Qian. A novel distributed multitask fuzzy clustering algorithm for automatic mr brain image segmentation. *Journal of medical systems*, 43(5):118, 2019.

[39] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

[40] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

[41] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[42] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019.

[43] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[44] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011.

[45] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.

[46] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017.

[47] Juho Lee, Yoonho Lee, and Yee Whye Teh. Deep amortized clustering. *arXiv preprint arXiv:1909.13433*, 2019.

[48] Junnan Li, Pan Zhou, Caiming Xiong, Richard Socher, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.

[49] Tianhao Li and Limin Wang. Learning spatiotemporal features via video and text pair discrimination, 2020.

[50] Zhuoling Li, Minghui Dong, Shiping Wen, Xiang Hu, Pan Zhou, and Zhigang Zeng. Clu-cnns: Object detection for medical images. *Neurocomputing*, 350:53–59, 2019.

[51] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *AAAI*, 2020.

[52] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos, 2019.

[53] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016.

[54] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement, 2020.

[55] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *CVPR*, 2020.

[56] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[57] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017.

[58] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.
[59] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.
[60] Mandela Patrick, Yuki M. Asano, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations, 2020.
[61] AJ Piergiovanni, Anelia Angelova, and Michael S. Ryoo. Evolving losses for unsupervised video representation learning, 2020.
[62] Mauricio GC Resende and Celso C Ribeiro. Greedy randomized adaptive search procedures: Advances and extensions. In *Handbook of metaheuristics*, pages 169–220. Springer, 2019.
[63] Andrew Rouditchenko, Hang Zhao, Chuang Gan, Josh McDermott, and Antonio Torralba. Self-supervised audio-visual co-segmentation. In *ICASSP*, 2019.
[64] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. *CVPR*, Jun 2018.
[65] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.
[66] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019.
[67] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, pages 7464–7473, 2019.
[68] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018.
[69] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.
[70] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.
[71] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *arXiv preprint arXiv:1711.10925*, 2017.
[72] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European Conference on Computer Vision (ECCV)*, 2020.
[73] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019.
[74] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018.
[75] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, June 2018.
[76] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.
[77] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.
[78] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019.
[79] Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. ClusterFit: Improving Generalization of Visual Representations. In *CVPR*, 2020.
[80] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *CVPR*, pages 5147–5156, 2016.
[81] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.
[82] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, pages 1735–1744, 2019.

A Appendix

A.1 Pretrained model details

Here we provide additional information about the pretrained models we have used in this work.

Table A.1: **Details for audio encoder.** Architectural and pretraining details for XDC's audio encoder used for benchmarking.

Method	Input shape	Architecture	Pretrain dataset
XDC	$40 \times 1 \times 100$	Resnet-18	Kinetics-400

A.2 Implementation details

We train our method using the Sinkhorn-Knopp parameter $\lambda = 20$, an inverse quadratic clustering schedule with 100 clustering operations and 10 heads which we adopt from [6]. For evaluation, we report results for head 0 to compare against the ground-truth, as we found no significant difference in performance between heads. For the Gaussian distribution, we take the marginals to be from $\mathcal{N}(1, 0.1) * N/K$. For the clustering-heads, we use two-layer MLP-heads as in [8, 18]. The video inputs are 30 frame long clips sampled consecutively from 30fps videos and are resized such that the shorter side is 128 and during training a random crop of size 112 is extracted, no color-jittering is applied. Random horizontal flipping is applied to the video frames with probability 0.5, and then the channels of the video frames are Z-normalized using mean and standard deviation statistics computed across the dataset. The audio is processed as a $1 \times 257 \times 199$ image, by taking the log-mel bank features with 257 filters and 199 time-frames and for training, random volume jittering between 90% and 110% is applied to raw waveform, similar to [54]. For evaluation, a center-crop is taken instead for the video inputs and audio volume is not jittered. We use a mini-batch size of 16 on each of our 64 GPUs giving an effective batch size of 1024 for distributed training for 200 epochs. The initial learning rate is set to 0.01 which we linearly scale with the number of GPUs, after following a gradual warm-up schedule for the first 10 epochs [28]. For training on Kinetics-Sound and AVE, we initialize our model with a VGG-Sound pretrained backbone due to the small training set sizes ($N = 22$k and $N = 3328$). The clustering heads are re-initialized randomly. This ensures a more fair comparison as XDC, DPC and the supervised model are pretrained on Kinetics-400 with $N = 230$k and MIL-NCE on HowTo100M with $N = 100$M videos. We train on VGG-Sound for 200 epochs, which takes around 2 days.

A.3 Pair-based optimization for AV-Alignment

For aligning the visual and audio encoder, we use a greedy switching algorithm that starts from a feasible initial solution [23, 24, 62]. In particular, we consider 50000 potential pair switches with 5 randomized restarts and take the final permutation that yields the lowest cost.

A.4 Evaluation metrics details

The **normalized mutual information** (NMI) is calculated by the formula

$$\text{NMI} = \frac{\text{MI}(U, V)}{0.5H(U) + 0.5H(V)}, \tag{8}$$

where the Mutual information MI is given by $\text{MI}(U, V) = \sum_{i=1}^{|U|} \sum_{j=1}^{|V|} P(i, j) \log \left(\frac{P(i,j)}{P(i)P'(j)} \right)$, and H is the standard entropy, with $H(U) = -\sum_{i=1}^{|U|} P(i) \log(P(i))$. The NMI ranges from 0 (no mutual information) to 100%, which implies perfect correlation.

The rand index (RI) is given by $\text{RI} = \frac{a+b}{C}$, where a, b are the number of pairs of elements that are in the same/different set in the ground truth labelling and in the same/different set in the predicted clustering and C is the total number of such pairs. The a**djusted Rand index** (ARI) corrects for

random assignments and is given by

$$\text{ARI} = \frac{\text{RI} - E[\text{RI}]}{\max(\text{RI}) - E[\text{RI}]}, \tag{9}$$

where the expected RI of a random label assignment is subtracted in both nominator and denominator. Due to the subtraction, the ARI varies from -1 to 1 with a value close to 0 implying random correlation and a value of 1 implying identical agreement.

The **mean entropy per cluster** is given by

$$\langle H \rangle = \frac{1}{K} \sum_{k \in K} H(p(y|\hat{y}_k = k)), \tag{10}$$

where $\hat{y}$ are unsupervisedly obtained clusters and $p(y|\hat{y}_k = k)$ is the distribution of ground-truth clusters for cluster k. Hence, the optimal number of this metric is 0 and a chance assignment yields $\langle H \rangle = -\log 1/K$.

Further, as we wish to understand the semantic purity compared to the ground truth labels of each cluster, so we additionally report the the **mean maximal purity per cluster**,

$$\langle p_{\max} \rangle = \frac{1}{K} \sum_{k \in K} \max(p(y|\hat{y}_k = k)), \tag{11}$$

which ranges from $\langle p_{\max} \rangle = 1/K$ (chance level) to perfect matching at $\langle p_{\max} \rangle = 100\%$.

A.5 Single modality degradation experiment details

We use the default input-sizes for each model, i.e. 112 for ours and the supervised model, 224 for MIL-NCE. Compression is implemented by nearest-neighbor downsampling and subsequently nearest-neighbor upsamling for speed. For this experiment only, we evaluate the performance on the smaller validation sets.

A.6 Further ablations

In Table A.2, we provide the results for varying the number of clusters K in our algorithm. We find that even when moving from the ground-truth number of classes ($K = 309$), to lower numbers ($K = 256$) or higher estimates ($K = 619, 1024$) our results remain stable with the NMI staying almost constant. While the ARI does drop for larger K, we also observe an increase in the purity of the clusters for a larger number of clusters from $\langle \mathbf{p}_{\max} \rangle = 38.0$ for $K = 309$ to $\langle \mathbf{p}_{\max} \rangle = 42.7$ for $K = 619$, which can be particularly useful when dividing the dataset into clusters and subsequently only obtaining human annotations for few examples per cluster.

Table A.2: **Varying K** in our method degrades performances only slightly, showing that our method is robust to various estimations of the ground-truth number of classes. Results on VGG-Sound.

Method	K	**NMI**	**ARI**	**Acc.**	$\langle \mathbf{H} \rangle$	$\langle \mathbf{p}_{\max} \rangle$
SeLaVi	309	56.7	22.5	32.3	2.4	38.0
SeLaVi	256	56.8	24.3	34.2	2.4	36.9
SeLaVi	619	56.9	16.8	23.0	2.2	42.7
SeLaVi	1024	55.1	16.3	9.6	2.1	42.2

A.7 Retrieval downstream task implementation details

We follow [78] in our evaluation protocol and use split 1 of UCF101 and HMDB-51. We uniformly sample 10 clips per video, and average the max-pooled features after the last residual block for each clip per video. We then utilize the averaged features from the validation set to query the videos in the training set. The cosine distance of representations between the query clip and all clips in the training set are computed and when the class of a test clip appears in the classes of k nearest training clips, it is considered to be correctly retrieved. R@k refers to the retrieval performance using k nearest neighbors.

A.8 Visual classification downstream task

Table A.3: **Representation learning downstream evaluation.** Self-supervised and fully-supervised trained methods on UCF101 and HMDB51 benchmarks. We follow the standard protocol and report the average top-1 accuracy over the official splits and show results for finetuning the whole network. Methods with † indicate the additional use of video titles and ASR generated text as supervision. Methods with * use ASR generated text.

Method	**Architecture**	**Pretrain Dataset**	**Top-1 Acc%**	
			UCF	HMDB
Full supervision [2]	R(2+1)D-18	ImageNet	82.8	46.7
Full supervision [2]	R(2+1)D-18	Kinetics-400	**93.1**	**63.6**
Weak supervision, CPD [49]†	3D-Resnet50	Kinetics-400	88.7	57.7
MotionPred [73]	C3D	Kinetics-400	61.2	33.4
RotNet3D [39]	3D-ResNet18	Kinetics-600	62.9	33.7
ST-Puzzle [42]	3D-ResNet18	Kinetics-400	65.8	33.7
ClipOrder [78]	R(2+1)D-18	Kinetics-400	72.4	30.9
DPC [30]	3D-ResNet34	Kinetics-400	75.7	35.7
CBT [66]	S3D	Kinetics-600	79.5	44.6
Multisensory [58]	3D-ResNet18	Kinetics-400	82.1	-
XDC [2]	R(2+1)D-18	Kinetics-400	84.2	47.1
AVTS [43]	MC3-18	Kinetics-400	85.8	56.9
AV Sync+RotNet [76]	AVSlowFast	Kinetics-400	87.0	54.6
GDT [60]	R(2+1)D-18	Kinetics-400	<u>88.7</u>	<u>57.8</u>
SeLaVi	R(2+1)D-18	Kinetics-400	83.1	47.1
SeLaVi	R(2+1)D-18	VGG-Sound	87.7	53.1

In Table A.3 we show the performance of our method on two common visual-only video feature representation benchmarks, UCF-101 [65] and HMDB-51 [44]. Note that, as is the standard in this evaluation, we use our visual encoder as initialization and fine-tune the whole network on the target down-stream task. In particular, we follow the finetuning schedule of the one of the current state-of-the-art methods [60]. We find that we achieve competitive performance when trained on VGG-Sound, even surpassing XDC, despite our method using only a spatial resolution of 112×112 and not 224×224.

9 Discussion

In this chapter, we summarize the main achievements and impact of the work presented in this thesis (Section 9.1), and then suggest areas for future work (Section 9.2).

9.1 Achievements and Impact

In this thesis, we have presented a number of contributions to improve representation learning performance from unlabelled multi-modal data, and additionally, enable greater interepretability of deep representations. At the time of submission in June 2021, the work in this thesis has been cited over 200 times according to Google Scholar, and we attempt to contextualize our contributions and impact below.

GDT (chap. 2) served as the foundation for the majority of the work in this thesis. The key findings in GDT were the importance of extremely large-scale data for pretraining of video representations, the choice of data transformations, in particular, transforming the input data to look at multiple modalities (in this case, audio, images, and text), and the use of noise-contrastive (NCE) training for multi-modal self-supervision. In subsequent chapters, we attempted to explore each of these axes even further.

9. Discussion

GDT demonstrated the benefits of very *large-scale pretraining* of multi-modal video representations by showing clear performance improvements when trained on millions of video clips from the HT100M [Miech et al., 2019] and IG65M [Ghadiyaram et al., 2019] datasets. The success of very large-scale pretraining in GDT inspired our use of the HT100M dataset in subsequent works on video-text representation learning such as SSB (chap. 4) and MMP (chap. 5), where we saw similar gains in performance on the video-text retrieval task. GDT has inspired a few works in the literature to collect and leverage very large-scale multi-modal video data to improve video representations [Akbari et al., 2021; Lee et al., 2021]. Furthermore, GDT has influenced the use of large-scale multi-modal pretraining in an industrial setting to improve content recommendations for Instagram Reels [Zweig et al., 2021].

While noise contrastive training [Hadsell et al., 2006; Gutmann and Hyvärinen, 2010] has lead to breakthroughs in self-supervised learning in the image domain [Chen et al., 2020; Misra and van der Maaten, 2020; He et al., 2020] by encoding invariance to differently cropped versions of the image, GDT was one of the first works (along with concurrent works Tian et al. [2020]; Miech et al. [2020]) to show its benefits for learning multi-modal representations. All of our subsequent works in representation learning have leveraged variants of this loss. Since being ArXived in March 2020, our work has inspired a range of works using noise contrastive training for multi-modal self-supervision: video-audio [Ma et al., 2021a,b; Pedro Morgado, 2021; Kalayeh et al., 2021], video-audio-text [Akbari et al., 2021] and even multi-modal 3D [Zhang et al., 2021] representation learning.

In addition to large-scale pretraining and noise contrastive training, another major axis we explored in this thesis was the importance of the choice of *data transformations*. In STiCA (chap. 3), we were interested in whether effective data transformations in the image domain were also important for video representation learning. In particular, taking multiple spatial crops of an image and enforcing invariance using a noise contrastive loss have proven to be very effective in image representation learning [Chen et al., 2020; Caron et al., 2020]. Unlike GDT, where we solely did

cross-modal comparisons, STiCA attempted to improve the cross-modal GDT baseline by additionally incorporating a within-modal contrastive loss from multiple spatial crops. We struggled to fit more than 2 crops in memory, particularly in a multi-modal setup, and this inspired taking crops in the feature space to reduce memory demands, while simultaneously increasing the number of crops we can use for NCE comparisons. Dubbed *Feature Crop*, we are excited for this to be incorporated in more representation learning works along with other feature-level augmentations [Kalantidis et al., 2020].

Given the success of extracting audio and visual modality as a data transformation for improving representation learning performance in GDT and STiCA, we then attempted to explore other multi-modal transformations for video representation learning. In particular, we were interested in whether having annotations in multiple languages can improve video-text representation learning. This led to MMP (chap. 5), where we showed that pretraining video-text representations with videos with annotations in multiple languages can lead to performance improvements compared to simply doing English-only video-text pretraining. To encourage more research in this area, we have publicly released a new dataset, Multi-HT100M, where we provide captions in 9 languages for most of the videos in the HT100M dataset [Miech et al., 2019]. Multi-lingual multi-modal representation learning is an exciting direction of research as it can open up opportunities to not only train better video-text representations, but also make searching video content on the Internet more accessible to a wider population.

While working on MMP, we found the use of the transformer architecture [Vaswani et al., 2017] to be extremely effective for aggregating information from sequences of both visual and textual features. This transformer pooling layer significantly improved upon the standard max and mean pooling approaches that were commonplace in the video-text literature [Liu et al., 2019]. The success of transformer pooling inspired our use of this layer in both STiCa (chap. 3) and Support-Set (chap. 4) as a late aggregation layer and we found it to be quite important for improving representation learning performance. This then lead to our follow-up work, Motionformer (chap. 6), where instead of using the transformer just as a late temporal aggregation block, we

attempted to replace the entire 3D-CNN backbone with a transformer, inspired by ViT [Dosovitskiy et al., 2021]. While the default self-attention block is very generic, it treats both the space and time dimensions in videos equally, which we argue is sub-optimal for modeling motion in videos. We proposed a new form of self-attention, *trajectory attention*, which aggregates features along implicitly determined motion paths, serving as a better inductive bias for video transformers. Given our initial results and other evidence in the literature [Fan et al., 2021; Arnab et al., 2021; Bertasius et al., 2021], we believe that the transformer, along with our trajectory attention block, will be an important component of video architectures used for representation learning.

Lastly, we were interested in how contrastive learning can be improved, given that its instance discrimination assumption is quite strong. We explored how using an attention-weighted reconstruction objective can be used as an auxiliary loss to extract shared semantics across videos (chap. 4), helping to alleviate the strictness of instance discrimination contrastive learning. While this approach does require a suitable text generator to work, we show it to be very effective for improving video-text representation learning.

In addition to representation learning, this thesis also explored how to make it easier to interpret deep representations. We developed the extremal perturbation framework (chap. 7) to understand the salient pixels and channels that are important for classification. Perturbation analysis is a very principled approach to the attribution problem, since it tries to directly estimate how changes in the input impact the output of the neural network. Several works have built upon our perturbation framework for interpreting deep image representations [Cooper et al., 2021; Yang et al., 2021; Khorram et al., 2021], and our work has even been extended to new domains such as explaining video networks [Li et al., 2021].

To develop more tools for interpretability, particularly in line with our work on multi-modal representation learning, we also explored how to automatically map multi-modal representations to human interpretable labels using clustering (chap. 8). In the case of multi-modal data, clustering is not trivial because clustering data

from multiple modalities, such as audio and video, can usually lead to two different sets of clusters [Alwassel et al., 2020]. In SeLaVi, we show how to solve this issue by viewing each modality as an augmentation of a common, latent concept, thus learning a single label for all the modalities of an input. Our work has since inspired a number of works that attempt to jointly cluster representations from multiple modalities [Brown et al., 2021; Chen et al., 2021]. Our method also serves as the clustering module in recent work on self-supervised object detection from unlabelled multi-modal videos [Afouras et al., 2021].

9.2 Future Work

Here we present a few areas that are exciting directions for our work:

Extreme multi-modal self-supervision. In this thesis, we have explored pairs of modalities (video-audio, video-text) for multi-modal self-supervision. While the combination of video, audio and text modalities have been used for multi-modal self-supervision [Alayrac et al., 2020; Akbari et al., 2021], we think that there is an opportunity to train representations from an even larger number of modalities such as optical flow and depth in tandem. Different modalities may capture different semantic features of the input and can lead to a better representation if the model can effectively correlate these signals. However, learning from an extreme number of modalities is also challenging because different modalities may have different semantic strengths and granularity [Kazakos et al., 2019; Alayrac et al., 2020] and learning speeds [Xiao et al., 2020; Wang et al., 2020]. There are also the challenges of training multiple modality encoders end-to-end using GPU memory, but recent advances in chip development [Warren and Vincent, 2020] may help mitigate this problem.

Joint Video and Image representations. As explored in depth in this thesis, video data is perfect for learning representations. The temporal dimension of video allows for multiple viewpoints of objects, and its rich multi-modal information offers

the opportunity to learn very semantic representations. However, most works that train representations on video data have been unable to push the state-of-the art in the image domain [Wu and Wang, 2021; Gordon et al., 2020; Purushwalkam and Gupta, 2020; Wang et al., 2019c]. Learning joint image and video representations from video data that can be applied to both image and video tasks is still a core challenge, however, we do think recent advances with the transformer architecture [Vaswani et al., 2017] will unlock this capability because image and video data can be tokenized in a consistent way [Bertasius et al., 2021] thus allowing for mixed dataset training [Bain et al., 2021].

Using Interpretability Techniques To Improve Self-Supervised Learning. To understand the bottlenecks in self-supervised learning, there has been embryonic research into understanding what type of representations are learnt using different self-supervised pretext tasks [Asano et al., 2020a] and how they differ compared to supervised pretraining [Epstein et al., 2020]. Using interpretability techniques such as feature visualization [Mahendran and Vedaldi, 2015], these works show that early self-supervised learning approaches such as rotation [Gidaris et al., 2018] are most effective at learning low-level features such as edge detectors, but struggle at higher-level semantic tasks such as object and scene classification. With self-supervised learning, there is a need to better and more precisely understand the relationship between the training paradigm and the representations learned, beyond just the performance on downstream tasks. Interpretability research has historically been constrained to the fully supervised domain [Zhang et al., 2017; Selvaraju et al., 2019], but given that this is not necessarily the current direction of the field, nor is it the way most mammals learn, it is important to expand and ground it in the self-supervised regime.

Bibliography

Triantafyllos Afouras, Yuki M. Asano, Francois Fagan, Andrea Vedaldi, and Florian Metze. Self-supervised object detection from audio-visual correspondence, 2021.

Hassan Akbari, Li Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. *ArXiv*, abs/2104.11178, 2021.

Jean-Baptiste Alayrac, Adrià Recasens, Rosalia Schneider, Relja Arandjelović, Jason Ramapuram, Jeffrey De Fauw, Lucas Smaira, Sander Dieleman, and Andrew Zisserman. Self-supervised multimodal versatile networks. In *Advances in Neural Information Processing Systems*, 2020.

Erin Alves, Devesh Bhatt, Brendan Hall, Kevin Driscoll, Anitha Murugesan, and John Rushby. Considerations in assuring safety of increasingly autonomous systems. *NASA*, 2018.

Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems*, 2020.

Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *International Conference on Computer Vision*, 2017.

Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021.

YM. Asano, C. Rupprecht, and A. Vedaldi. A critical analysis of self-supervision, or what we can learn from a single image. In *International Conference on Learning Representations*, 2020a.

Yuki M Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020b.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. *ArXiv*, abs/2104.00650, 2021.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, 2021.

Andrew Brown, Vicky S. Kalogeiton, and Andrew Zisserman. Face, body, voice: Video person-clustering with multiple modalities. *ArXiv*, abs/2105.09939, 2021.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020.

Brian Chen, Andrew Rouditchenko, Kevin Duarte, Hilde Kuehne, Samuel Thomas, Angie Boggust, R. Panda, Brian Kingsbury, R. Feris, David F. Harwath, J. Glass, M. Picheny, and Shih-Fu Chang. Multimodal clustering networks for self-supervised learning from unlabeled videos. *ArXiv*, abs/2104.12671, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Conference on Computer Vision and Pattern Recognition*, 2005.

Jessica Cooper, Ognjen Arandjelovic, and David Harrison. Believe the hipe: Hierarchical perturbation for fast and robust explanation of black box models. *ArXiv*, abs/2103.05108, 2021.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, 2013.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Conference on Computer Vision and Pattern Recognition*, 2005.

Virginia R. de Sa. Learning classification with unlabeled data. In *Advances in Neural Information Processing Systems*, 1994.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Gerald M Edelman. *Neural Darwinism: The theory of neuronal group selection.* Basic books, 1987.

Dave Epstein, Yiliang Shi, Eugene Wu, and Carl Vondrick. What's missing from self-supervised representation learning? 2020.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers, 2021.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *International Conference on Computer Vision*, 2019.

Ruth Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *International Conference on Computer Vision*, 2017.

Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *International Conference on Acoustics, Speech, & Signal Processing*, 2017.

Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Conference on Computer Vision and Pattern Recognition*, 2019.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.

Bryce Goodman and Seth Flaxman. European union regulations on algorithmic decision-making and a "right to explanation". *AI magazine*, 38(3):50–57, 2017.

Daniel Gordon, Kiana Ehsani, Dieter Fox, and Ali Farhadi. Watching the world go by: Representation learning from unlabeled videos. *arXiv preprint arXiv:2003.07990*, 2020.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *International Conference on Artificial Intelligence and Statistics*, 2010.

Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Conference on Computer Vision and Pattern Recognition*, 2006.

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *International Conference on Computer Vision Workshops*, 2019.

Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2018.

Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018.

John Jumper, R Evans, A Pritzel, T Green, M Figurnov, K Tunyasuvunakool, O Ronneberger, R Bates, A Zidek, A Bridgland, et al. High accuracy protein structure prediction using deep learning. *Fourteenth Critical Assessment of Techniques for Protein Structure Prediction (Abstract Book)*, 22:24, 2020.

Yannis Kalantidis, Mert Bulent Sariyildiz, Noe Pion, Philippe Weinzaepfel, and Diane Larlus. Hard negative mixing for contrastive learning. In *NeurIPS*, 2020.

M. Kalayeh, Nagendra Kamath, Lingyi Liu, and A. Chandrashekar. Watching too much television is good: Self-supervised audio-visual representation learning from movies and tv shows. *ArXiv*, abs/2106.08513, 2021.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *International Conference on Computer Vision*, 2019.

Saeed Khorram, Tyler Lawson, and F. Li. igos++: integrated gradient optimized saliency by bilateral perturbations. *Proceedings of the Conference on Health, Inference, and Learning*, 2021.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, and Rory Sayres. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International Conference on Machine Learning*, 2018.

Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Association for the Advancement of Artificial Intelligence*, 2019.

Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems*, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25: 1097–1105, 2012.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision*, 2011.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

Sangho Lee, Jiwan Chung, Youngjae Yu, Gunhee Kim, Thomas Breuel, Gal Chechik, and Yale Song. Automatic curation of large-scale datasets for audio-visual representation learning. *ArXiv*, abs/2101.10803, 2021.

Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *European Conference on Computer Vision*, 2018.

Zhenqiang Li, Weimin Wang, Zuoyue Li, Yifei Huang, and Yoichi Sato. Towards visually explaining video understanding networks with perturbation. In *Winter Conference on Applications of Computer Vision*, 2021.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, 2014.

Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: fast feature extraction and svm training. In *Conference on Computer Vision and Pattern Recognition*, 2011.

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *BMVC*, 2019.

David G Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, 1999.

Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Learning audio-visual representations with active contrastive coding. In *International Conference on Learning Representations*, 2021a.

Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Contrastive self-supervised learning of global-local audio-visual representations, 2021b. URL `https://openreview.net/forum?id=Py4VjN6V2JX`.

A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Conference on Computer Vision and Pattern Recognition*, 2015.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *arXiv preprint arXiv:1908.09635*, 2019.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *International Conference on Computer Vision*, 2019.

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Marvin Minsky and Seymour A Papert. *Perceptrons: An introduction to computational geometry*. MIT press, 1969.

Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Conference on Computer Vision and Pattern Recognition*, 2020.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, 2016.

Alexander Mordvintsev, Nicola Pezzotti, Ludwig Schubert, and Chris Olah. Differentiable image parameterizations. *Distill*, 2018. doi: 10.23915/distill.00012. https://distill.pub/2018/differentiable-parameterizations.

Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *Advances in Neural Information Processing Systems*, 2016.

Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. 2017.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. doi: 10.23915/distill.00007. https://distill.pub/2017/feature-visualization.

Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *European Conference on Computer Vision*, 2018.

Nuno Vasconcelos Pedro Morgado, Ishan Misra. Robust audio-visual instance discrimination. In *CVPR*, 2021.

Karol J. Piczak. Esc: Dataset for environmental sound classification. In *ACM Multimedia*, 2015.

Senthil Purushwalkam and Abhinav Gupta. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. In *Advances in Neural Information Processing Systems*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *preprint*, 2019.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" explaining the predictions of any classifier. In *Proc. KDD*, 2016.

Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller. *Explainable AI: interpreting, explaining and visualizing deep learning*, volume 11700. Springer Nature, 2019.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *International Conference on Computer Vision*, 2017.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL `http://dx.doi.org/10.1007/s11263-019-01228-7`.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, 2014.

Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise, 2017.

Linda Smith and Michael Gasser. The development of embodied cognition: Six lessons from babies. *Artif. Life*, 11(1–2):13–30, January 2005. ISSN 1064-5462. doi: 10.1162/1064546053278973. URL `https://doi.org/10.1162/1064546053278973`.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human action classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net, 2014.

D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia*, 2015.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. 2020.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *International Conference on Computer Vision*, 2015.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. *International Journal of Computer Vision*, 128(7):1867–1888, Mar 2020. ISSN 1573-1405. doi: 10.1007/s11263-020-01303-4. URL `http://dx.doi.org/10.1007/s11263-020-01303-4`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.

Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *Conference on Computer Vision and Pattern Recognition*, 2019a.

Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Conference on Computer Vision and Pattern Recognition*, 2020.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4581–4591, 2019b.

Yufei Wang, Du Tran, and Lorenzo Torresani. Unidual: A unified model for image and video understanding. *arXiv preprint arXiv:1906.03857*, 2019c.

Tom Warren and James Vincent. Nvidia's first ampere gpu is designed for data centers and ai, not your pc, May 2020. URL `https://www.theverge.com/2020/5/14/21258419/nvidia-ampere-gpu-ai-data-centers-specs-a100-dgx-supercomputer`.

Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Haiping Wu and Xiaolong Wang. Contrastive learning of image representations with cross-video cycle-consistency, 2021.

Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Conference on Computer Vision and Pattern Recognition*, 2018.

Fanyi Xiao, Y. Lee, K. Grauman, J. Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *ArXiv*, abs/2001.08740, 2020.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision*, 2018.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Conference on Computer Vision and Pattern Recognition*, 2019.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Conference on Computer Vision and Pattern Recognition*, 2016.

Qing Yang, Xia Zhu, Yun Ye, Jong-Kae Fwu, Ganmei You, and Yuan Zhu. Mfpp: Morphological fragmental perturbation pyramid for black-box model explanations. *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 1376–1383, 2021.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014.

Matthew D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, Dec 2017. ISSN 1573-1405. doi: 10.1007/s11263-017-1059-x. URL `http://dx.doi.org/10.1007/s11263-017-1059-x`.

Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10):1084–1102, 2018.

Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3d features on any point-cloud. *ArXiv*, abs/2101.02691, 2021.

B. Zhou, D. Bau, A. Oliva, and A. Torralba. Interpreting deep visual representations via network dissection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41 (9):2131–2145, 2019.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *International Conference of Learning Representations*, 2015.

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition*, 2016.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Revisiting the importance of individual units in cnns via ablation, 2018.

Geoffrey Zweig, Polina Kuznetsova, Michael Auli, and Francois Fagan. Learning from videos to understand the world, 2021. URL `https://ai.facebook.com/blog/learning-from-videos-to-understand-the-world/`.

Appendices

Authorship statements

Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

Title of Paper	Understanding Deep Networks via Extremal Perturbations and Smooth Masks
Publication Status	✓Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Ruth Fong*, Mandela Patrick*, and Andrea Vedaldi. Understanding Deep Networks via Extremal Perturbations and Smooth Masks. IEEE International Conference on Computer Vision (ICCV) 2019.

Student Confirmation

Student Name:	Mandela Patrick	
Contribution to the Paper	*R.F and A.V. came up with the initial idea. R.F. and M.P. ran experiments and generated the figures. R.F., M.P., and A.V. jointly developed the methods into working algorithms and wrote the paper text.*	
Signature: Mandela Patrick	Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature: Andrea Vedaldi	Date	8 June 2021

Title of Paper	On Compositions of Transformations in Contrastive Self-Supervised Learning
Publication Status	✓ Published □ Accepted for Publication □ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Mandela Patrick*, Yuki Asano*, Polina Kuznetsova, Ruth Fong, João Henriques, Geoffrey Zweig, Andrea Vedaldi. On Compositions of Transformations in Contrastive Self-Supervised Learning. IEEE International Conference on Computer Vision (ICCV) 2021.

Student Confirmation

Student Name:	Mandela Patrick		
Contribution to the Paper	*M.P and Y.A. ideated, prototyped and developed the initial idea. M.P. and Y.A. designed the experiments and a majority of them were run by M.P. Figures were generated by Y.A. and M.P. and J.H., P.K. ran a large-scale experiment on IG65M. All authors wrote and edited the paper text.*		
Signature 		Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature 	Date	8 June 2021

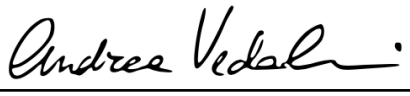
Title of Paper	Labelling Unlabelled Videos from Scratch Using Multi-Modal Self-Supervision
Publication Status	✓Published □ Accepted for Publication □Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Yuki Asano*, Mandela Patrick*, Christian Rupprecht and Andrea Vedaldi. Labelling Unlabelled Videos from Scratch Using Multi-Modal Self-Supervision. Advances of Neural Information Processing Systems (NeurIPS) 2020.

Student Confirmation

Student Name:	Mandela Patrick	
Contribution to the Paper	*M.P and Y.A. developed and prototyped the initial idea. Y.A. and M.P. ran and designed experiments and generated the figures. C.R. developed the webtool for visualisation and aided in fleshing out the method. All authors wrote and edited the paper text.*	
Signature 	Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature 	Date	8 June 2021

Title of Paper	Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning
Publication Status	✓Published □ Accepted for Publication □ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Mandela Patrick*, Yuki Asano*, Po-Yao Huang*, Ishan Misra, Florian Metze, João Henriques, Andrea Vedaldi. Space-Time Crop & Attend: Improving Cross-modal Video Representation Learning. IEEE International Conference on Computer Vision (ICCV) 2021.

Student Confirmation

Student Name:	Mandela Patrick		
Contribution to the Paper	*M.P and Y.A. ideated, prototyped and developed the initial idea. M.P. and Y.A. designed the experiments and a majority of them were run by M.P and P.H.. Figures were generated by Y.A. and M.P.. All authors wrote and edited the paper text.*		
Signature Mandela Patrick		Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature Andrea Vedaldi	Date	8 June 2021

Title of Paper	Support-Set Bottlenecks for Video-Text Representation Learning
Publication Status	✓Published □ Accepted for Publication □ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Mandela Patrick*, Po-Yao Huang*, Yuki Asano*, Florian Metze, Alexander Hauptmann, João Henriques, Geoffrey Zweig, Andrea Vedaldi. Support-Set Bottlenecks for Video-Text Representation Learning. International Conference of Learning Representations (ICLR) 2021.

Student Confirmation

Student Name:	Mandela Patrick		
Contribution to the Paper	*M.P, P.H. and Y.A. ideated, prototyped and developed the initial idea. M.P. and P.H. designed and ran the experiments. Figures were generated by Y.A., P.H., M.P. and J.H. All authors wrote and edited the paper text.*		
Signature Mandela Patrick		Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature Andrea Vedaldi	Date	8 June 2021

Title of Paper	Multilingual Multimodal Pre-training for Zero-Shot Cross-Lingual Transfer of Vision-Language Models
Publication Status	✓Published □ Accepted for Publication □ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Po-Yao Huang*, Mandela Patrick*, Junjie Hu, Graham Neubig, Florian Metze, Alexander Hauptmann. Multilingual Multimodal Pre-training for Zero-Shot Cross-Lingual Transfer of Vision-Language Models. North American Chapter of the Association for Computational Linguistics (NACCL) 2021 .

Student Confirmation

Student Name:	Mandela Patrick	
Contribution to the Paper	*P.H. and M.P ideated, prototyped and developed the initial idea. P.H. designed and ran the experiments. All authors wrote and edited the paper text.*	
Signature Mandela Patrick	Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature Andrea Vedaldi	Date	8 June 2021

Title of Paper	Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers
Publication Status	□Published □ Accepted for Publication ✓Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style
Publication Details	Submitted to Advances in Neural Information Processing Systems (NeurIPS) 2021 Mandela Patrick*, Dylan Campbell*, Yuki Asano*, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, João Henriques. Keeping Your Eye on the Ball: Trajectory Attention in Video Transformers. ArXiv.

Student Confirmation

Student Name:	Mandela Patrick		
Contribution to the Paper	*M.P, D.C., Y.A., J.H. and A.V. ideated, prototyped and developed the initial idea. M.P. designed and ran the experiments. Figures were generated by Y.A. All authors wrote and edited the paper text.*		
Signature Mandela Patrick		Date	07 / 06 / 2021

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Andrea Vedaldi		
Supervisor comments The description of the contributions is accurate.		
Signature Andrea Vedaldi	Date	8 June 2021