Keeping Your Eye on the Ball: Trajectory Attention for Video Transformers

Mandela Patrick\textsuperscript{1}, Dylan Campbell\textsuperscript{2}, Yuki M. Asano\textsuperscript{2}
Ishan Misra\textsuperscript{1}, Florian Metze\textsuperscript{1}, Christoph Feichtenhofer\textsuperscript{1}
Andrea Vedaldi\textsuperscript{1}, João F. Henriques\textsuperscript{2}

\textsuperscript{1}Facebook AI
\textsuperscript{2}Visual Geometry Group, Oxford
 Equal contribution
Recognising actions in video

- **Proxy** for other video recognition tasks (≈ classification for images)
- Often requires **fine-grained** distinctions between subtle motions
- Often requires **long-range** associations
- E.g.: swing dancing vs. salsa dancing; dribbling basketball vs. dunking basketball; catching ball vs. throwing ball; ...
Recognising actions in video
Recognising actions in video

- Camera motion
- Object motion
Background

Convolutional networks

Convolutions limit the receptive field, both spatially and temporally

- Alleviated with *atrous* convolution
- Receptive field varies with resolution and framerate; can be difficult to tune

Transformer networks

- Long-range associations / receptive field covers the **full input** at all stages
- Very little inductive bias compared to CNNs ⇒ often harder to train, but more flexible
- Computation grows quadratically with input ($\mathcal{O}(S^2T^2)$ for input with $T$ frames and $S$ pixels)
Inductive biases in video processing

Physical motivation:

• Camera motion does not affect scene properties
• Motion path and appearance of an object can be disentangled
  • Translation equivariance is a subset of this desired behaviour

Advantages:

• Data efficiency
• Extrapolation beyond training set (generalization)
• Sometimes:
  • Improves computational efficiency
  • Reduces # of parameters and overfitting
Video attention strategies: joint space-time
Video attention strategies: joint space-time
Video attention strategies: joint space-time

Softmax normalization across volume
Video attention strategies: joint space-time
Video attention strategies: joint space-time

- Computational complexity: $\mathcal{O}(S^2T^2)$
- Infeasible for long and high-res videos
- Can we get closer to $\mathcal{O}(ST)$?

Bertasius et al., Is space-time attention all you need for video understanding? In ICML, 2021.
Arnab et al., Vivit: A video vision transformer, 2021.
Video attention strategies: divided space-time
Video attention strategies: divided space-time
Video attention strategies: divided space-time

Softmax normalization
Video attention strategies: divided space-time
Video attention strategies: divided space-time
Video attention strategies: divided space-time
Video attention strategies: divided space-time

- Significant computation/memory gains: spatial att $O(S^2T)$, temporal att $O(ST^2)$
- Still has a quadratic bottleneck in each dimension
- Axis-aligned pooling is artificial

Moving camera + moving objects

Bertasius et al., Is space-time attention all you need for video understanding? In ICML, 2021.
Arnab et al., ViViT: A video vision transformer, 2021.
Aim: find other patches that contain the ball and aggregate their information into a single output

Why?
- To leverage **multiple views** of the same object to better understand its properties
- To reason about the **motion** of the object

How?
- **Attention**: computes feature similarities across space-time and pools information
Trajectory attention
Trajectory attention

Softmax normalization
per frame
Trajectory attention
Trajectory attention
Trajectory attention
Trajectory attention

- Overall complexity: $\mathcal{O}(S^2T^2)$
  - No better than before
  - Needs to be improved by other means
Idea: Take inspiration from matrix factorization methods / low-rank decomposition

- Not just multiplication, in general: $A \approx f(\cdot, \cdot)$
- Due to the softmax, attention matrices usually have high rank
  $\Rightarrow$ Poorly approximated by PCA/low-rank decompositions
- Prototypes must be few, and representative of all keys/queries

Cost of multiplying this matrix by an arbitrary vector: $\Theta(S^2T^2) \rightarrow \Theta(STP)$


Formulate attention probabilistically

- Attention operator defines a **parametric model** of the probability of event $A_{ij}$ (assignment of key $j$ to query $i$), with a multinomial logistic function:

  \[ P(A_{ij}) = S(q_i^T K) \]

  - Softmax
  - Query vector
  - Key vectors (concatenated)

- Introduce **latent variables** $U_{jl}$ (assignment of key $j$ to prototype $l$)

- Then (without approximation):

  \[ P(A_{ij}) = \sum_{\ell} P(A_{ij} | U_{\ell j}) P(U_{\ell j}) \]

- But, $P(A | U)$ is intractable $\Rightarrow$ approximate with a similar parametric model

- All together:

  \[ \tilde{P}(A)V = S(Q^T P) (S(P^T K)V) \]

Computational efficiency

- $\mathcal{O}(S^2 T^2) \rightarrow \mathcal{O}(STP)$
Selecting prototypes

Priorities:

• **Dynamically** adjust to keys/queries to ensure their region is reconstructed well
• Minimize **redundancy** between prototypes

Some suboptimal choices:

• Trainable vectors *(not adaptive)*
• Random sampling from keys/queries *(often selects collinear vectors)*
• Clustering keys/queries online *(expensive)*
**Objective:**
Select the **most orthogonal** subset of keys/queries

**A greedy algorithm:**

\[ X \leftarrow \text{random subset of } K \cup Q \]

For \( l \in \{1, \ldots, |P|\} \):

\[ i^* \leftarrow \arg\min_i \sum_{j=1}^{l-1} \left| \langle X_i, P_j \rangle \right| \]

\[ P_l \leftarrow X_{i^*} \]
Experiments: approximating attention

Comparison to state-of-the-art on the Long Range Arena benchmark

<table>
<thead>
<tr>
<th>Model</th>
<th>ListOps</th>
<th>Text</th>
<th>Retrieval</th>
<th>Image</th>
<th>Pathfinder</th>
<th>Avg↑</th>
<th>GFLOPS↓</th>
<th>Mem.↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact [76]</td>
<td>36.69</td>
<td>63.09</td>
<td>78.22</td>
<td>31.47</td>
<td>66.35</td>
<td>55.16</td>
<td>1.21</td>
<td>4579</td>
</tr>
<tr>
<td>Performer-256 [14]</td>
<td>36.69</td>
<td>63.22</td>
<td>78.98</td>
<td>29.39</td>
<td>66.55</td>
<td>54.97</td>
<td>0.49</td>
<td>885</td>
</tr>
<tr>
<td>Nyströmformer-128 [85]</td>
<td>36.90</td>
<td>64.17</td>
<td>78.67</td>
<td>36.16</td>
<td>52.32</td>
<td>53.64</td>
<td>0.62</td>
<td>745</td>
</tr>
<tr>
<td>Orthoformer-64</td>
<td>33.87</td>
<td>64.42</td>
<td>78.36</td>
<td>33.26</td>
<td>66.41</td>
<td>55.26</td>
<td>0.24</td>
<td>344</td>
</tr>
</tbody>
</table>

- Best overall results with far fewer prototypes (64) than other methods
- About **half** the memory and GFLOPS of the best approximations
- **No loss** of performance on average (unlike the other approximations)
### Experiments: approximating attention

**Comparison on action recognition datasets (Kinetics-400, Something-Something)**

(a) Orthoformer is competitive with Nyström.

<table>
<thead>
<tr>
<th>Attention</th>
<th>Approx.</th>
<th>Mem.</th>
<th>K-400</th>
<th>SSv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory (E)</td>
<td>N/A</td>
<td>7.4</td>
<td>79.7</td>
<td>66.5</td>
</tr>
<tr>
<td>Trajectory (A)</td>
<td>Performer</td>
<td>5.1</td>
<td>72.9</td>
<td>52.7</td>
</tr>
<tr>
<td>Nyströmformer</td>
<td>3.8</td>
<td>77.5</td>
<td>64.0</td>
<td></td>
</tr>
<tr>
<td><strong>Orthoformer</strong></td>
<td>3.6</td>
<td>77.5</td>
<td>63.8</td>
<td></td>
</tr>
</tbody>
</table>

(b) Selecting orthogonal prototypes is the best strategy.

<table>
<thead>
<tr>
<th>Attention</th>
<th>Selection</th>
<th>Mem.</th>
<th>K-400</th>
<th>SSv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trajectory (E)</td>
<td>N/A</td>
<td>7.4</td>
<td>79.7</td>
<td>66.5</td>
</tr>
<tr>
<td>Trajectory (A)</td>
<td>Seg-Means</td>
<td>3.6</td>
<td>75.8</td>
<td>60.3</td>
</tr>
<tr>
<td>Random</td>
<td>Orthogonal</td>
<td>3.6</td>
<td>77.5</td>
<td>63.8</td>
</tr>
</tbody>
</table>
Experiments: setup

Application: action recognition

- Use ViT [1] as the base model (12 layers / 12 attention heads / embeddings size 768)
- Separate space and time positional encodings (TimeSformer [2])
- Cubic image tokenization (ViViT [3])
- Adding our Trajectory Attention

Datasets:
- Kinetics-400/600 (*appearance cues are more dominant*)
- Something-Something V2 (*motion cues are more dominant*)
- Epic Kitchens 100

Keeps objects consistent across different action classes

---

Experiments: setup

Train model on **single frames only** and assess drop in performance

![Bar chart showing performance drop]

- **39% performance drop**
Comparison of attention mechanisms

<table>
<thead>
<tr>
<th>Attention</th>
<th>K-400</th>
<th>SSv2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint Space-Time</td>
<td>79.2</td>
<td>64.0</td>
</tr>
<tr>
<td>Divided Space-Time</td>
<td>78.5</td>
<td>64.2</td>
</tr>
<tr>
<td>Trajectory</td>
<td><strong>79.7</strong></td>
<td><strong>66.5</strong></td>
</tr>
</tbody>
</table>
Experiments: benchmark results

(a) Something–Something V2

<table>
<thead>
<tr>
<th>Model</th>
<th>Pretrain</th>
<th>Top-1</th>
<th>Top-5</th>
<th>GFLOPs × views</th>
</tr>
</thead>
<tbody>
<tr>
<td>SlowFast</td>
<td>K-400</td>
<td>61.7</td>
<td>-</td>
<td>65.7×3×1</td>
</tr>
<tr>
<td>TSM</td>
<td>K-400</td>
<td>63.4</td>
<td>88.5</td>
<td>62.4×3×2</td>
</tr>
<tr>
<td>STM</td>
<td>IN-1K</td>
<td>64.2</td>
<td>89.8</td>
<td>66.5×3×10</td>
</tr>
<tr>
<td>MSNet</td>
<td>IN-1K</td>
<td>64.7</td>
<td>89.4</td>
<td>67×1×1</td>
</tr>
<tr>
<td>TEA</td>
<td>IN-1K</td>
<td>65.1</td>
<td>-</td>
<td>70×3×10</td>
</tr>
<tr>
<td>bLVNet</td>
<td>IN-1K</td>
<td>65.2</td>
<td>90.3</td>
<td>128.6×3×10</td>
</tr>
<tr>
<td>VidTr-L</td>
<td>IN-21K+K-400</td>
<td>60.2</td>
<td>-</td>
<td>351×3×10</td>
</tr>
<tr>
<td>Tformer-L</td>
<td>IN-21K</td>
<td>62.5</td>
<td>-</td>
<td>1703×3×1</td>
</tr>
<tr>
<td>ViViT-L</td>
<td>IN-21K+K-400</td>
<td>65.4</td>
<td>89.8</td>
<td>3992×4×3</td>
</tr>
<tr>
<td>MViT-B</td>
<td>K-400</td>
<td>67.1</td>
<td>90.8</td>
<td>170×3×1</td>
</tr>
<tr>
<td>Mformer</td>
<td>IN-21K+K-400</td>
<td>66.5</td>
<td>90.1</td>
<td>369.5×3×1</td>
</tr>
<tr>
<td>Mformer-L</td>
<td>IN-21K+K-400</td>
<td>68.1</td>
<td>91.2</td>
<td>1185.1×3×1</td>
</tr>
<tr>
<td>Mformer-HR</td>
<td>IN-21K+K-400</td>
<td>67.1</td>
<td>90.6</td>
<td>958.8×3×1</td>
</tr>
</tbody>
</table>

(b) Kinetics-400

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrain</th>
<th>Top-1</th>
<th>Top-5</th>
<th>GFLOPs × views</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D [10]</td>
<td>IN-1K</td>
<td>72.1</td>
<td>89.3</td>
<td>108×N/A</td>
</tr>
<tr>
<td>R(2+1)D [75]</td>
<td>-</td>
<td>72.0</td>
<td>90.0</td>
<td>152×5×23</td>
</tr>
<tr>
<td>S3D-G [84]</td>
<td>IN-1K</td>
<td>74.7</td>
<td>93.4</td>
<td>142.8×N/A</td>
</tr>
<tr>
<td>X3D-XL [24]</td>
<td>-</td>
<td>79.1</td>
<td>93.9</td>
<td>48.4×3×10</td>
</tr>
<tr>
<td>SlowFast</td>
<td>-</td>
<td>79.8</td>
<td>93.9</td>
<td>234×3×10</td>
</tr>
<tr>
<td>VTN [51]</td>
<td>IN-21K</td>
<td>78.6</td>
<td>93.7</td>
<td>4218×1×1</td>
</tr>
<tr>
<td>VidTr-L</td>
<td>IN-21K</td>
<td>79.1</td>
<td>93.9</td>
<td>392×3×10</td>
</tr>
<tr>
<td>Tformer-L</td>
<td>IN-21K</td>
<td>80.7</td>
<td>94.7</td>
<td>2380×3×1</td>
</tr>
<tr>
<td>MViT-B</td>
<td>-</td>
<td>81.2</td>
<td>95.1</td>
<td>455×3×3</td>
</tr>
<tr>
<td>ViViT-L</td>
<td>IN-21K</td>
<td>81.3</td>
<td>94.7</td>
<td>3992×3×4</td>
</tr>
<tr>
<td>Mformer</td>
<td>IN-21K</td>
<td>79.7</td>
<td>94.2</td>
<td>369.5×3×10</td>
</tr>
<tr>
<td>Mformer-L</td>
<td>IN-21K</td>
<td>80.2</td>
<td>94.8</td>
<td>1185.1×3×10</td>
</tr>
<tr>
<td>Mformer-HR</td>
<td>IN-21K</td>
<td>81.1</td>
<td>95.2</td>
<td>958.8×3×10</td>
</tr>
</tbody>
</table>

- **SOTA** on SSv2 (+1%), which is more reliant on motion cues
- Competitive with the much larger ViViT-L model on K400
Experiments: benchmark results

<table>
<thead>
<tr>
<th>Method</th>
<th>Pretrain</th>
<th>A</th>
<th>V</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSN [78]</td>
<td>IN-1K</td>
<td>33.2</td>
<td>60.2</td>
<td>46.0</td>
</tr>
<tr>
<td>TRN [86]</td>
<td>IN-1K</td>
<td>35.3</td>
<td>65.9</td>
<td>45.4</td>
</tr>
<tr>
<td>TBN [36]</td>
<td>IN-1K</td>
<td>36.7</td>
<td>66.0</td>
<td>47.2</td>
</tr>
<tr>
<td>TSM [46]</td>
<td>IN-1K</td>
<td>38.3</td>
<td>67.9</td>
<td>49.0</td>
</tr>
<tr>
<td>SlowFast</td>
<td>K-400</td>
<td>38.5</td>
<td>65.6</td>
<td>50.0</td>
</tr>
<tr>
<td>ViViT-L</td>
<td>IN-21K+K-400</td>
<td>44.0</td>
<td>66.4</td>
<td>56.8</td>
</tr>
<tr>
<td>Mformer</td>
<td>IN-21K+K-400</td>
<td>43.1</td>
<td>66.7</td>
<td>56.5</td>
</tr>
<tr>
<td>Mformer-L</td>
<td>IN-21K+K-400</td>
<td>44.1</td>
<td>67.1</td>
<td>57.6</td>
</tr>
<tr>
<td>Mformer-HR</td>
<td>IN-21K+K-400</td>
<td>44.5</td>
<td>67.0</td>
<td>58.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>Pretrain</th>
<th>Top-1</th>
<th>Top-5</th>
<th>GFLOPs x views</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttnNAS [81]</td>
<td>-</td>
<td>79.8</td>
<td>94.4</td>
<td>-</td>
</tr>
<tr>
<td>LGD-3D [56]</td>
<td>IN-1K</td>
<td>81.5</td>
<td>95.6</td>
<td>-</td>
</tr>
<tr>
<td>SlowFast [25]</td>
<td>-</td>
<td>81.8</td>
<td>95.1</td>
<td>234x3x10</td>
</tr>
<tr>
<td>X3D-XL [25]</td>
<td>-</td>
<td>81.9</td>
<td>95.5</td>
<td>48.4x3x10</td>
</tr>
<tr>
<td>Tformer-HR [7]</td>
<td>IN-21K</td>
<td>82.4</td>
<td>96.0</td>
<td>1703x3x1</td>
</tr>
<tr>
<td>ViViT-L [2]</td>
<td>IN-21K</td>
<td>83.0</td>
<td>95.7</td>
<td>3992x3x4</td>
</tr>
<tr>
<td>MViT-B-24 [22]</td>
<td>-</td>
<td>83.8</td>
<td>96.3</td>
<td>236x1x5</td>
</tr>
<tr>
<td>Mformer</td>
<td>IN-21K</td>
<td>81.6</td>
<td>95.6</td>
<td>369.5x3x10</td>
</tr>
<tr>
<td>Mformer-L</td>
<td>IN-21K</td>
<td>82.2</td>
<td>96.0</td>
<td>1185.1x3x10</td>
</tr>
<tr>
<td>Mformer-HR</td>
<td>IN-21K</td>
<td>82.7</td>
<td>96.1</td>
<td>958.8x3x10</td>
</tr>
</tbody>
</table>

- **SOTA** on Epic-Kitchens Nouns (+2.3%), which is more reliant on motion cues
- Competitive performance on K600
Experiments: attention maps

![Attention Maps Example]
Conclusions

✓ Aggregating information along implicit motion trajectories can inject a helpful inductive bias into video transformers

✓ Quadratic dependency on input size can be reduced to linear

✓ Orthogonality is the most effective prototype selection criteria

✓ SOTA results on motion-focused datasets
Thank you

Mandela Patrick

Dylan Campbell

Yuki M. Asano

Ishan Misra

Florian Metze

Christoph Feichtenhofer

Andrea Vedaldi

João F. Henriques

University of Oxford / Facebook AI Research

Equal Contribution