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

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

Facebook AI Research and Visual Geometry Group, University of Oxford

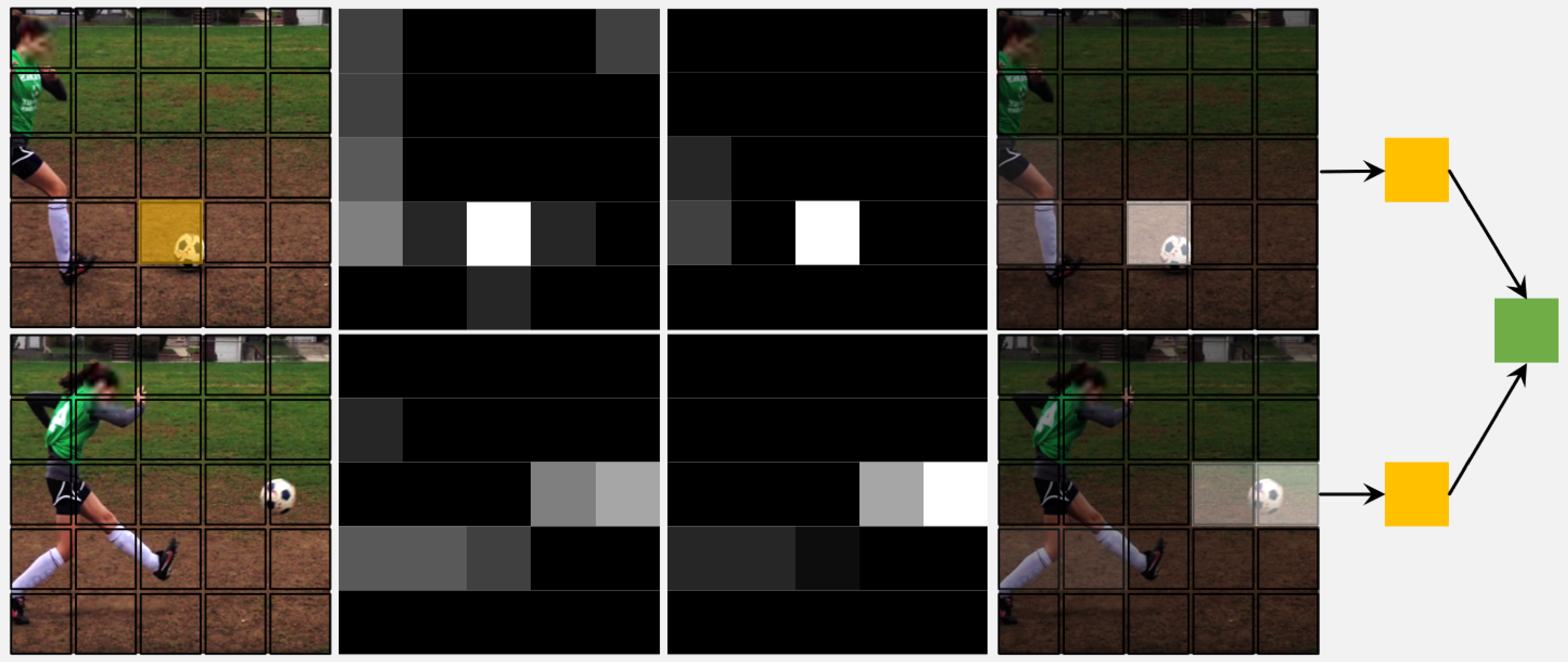
*Equal contribution †mandelapatrick1@gmail.com

Motivation

- **Action recognition** requires fine-grained distinctions between subtle motions that evolve over many frames.
- **Video transformers** can make the requisite long-range associations, but have very little inductive bias and quadratic complexity.
- Can we design an attention module that has a motion-inductive bias and is computationally tractable for long videos?

Trajectory Attention

- **Aim:** Find all patches with the same object and pool information.
- **Why?** Multiple views to better understand the object and its motion.
- **How?** Attention to compute and pool patch similarities over space-time.



(a) (b) (c) (d) (e)

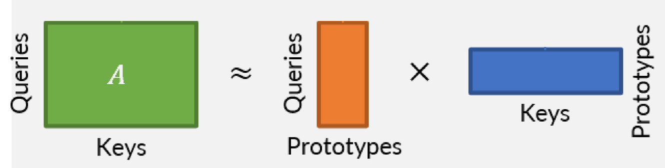
A reference patch (a) is compared to every other patch to obtain a heatmap (b), which is softmax-normalized *per frame* (c). A weighted sum is taken for each frame (d), resulting in one embedding vector per frame per reference patch, which are pooled via 1D temporal attention (e).

Orthoformer Linear Attention Approximation

- **Formulate attention probabilistically:** a parametric model of the probability of event A_{ij} (assignment of key j to query i).
- Introduce latent variables $U_{j\ell}$ (assignment of key j to prototype ℓ).

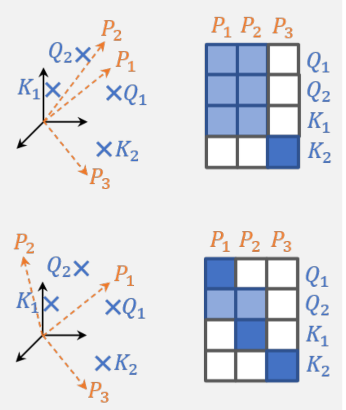
$$P(A_{i:}) = \mathcal{S}(\mathbf{q}_i^\top \mathbf{K})$$

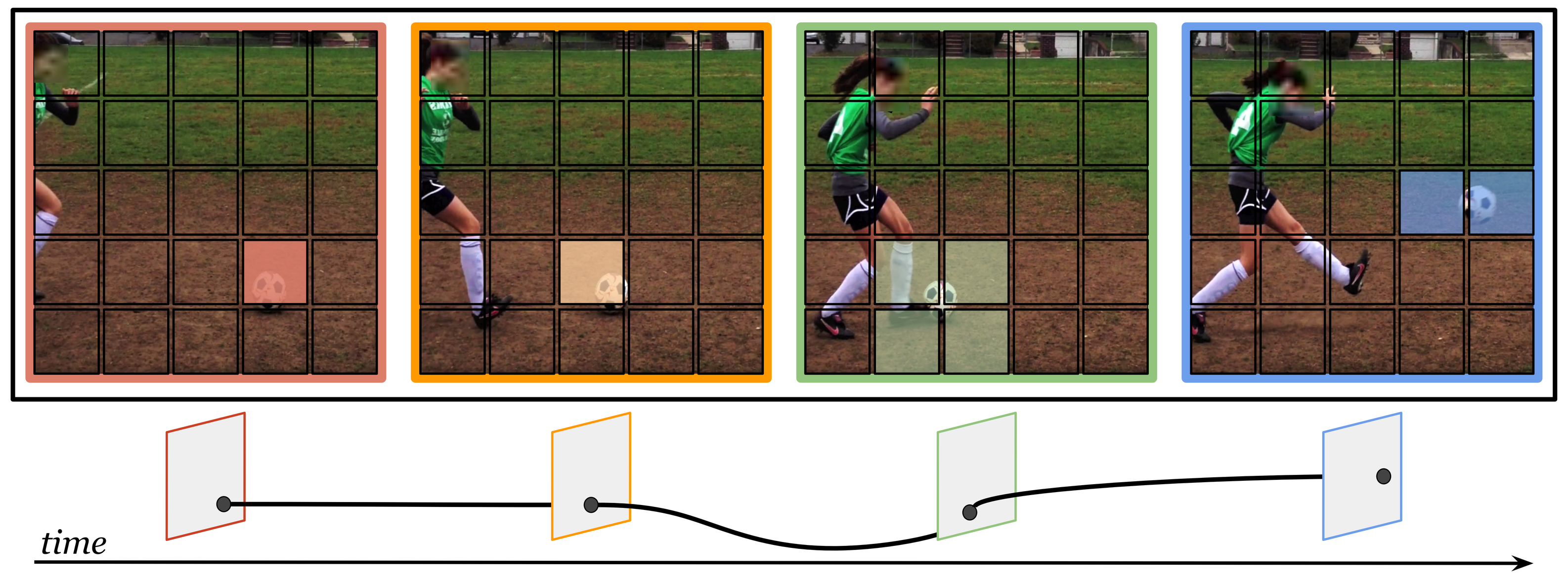
$$\tilde{P}(A)\mathbf{V} = \mathcal{S}(\mathbf{Q}^\top \mathbf{P})(\mathcal{S}(\mathbf{P}^\top \mathbf{K})\mathbf{V})$$



$$P(A_{ij}) = \textstyle\sum_\ell P(A_{ij} \mid U_{\ell j}) P(U_{\ell j})$$

Dynamically select the most orthogonal keys/queries as prototypes to minimize redundancy.





Ablation Studies

Table: **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**

Table: **Trajectory Approximation ablations:** We ablate various aspects of our attention approximation: 1) approximation method 2) prototype selection strategy.

(a) Orthoformer is competitive with Nyströmformer.

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**

Long Range Arena Benchmark

Table: **Comparison to the state-of-the-art on Long Range Arena benchmark.** GFLOPS and CUDA maximum Memory (MB) are reported for the ListOps task

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

- Best overall results with far fewer prototypes (64)
- About half the memory and GFLOPS
- No loss of performance on average

Comparison to State-of-the-Art Approaches

(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

- SOTA on SSv2 ($+1\%$), which is more reliant on motion cues
- Competitive with the much larger ViViT-L model on K400

(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**	<u>67.0</u>	**58.5**

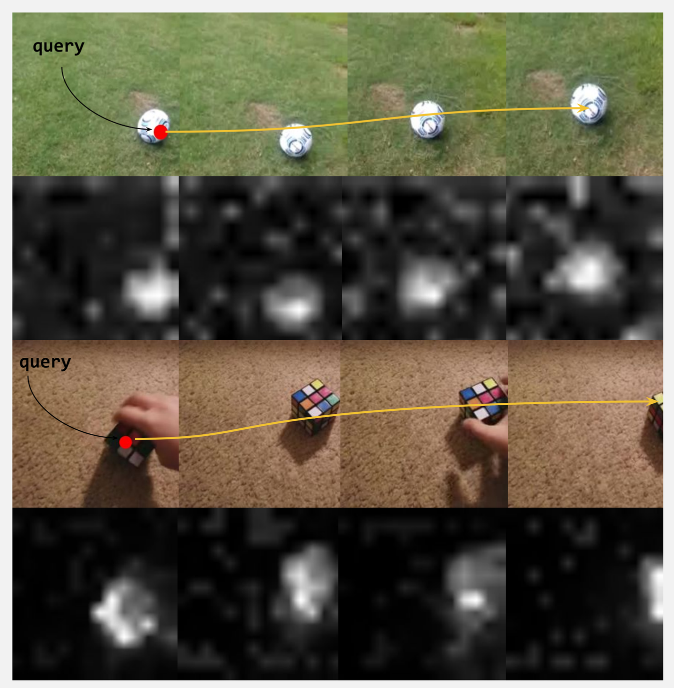
(d) **Kinetics-600**

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	<u>82.7</u>	96.1	958.8×3×10

- SOTA on Epic-Kitchens Nouns ($+2.3\%$), competitive on K600

Trajectory Attention Maps

Learned attention maps implicitly track query points across time.



Key Conclusions

- **Trajectory Attention:** aggregating information along implicit motion trajectories injects a helpful inductive bias.
- **Orthoformer:** reduces quadratic complexity to linear.
- **SOTA** results on motion-dependent datasets such as Something-Something V2, and Epic-Kitchens.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021.

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

Code