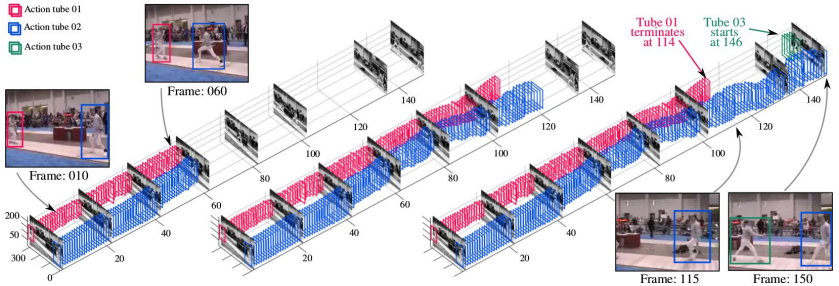
Structured Models for Video Understanding

Anurag Arnab

Google Research

Video Understanding

Video classification



Spatio-temporal detection

3D Pose Estimation

Video Segmentation

Video Understanding - Challenges

- Models [Arxiv 2021, Arxiv 2021]
  - Capture spatio-temporal relationships
  - Efficiently process long videos
    - Model must process *T* times more frames than an image architecture.
  - Fit to comparatively small datasets
- Data [CVPR 2019, ECCV 2020, IROS 2020]
  - Often weakly-supervised
  - Clean annotations for complex tasks are difficult and/or expensive to obtain.

Video Understanding - Challenges

- Models [Arxiv 2021, Arxiv 2021]
  - Capture spatio-temporal relationships
  - Efficiently process long videos
    - Model must process *T* times more frames than an image architecture.
  - Fit to comparatively small datasets
- Data [CVPR 2019, ECCV 2020, IROS 2020]
  - Often weakly-supervised
  - Clean annotations for complex tasks are difficult and/or expensive to obtain.

Unified Graph Structured Models for Video Understanding

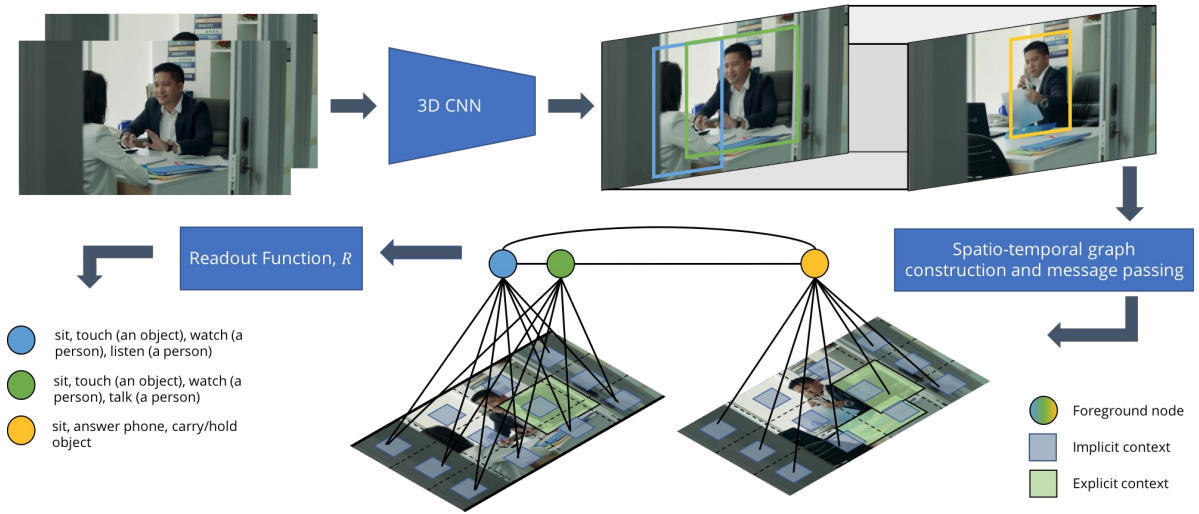
Google Research

Introduction

- Video understanding requires reasoning about spatio-temporal interactions between actors, objects and the environment.
- Often over long time intervals

Model

- Explicitly model spatio-temporal relationships explicitly with graph neural networks.
- Show how previous structured models for video are special cases of ours.



Model

- Based on Message Passing Neural Networks [1]
  - Generalises many previous GNNs
- Message-passing phase
  - Pass messages from nodes to their neighbours
  - Update internal state of each node
- Readout phase
  - Classify nodes based on their updated state

$$m_v^{i+1} = \sum_{w \in \mathcal{N}} M(h_v^i, h_w^i; \theta_s^i)$$

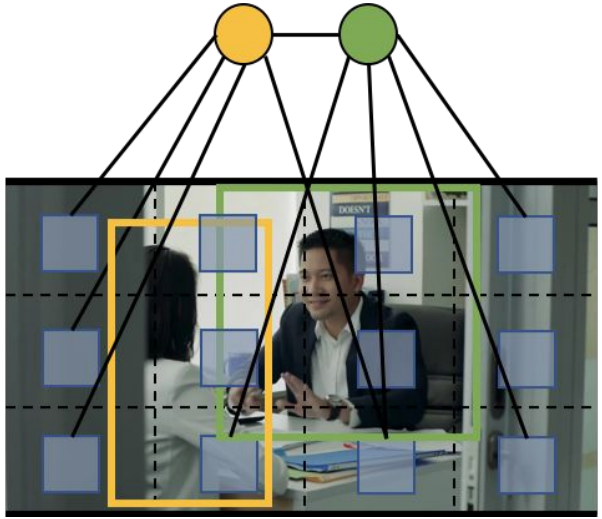
$$h_v^{i+1} = U(m_v^{i+1}, h_v^i)$$

$$y = R(\{h_v^i\} | v \in \mathcal{G})$$

[1] Gilmer *et al.* ICML 2017

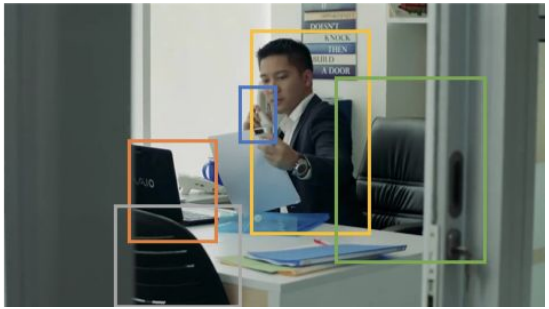
Spatial Model

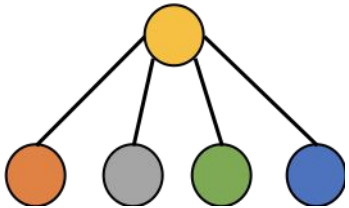
- Model interactions between actors, objects and environment in a single keyframe
- Actor nodes are ROI-aligned, spatio-temporally pooled features from a person detector.
- *Implicit model*
  - Use cells of feature map as object
  - Does not require supervision of objects
- Dense connections to actors.



Spatial Model

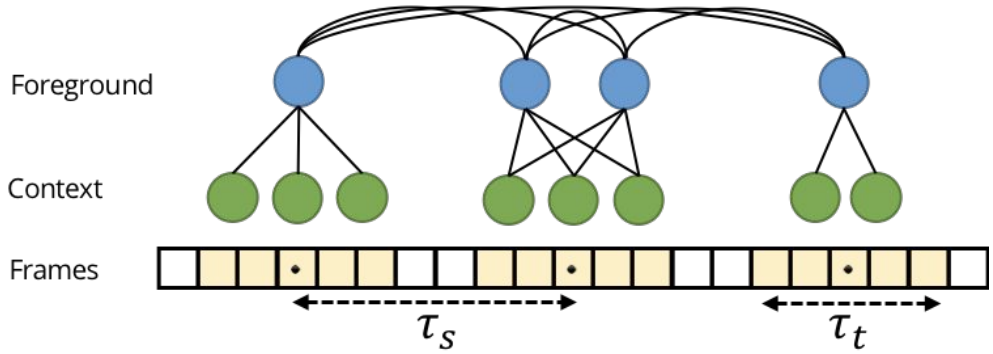
- Model interactions between actors, objects and environment in a single keyframe
- *Explicit model*
  - Use Region Proposal Network trained on Open-Images
  - Assume that RPN will extract object proposals that are discriminative of actions.





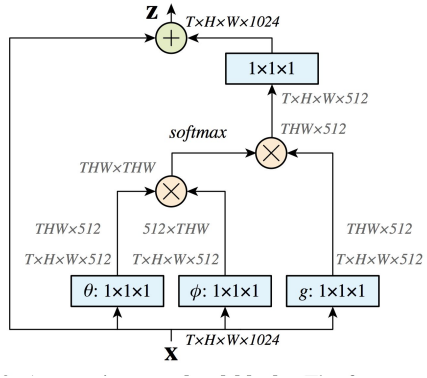
Temporal Model

- Densely connect actor nodes from different keyframes
- Connectivity controlled by
  - Temporal context, τ_c , number of keyframes considered
  - Temporal stride, τ_s , stride at which to select keyframes. Larger stride enlarges temporal receptive field.
- Temporal receptive field, τ_t , of CNN features.

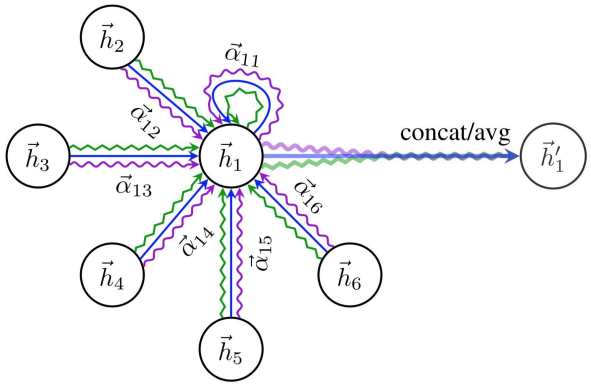


Message Passing Functions

- Non-local / Multihead Self-Attention
  - Modified to only update actor nodes
- Graph attention (GAT)
- Combination of above



Non-local



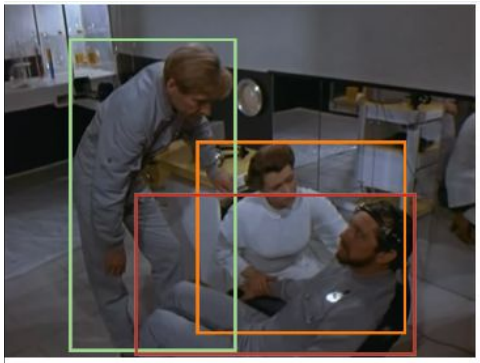
Graph-Attention

Relation to prior work

- Wu *et al*. CVPR 2019
  - Only temporal model. Non-local for message passing
- Girdhar *et al*. CVPR 2019
  - Only spatial model. Implicit objects. Non-local
- Sun *et al*. ECCV 2018
  - Only spatial model. Implicit objects. Relation networks
- Zhang *et al*. CVPR 2019
  - Spatio-temporal model. But actually uses three separate graphs, with hand-defined similarity functions which is similar to GAT.

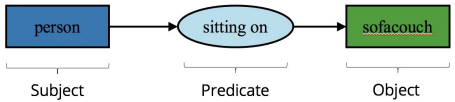
Experiments

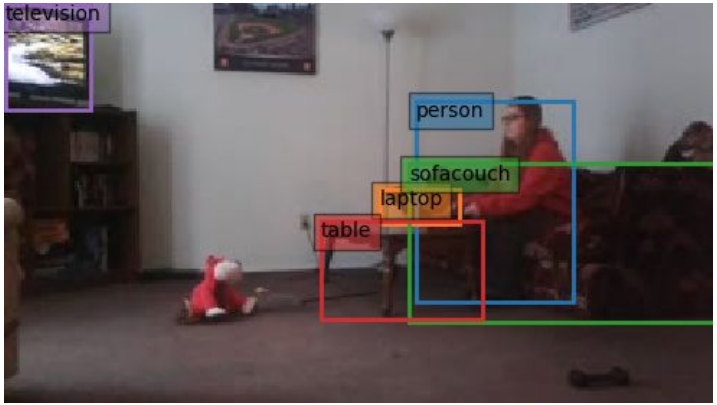
- Evaluate on two tasks that require reasoning about interactions in video
  - Action detection
  - Scene graphs

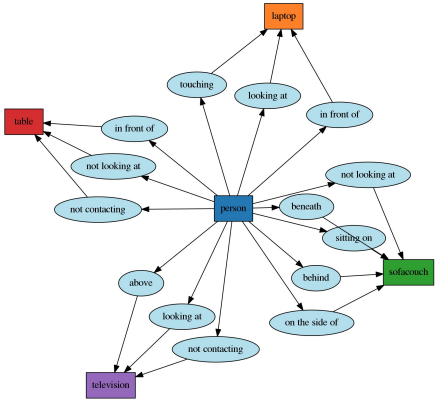


Experiments

- Evaluate on two tasks that require reasoning about interactions in video
  - Action detection
  - Scene graphs







Experiments on AVA

- AVA is an action recognition dataset.
- Actions divided into three types: Pose, Human-Human, Human-Object

Message passing neighbourhood	Pose	Human-Human	Human-Object	Overall
Baseline (None)	43.1	25.2	17.4	24.8
Actors only	43.2	27.0	17.8	25.6
Implicit objects only	43.4	26.7	18.0	25.7
Explicit objects only	43.0	26.7	17.8	25.5
Actor + Implicit	43.4	26.8	18.3	25.9
Actor + Implicit + Explicit	43.7	27.0	18.4	26.1
Spatio-temporal	43.7	27.5	19.7	26.8

Experiments on AVA

- AVA is an action recognition dataset.
- Actions divided into three types: Pose, Human-Human, Human-Object

Message passing neighbourhood	Pose	Human-Human	Human-Object	Overall
Baseline (None)	43.1	25.2	17.4	24.8
Actors only	43.2	27.0	17.8	25.6
Implicit objects only	43.4	26.7	18.0	25.7
Explicit objects only	43.0	26.7	17.8	25.5
Actor + Implicit	43.4	26.8	18.3	25.9
Actor + Implicit + Explicit	43.7	27.0	18.4	26.1
Spatio-temporal	43.7	27.5	19.7	26.8

Experiments on AVA

- AVA is an action recognition dataset.
- Actions divided into three types: Pose, Human-Human, Human-Object

Message passing neighbourhood	Pose	Human-Human	Human-Object	Overall
Baseline (None)	43.1	25.2	17.4	24.8
Actors only	43.2	27.0	17.8	25.6
Implicit objects only	43.4	26.7	18.0	25.7
Explicit objects only	43.0	26.7	17.8	25.5
Actor + Implicit	43.4	26.8	18.3	25.9
Actor + Implicit + Explicit	43.7	27.0	18.4	26.1
Spatio-temporal	43.7	27.5	19.7	26.8

Experiments on AVA

- AVA is an action recognition dataset.
- Actions divided into three types: Pose, Human-Human, Human-Object

Message passing neighbourhood	Pose	Human-Human	Human-Object	Overall
Baseline (None)	43.1	25.2	17.4	24.8
Actors only	43.2	27.0	17.8	25.6
Implicit objects only	43.4	26.7	18.0	25.7
Explicit objects only	43.0	26.7	17.8	25.5
Actor + Implicit	43.4	26.8	18.3	25.9
Actor + Implicit + Explicit	43.7	27.0	18.4	26.1
Spatio-temporal	43.7	27.5	19.7	26.8

Experiments on AVA

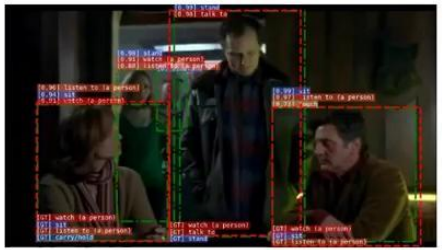
- AVA is an action recognition dataset.
- Actions divided into three types: Pose, Human-Human, Human-Object

Message passing neighbourhood	Pose	Human-Human	Human-Object	Overall
Baseline (None)	43.1	25.2	17.4	24.8
Actors only	43.2	27.0	17.8	25.6
Implicit objects only	43.4	26.7	18.0	25.7
Explicit objects only	43.0	26.7	17.8	25.5
Actor + Implicit	43.4	26.8	18.3	25.9
Actor + Implicit + Explicit	43.7	27.0	18.4	26.1
Spatio-temporal	43.7	27.5	19.7	26.8

Visualisation of spatial model

Input video

Predicted keyframes



Visualised Actor

Spatial messages received

Further ablation (Scene graphs)

(a) Message passing functions

	SGCls
	R@20
Baseline	48.9
Non-local in backbone	49.1
Non-local	50.4
GAT	51.1
GAT + Non-local	**51.3**

(b) Temporal graph structure

Temporal parameters		SGCls
τ_c	τ_s	R@20
Spatial only		51.1
3	2	52.9
3	5	53.3
3	7	53.5
5	2	53.4
5	5	**53.8**
5	7	53.6

(c) Message passing iterations

Iterations	SGCls
	R@20
1	51.1
2	51.6
3	51.6
5	51.8

(d) Comparison to existing methods

	SGCls		PredCls	
	R@20	R@50	R@10	R@20
MSDN [33]	44.0	47.2	–	–
IMP [61]	44.1	47.4	–	–
RelDN [66]	46.7	49.4	–	–
SlowFast (ResNet 50)	48.9	51.3	78.7	93.8
Ours (ResNet 50)	**53.8**	**56.0**	**79.3**	**94.2**

Action detection state-of-the-art

AVA

Method	v2.1	v2.2
ACRN (S3D) [54]	17.4	–
Zhang *et al.* (R50) [71]	22.2	–
SlowFast baseline (R50)	24.5	24.8
Girdhar *et al.* (I3D) [14]	25.0	–
LFB (R50) [61]	25.8	–
Ours (R50)	**26.5**	**27.0**
Ours Multiscale (R50)	**27.3**	**27.7**
SlowFast baseline (R101)	26.3	26.7
LFB (R101) [61]	26.8	–
Ours (R101)	**28.3**	**28.8**
LFB Multiscale (R101) [61]	27.7	–
Ours Multiscale (R101)	**29.5**	**30.0**

UCF101-24

Method	Modality	Mean AP
ACT [4]	RGB + Flow	69.5
Song *et al.* [7]	RGB + Flow	72.1
STEP [10]	RGB + Flow	75.0
Gu *et al.* [3]	RGB + Flow	76.3
MOC [5]	RGB + Flow	78.0
SlowFast (R50)	RGB	76.6
SlowFast (R101)	RGB	77.4
Ours (R50)	RGB	78.6
Ours (R101)	RGB	**79.3**

Conclusion

- Model spatio-temporal relationships between actors, objects and environment explicitly in video
- Previous structured models are a special case
- Achieve state-of-the-art results on 3 datasets.

Conclusion

- Model spatio-temporal relationships between actors, objects and environment explicitly in video
- Previous structured models are a special case
- Achieve state-of-the-art results on 3 datasets.

- *Instead of having a module for relational reasoning at the end of the network, can we perform this interaction-modelling throughout the entire network architecture?*

ViViT: A Video Vision Transformer

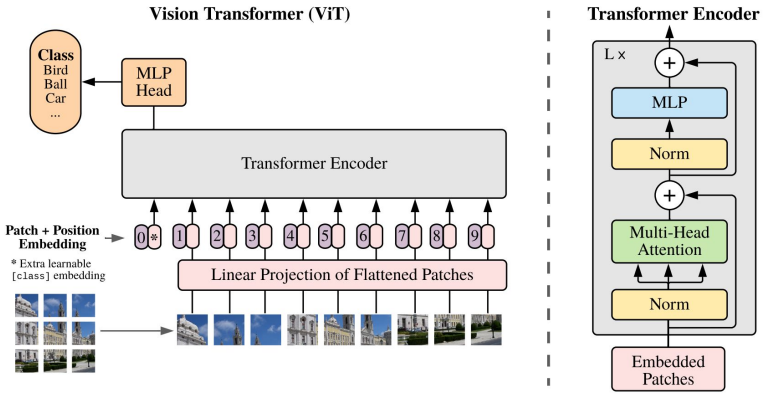
Google Research

Introduction

- CNNs are the architecture of choice in Vision
- Transformers are the architecture of choice in NLP
- Numerous attempts to incorporate self-attention into CNNs:
  - Wang CVPR 2018, Bello ICCV 2019, Huang ICCV 2019, Carion ECCV 2020, Arnab arXiv 2021
- Or to replace convolutions entirely with self-attention
  - Parmar ICML 2018, Ramachandran NeurIPS 2019

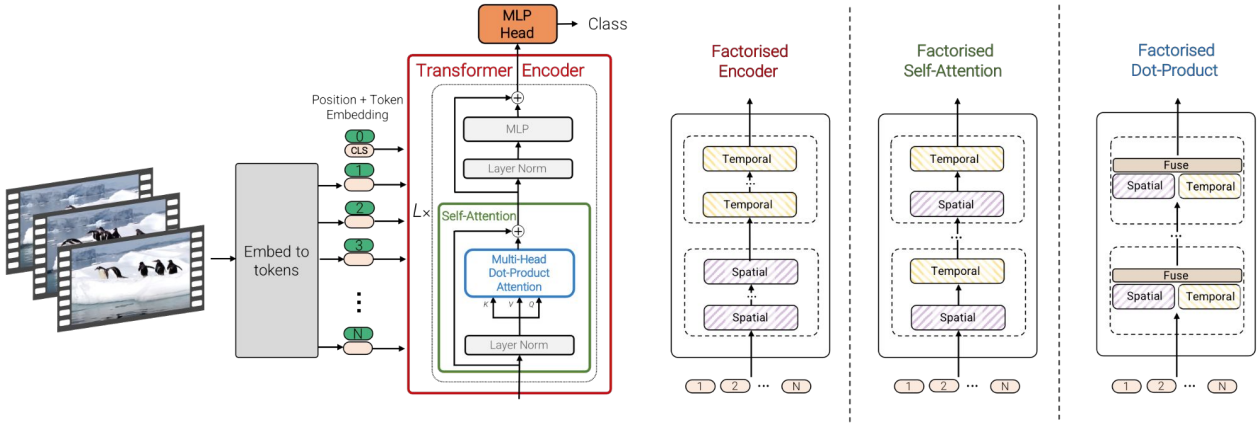
Vision Transformers (ViT) [1]

- Pure-transformer architecture for image classification
- Architecture almost identical to original transformer Vaswani *et al*.
- Such architectures are only very effective at large scale
  - Large datasets (i.e. ImageNet 21K, JFT) required.



ViViT: Video Vision Transformers

- Extend idea of ViT (static images) to videos
- To handle large number of tokens, explore more efficient factorised attention variants.
- Regularisation to train on comparatively small video datasets.



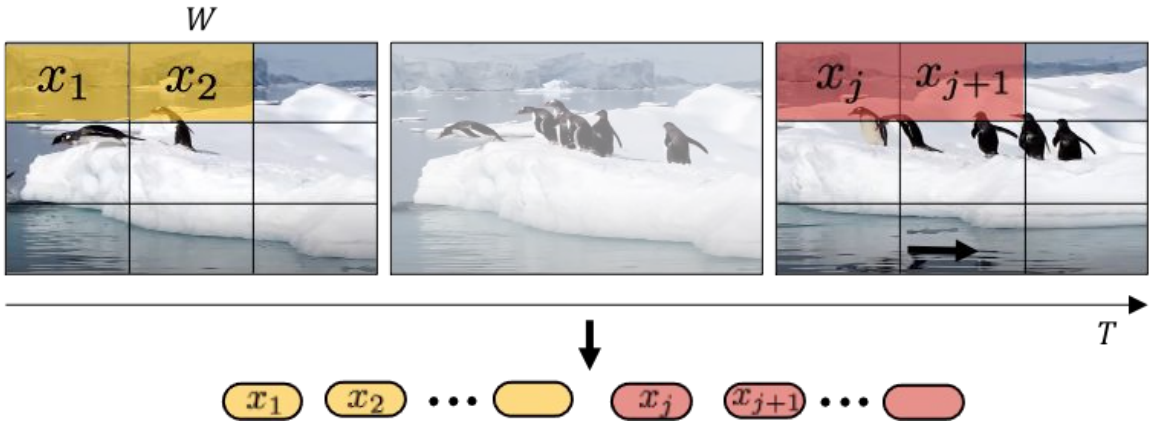
Google Research

Input encoding

- Transformer is a generic architecture that operates on a sequence of tokens.

Input encoding: Uniform Frame Sampling

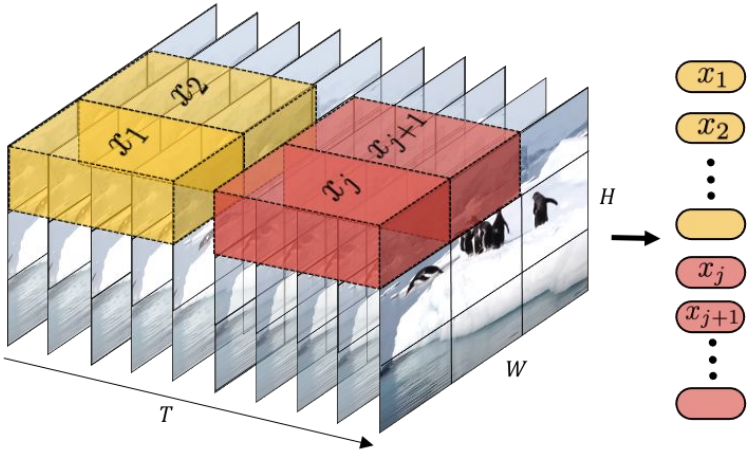
- Sample frames, extract 2D patches and linearly project (as in ViT)
- Effectively consider a video as a “big image”



Google Research

Input encoding: Tubelet embedding

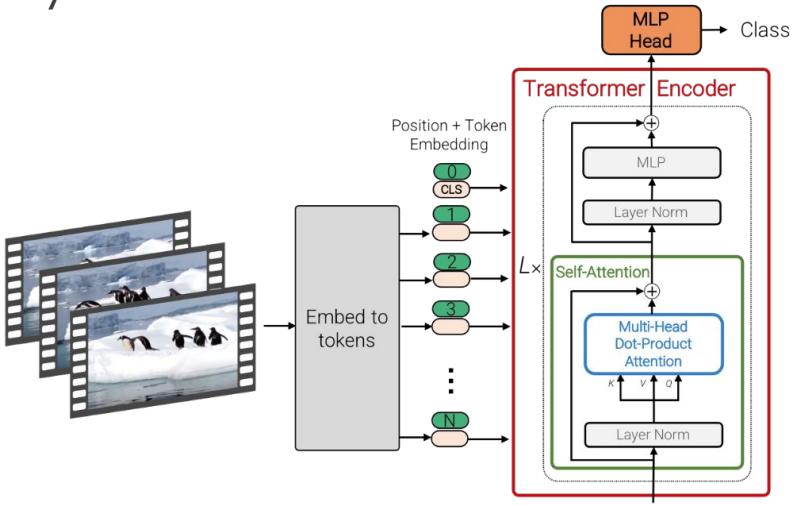
- Extract 3D tubelets to encode spatio-temporal “tubes” into tokens
- Temporal information included from the initial tokenisation stage.
- Works better when initialised appropriately.



Google Research

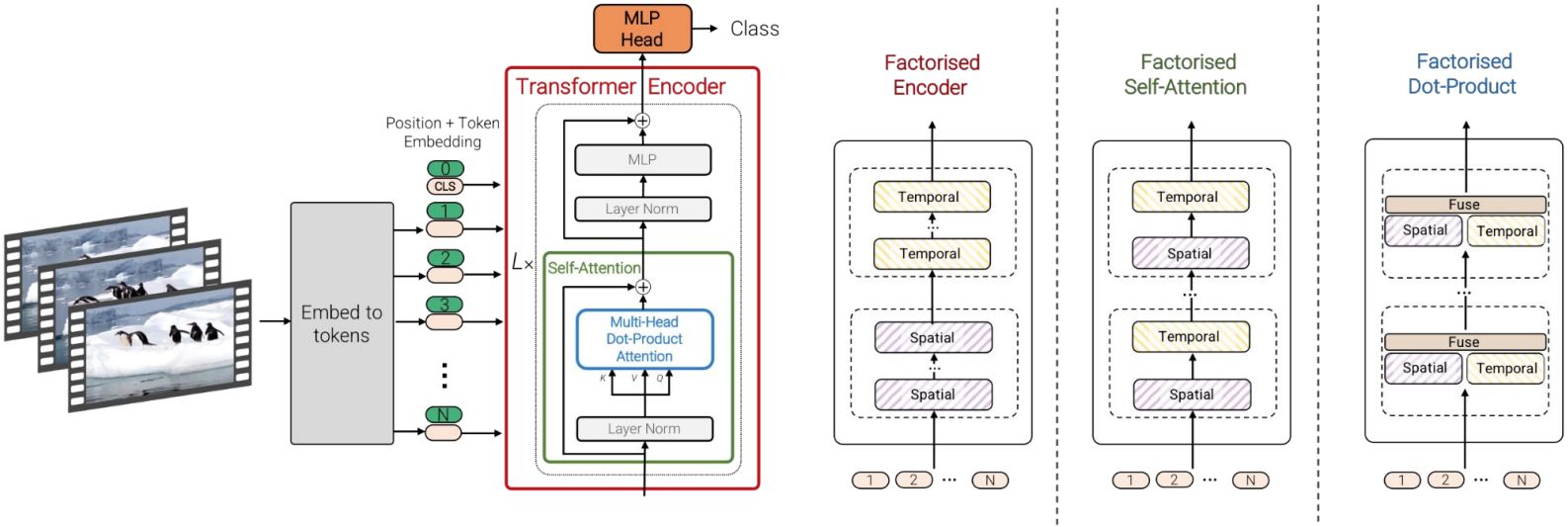
ViViT: Joint Spatio-Temporal Attention

- Simply forward many spatio-temporal tokens through multiple transformer layers.
- Requires a lot of computation, and high-capacity means it can overfit easily on smaller datasets.



Google Research

ViViT: Space/Time Factorisations

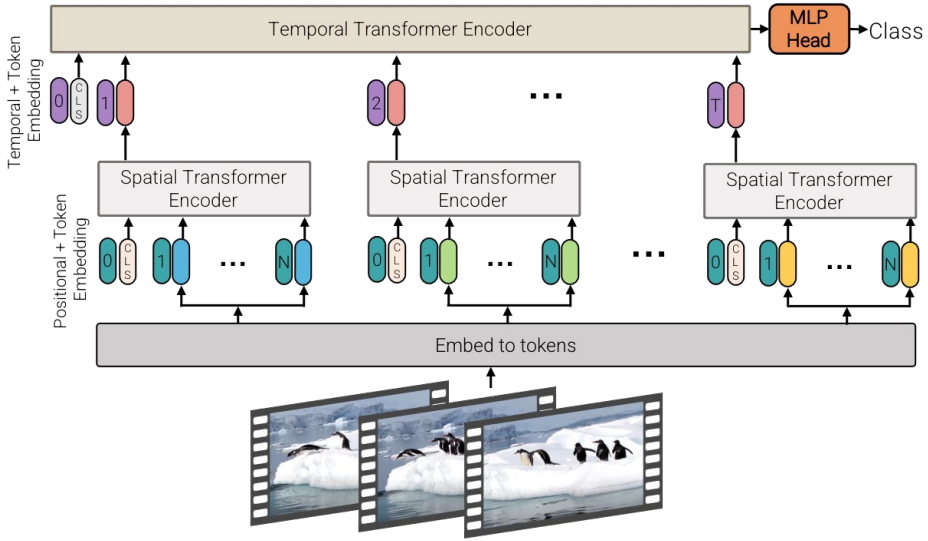


Alternative ways of mixing the temporal and spatial information

Reduces complexity from $O((w * h)^2 + t^2)$ instead of $O((w*h)^2 * t^2)$

ViViT Factorised Encoder (“Late fusion”)

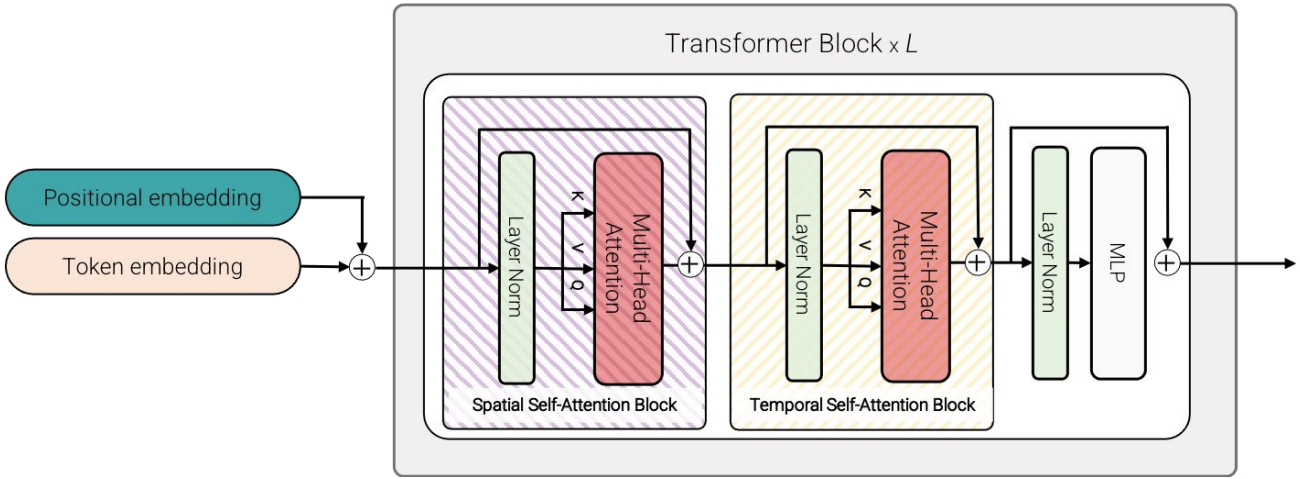
- Separate encoders for encoding spatial and temporal information
  - Spatial encoder is initialised from a pretrained-ViT model
  - “*Late fusion*” of spatial and temporal information
  - Has more parameters compared to ViViT, but less compute compared to ViViT



Google Research

ViViT Factorised Self-Attention

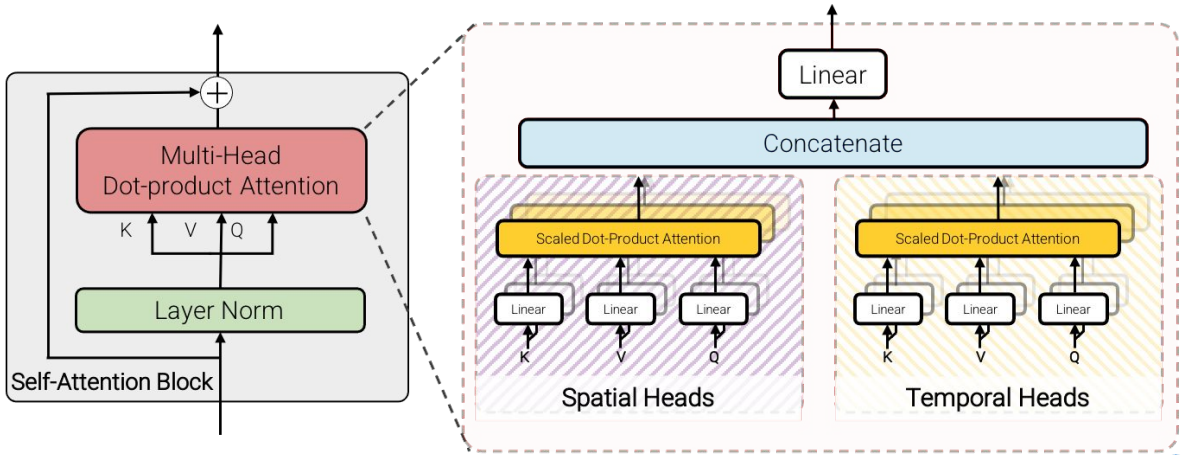
- Two self-attention blocks for running attention on spatial and temporal axes respectively
  - More parameters compared to ViViT
  - Less compute compared to ViViT



Google Research

ViViT Factorised Dot-Product-Attention

- Split heads into two sets and compute dot-product attention independently, using separate spatial- and temporal-heads.
  - Same number of parameters compared to ViViT
  - Less compute compared to ViViT



Google Research

Input Encoding

- Tubelet embedding works better if 3D filter is initialised appropriately.
  - Initialise to “select” central frame using 2D filter weights.

	Top-1 accuracy
Uniform frame sampling	78.5
Tubelet embedding	
Random initialisation [22]	73.2
Filter inflation [6]	77.6
Central frame	79.2

Google Research

Experimental Evaluation

Consider five standard classification datasets:

- Kinetics 400
- Kinetics 600
- Moments in Time
- Epic Kitchens
- Something-Something v2

Google Research

Input Encoding

- Tubelet embedding works better if 3D filter is initialised appropriately.
  - Filter inflation [1, 2]: $\mathbf{E} = \frac{1}{t}[\mathbf{E}_{\text{image}}, \ldots, \mathbf{E}_{\text{image}}, \ldots, \mathbf{E}_{\text{image}}]$.
  - Central frame initialiser: $\mathbf{E} = [\mathbf{0}, \ldots, \mathbf{E}_{\text{image}}, \ldots, \mathbf{0}]$.
    - Initialise to “select” central frame using 2D filter weights.

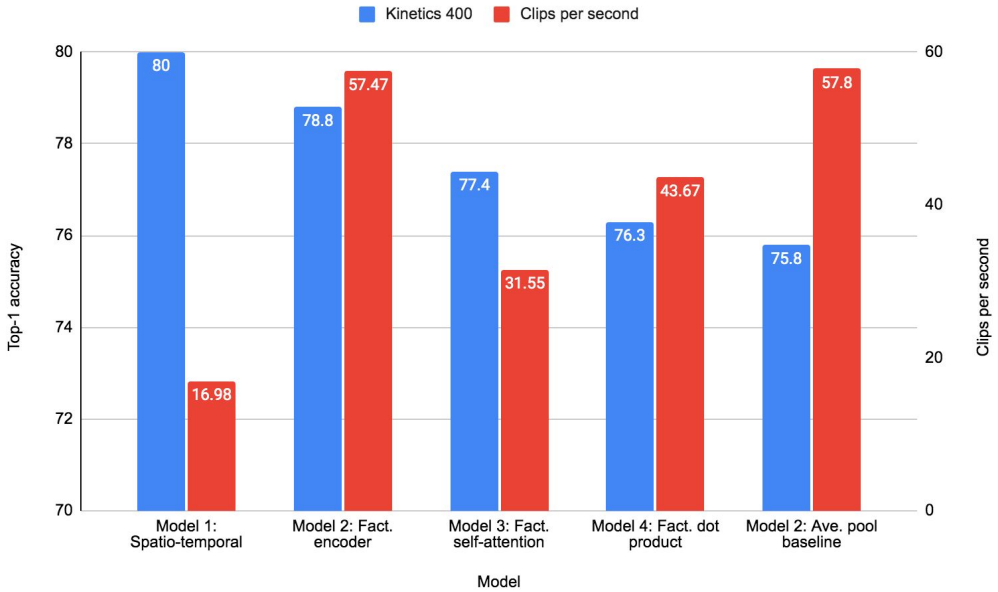
	Top-1 accuracy
Uniform frame sampling	78.5
Tubelet embedding	
Random initialisation [22]	73.2
Filter inflation [6]	77.6
Central frame	79.2

[1] Carreira and Zisserman. CVPR 2017.
[2] Feichtenhofer et al. NeurIPS 2016

Google Research

Model Variants

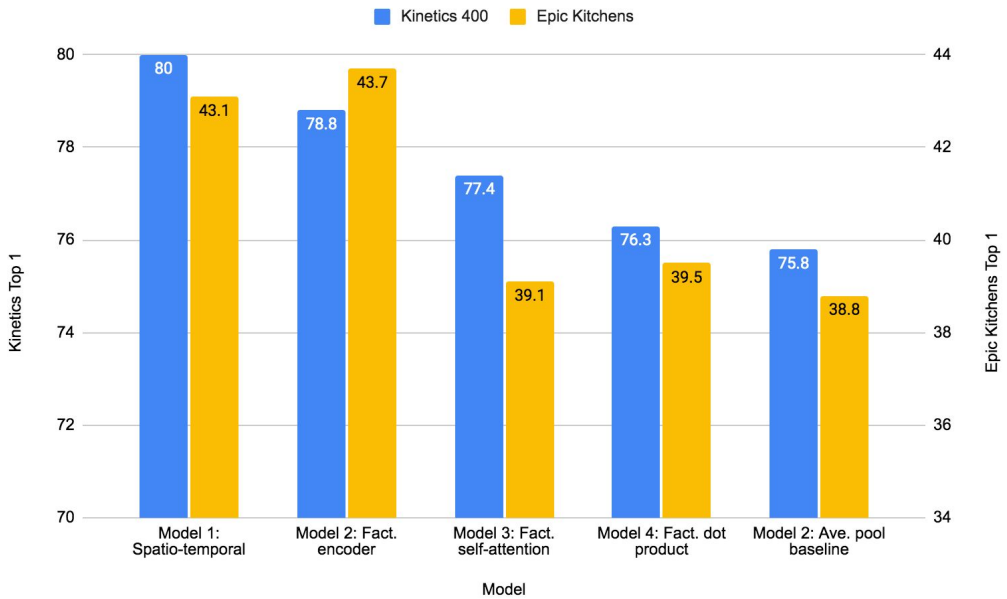
- Tokens fixed across models
- Unfactorised model works best on larger datasets (ie Kinetics), but slowest.



Google Research

Model Variants

- Factorised encoder works best on smaller datasets (ie Epic Kitchens) as it overfits less.



Google Research

Model Variants

- Unfactorised model works best on larger datasets (ie Kinetics)
- Factorised encoder works best on smaller datasets (ie Epic Kitchens) as it overfits less.

	K400	EK	FLOPs ($\times 10^9$)	Params ($\times 10^6$)	Runtime (ms)
Model 1: Spatio-temporal	80.0	43.1	455.2	88.9	58.9
Model 2: Fact. encoder	78.8	43.7	284.4	100.7	17.4
Model 3: Fact. self-attention	77.4	39.1	372.3	117.3	31.7
Model 4: Fact. dot product	76.3	39.5	277.1	88.9	22.9
Model 2: Ave. pool baseline	75.8	38.8	283.9	86.7	17.3

Google Research

Regularisation

- Video datasets are not as large as ImageNet / ImageNet21k / JFT
  - Original ViT paper didn't get good performance on ImageNet.
- Strategies
  - Use pretrained image models from ImageNet-21K or JFT
  - For smaller datasets, we use further regularisation methods, inspired by DeiT [1]

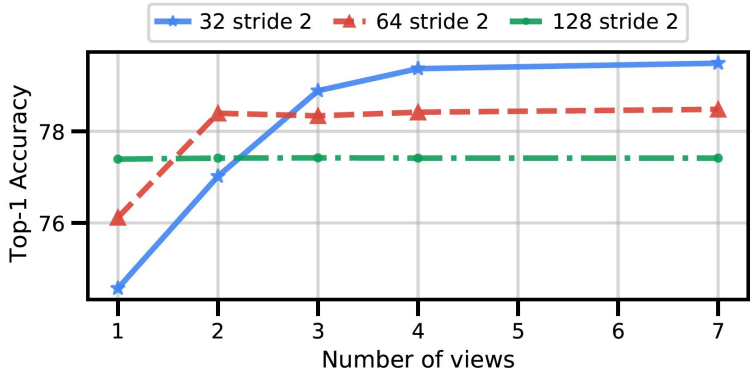
	Top-1 accuracy
Random crop, flip, colour jitter	38.4
+ Kinetics 400 initialisation	39.6
+ Stochastic depth [28]	40.2
+ Random augment [10]	41.1
+ Label smoothing [58]	43.1
+ Mixup [79]	43.7

5.3% gain on Epic Kitchens

Google Research

Processing long videos

- Common 3D CNN architectures process 32 frames at a time [1, 2]
  - At test time, average results over multiple “views” of the same video.
- We can process longer videos, but keep compute essentially constant, by increasing tubelet length and keeping the number of temporal tokens constant.
- Not useful for current datasets, but avenue for future work.
  - “Multi-view” evaluation achieves higher accuracy accurate.

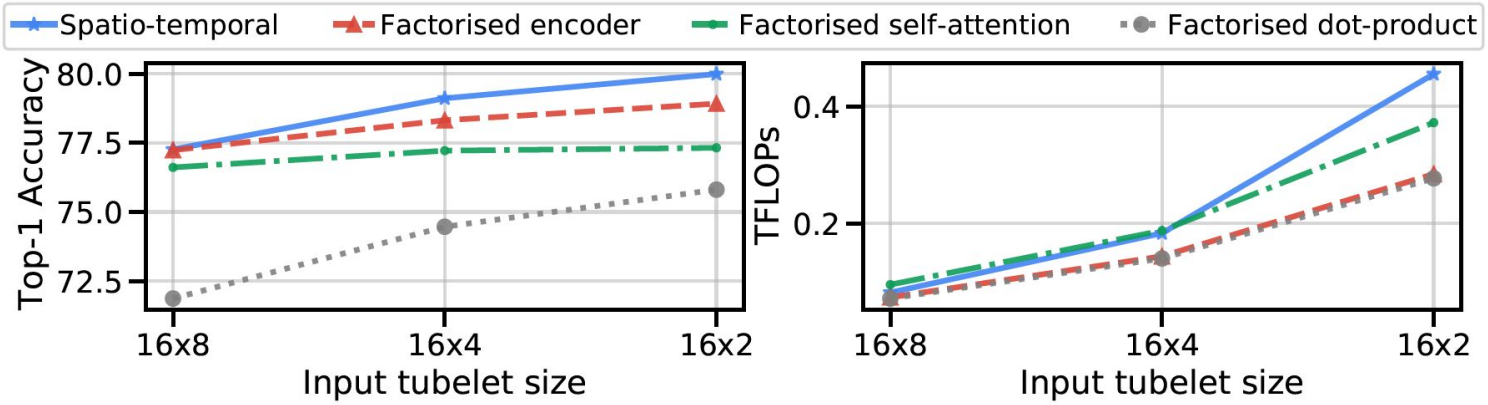


Kinetics video has 250 frames. Accuracy saturates once network has “seen” the whole video.

Google Research

More tokens help

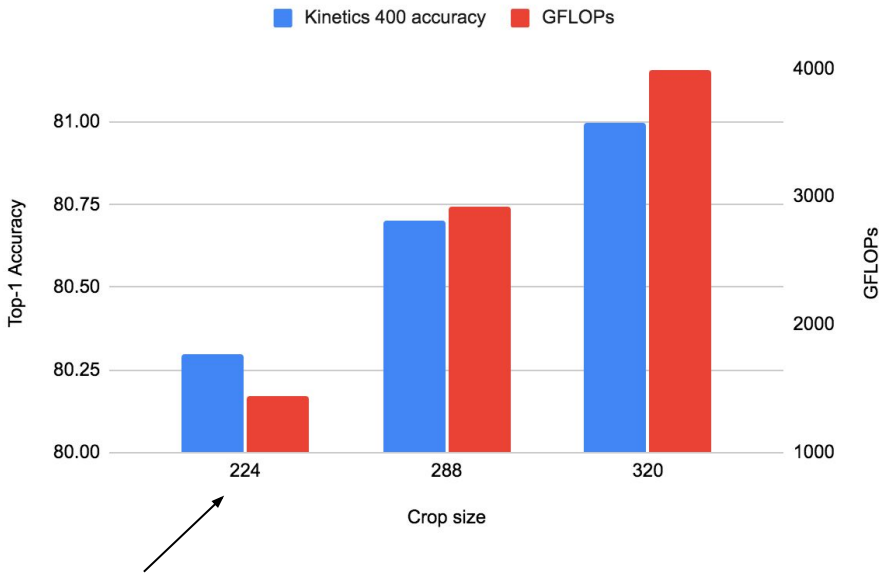
- More temporal tokens help.
- More spatial tokens help too.
- Both increase computation.



Google Research

More tokens help

- More temporal tokens help.
- More spatial tokens help too.
- Both increase computation.



Enough to get SOTA results

Google Research

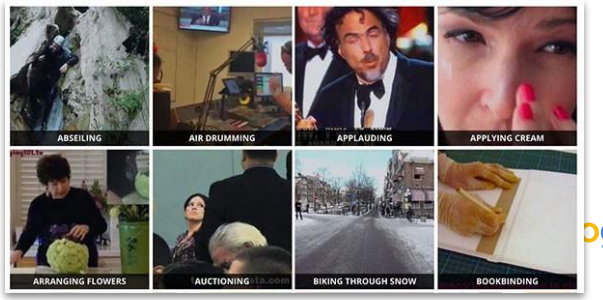
Kinetics 400 and Kinetics 600

(a) Kinetics 400

Method	Top 1	Top 5	Views
blVNet [16]	73.5	91.2	–
STM [30]	73.7	91.6	–
TEA [39]	76.1	92.5	10 × 3
TSM-ResNeXt-101 [40]	76.3	–	–
I3D NL [72]	77.7	93.3	10 × 3
CorrNet-101 [67]	79.2	–	10 × 3
ip-CSN-152 [63]	79.2	93.8	10 × 3
LGD-3D R101 [48]	79.4	94.4	–
SlowFast R101-NL [18]	79.8	93.9	10 × 3
X3D-XXL [17]	80.4	94.6	10 × 3
TimeSformer-L [2]	80.7	94.7	1 × 3
ViViT-L/16x2	80.6	94.7	4 × 3
ViViT-L/16x2 320	**81.3**	**94.7**	4 × 3
Methods with large-scale pretraining			
ip-CSN-152 [63] (IG [41])	82.5	95.3	10 × 3
ViViT-L/16x2 (JFT)	82.8	95.5	4 × 3
ViViT-L/16x2 320 (JFT)	83.5	95.5	4 × 3
ViViT-H/16x2 (JFT)	**84.8**	**95.8**	4 × 3

(b) Kinetics 600

Method	Top 1	Top 5	Views
AttentionNAS [73]	79.8	94.4	–
LGD-3D R101 [48]	81.5	95.6	–
SlowFast R101-NL [18]	81.8	95.1	10 × 3
X3D-XL [17]	81.9	95.5	10 × 3
TimeSformer-HR [2]	82.4	**96.0**	–
ViViT-L/16x2	82.5	95.6	4 × 3
ViViT-L/16x2 320	**83.0**	95.7	4 × 3
ViViT-L/16x2 (JFT)	84.3	96.2	4 × 3
ViViT-H/16x2 (JFT)	**85.8**	**96.5**	4 × 3



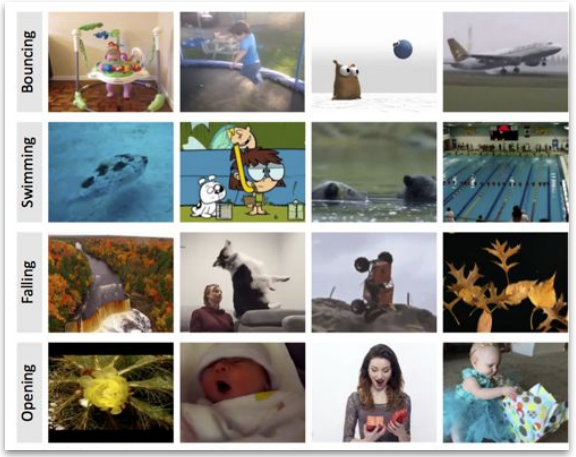
ogle Research

Moments in Time

- Similar to Kinetics in consisting of YouTube videos.
- Significant label noise, which is why accuracies are lower

(c) Moments in Time

	Top 1	Top 5
TSN [69]	25.3	50.1
TRN [83]	28.3	53.4
I3D [6]	29.5	56.1
blVNet [16]	31.4	59.3
AssembleNet-101 [51]	34.3	62.7
ViViT-L/16x2	**38.0**	**64.9**



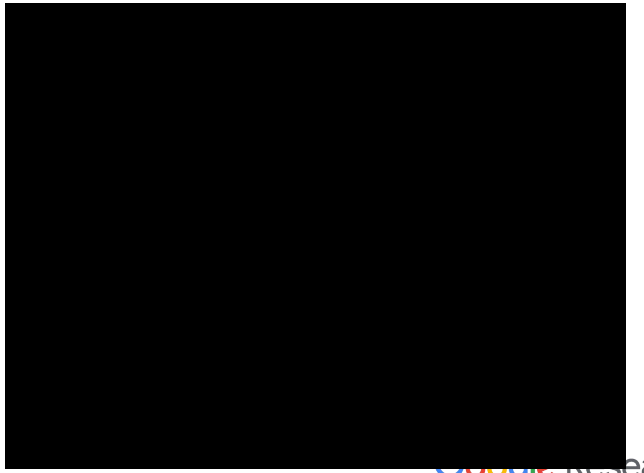
Google Research

Epic Kitchens

- Verb = what person is doing. Noun = what they are interacting with.
- Action = combination of two. Main metric.
- Other method getting higher verb accuracy are using optical flow.

(d) Epic Kitchens 100 Top 1 accuracy

Method	Action	Verb	Noun
TSN [69]	33.2	60.2	46.0
TRN [83]	35.3	65.9	45.4
TBN [33]	36.7	66.0	47.2
TSM [40]	38.3	**67.9**	49.0
SlowFast [18]	38.5	65.6	50.0
ViViT-L/16x2 Fact. encoder	**44.0**	66.4	**56.8**



Google Research

Something-Something v2

- Margin smaller with respect to other methods
- Requires capturing fine-grained motions, as objects and background are consistent across classes
- Large gap to TimeSformer, as they did not regularise as much. Or use tubelet embedding

(e) Something-Something v2

Method	Top 1	Top 5
TRN [83]	48.8	77.6
SlowFast [17, 77]	61.7	–
TimeSformer-HR [2]	62.5	–
TSM [40]	63.4	88.5
STM [30]	64.2	89.8
TEA [39]	65.1	–
blVNet [16]	65.2	**90.3**
ViViT-L/16x2 Fact. encoder	**65.4**	89.8

Something-Something v2

- Margin smaller with respect to other methods
- Requires capturing fine-grained motions, as objects and background are consistent across classes
- Large gap to TimeSformer, as they did not regularise as much. Or use tubelet embedding

(e) Something-Something v2

Method	Top 1	Top 5
TRN [83]	48.8	77.6
SlowFast [17, 77]	61.7	–
TimeSformer-HR [2]	62.5	–
TSM [40]	63.4	88.5
STM [30]	64.2	89.8
TEA [39]	65.1	–
blVNet [16]	65.2	**90.3**
ViViT-L/16x2 Fact. encoder	**65.4**	89.8

(a) Kinetics 400

Method	Top 1	Top 5	Views
blVNet [16]	73.5	91.2	–
STM [30]	73.7	91.6	–
TEA [39]	76.1	92.5	10 × 3
TSM-ResNeXt-101 [40]	76.3	–	–
I3D NL [72]	77.7	93.3	10 × 3
CorrNet-101 [67]	79.2	–	10 × 3
ip-CSN-152 [63]	79.2	93.8	10 × 3
LGD-3D R101 [48]	79.4	94.4	–
SlowFast R101-NL [18]	79.8	93.9	10 × 3
X3D-XXL [17]	80.4	94.6	10 × 3
TimeSformer-L [2]	80.7	94.7	1 × 3
ViViT-L/16x2	80.6	94.7	4 × 3
ViViT-L/16x2 320	**81.3**	**94.7**	4 × 3
Methods with large-scale pretraining			
ip-CSN-152 [63] (IG [41])	82.5	95.3	10 × 3
ViViT-L/16x2 (JFT)	82.8	95.5	4 × 3
ViViT-L/16x2 320 (JFT)	83.5	95.5	4 × 3
ViViT-H/16x2 (JFT)	**84.8**	**95.8**	4 × 3

Conclusion

- Family of pure-transformer architectures for video
- Showed how to regularise models appropriately to train on smaller datasets
- State-of-the-art results on 5 video datasets

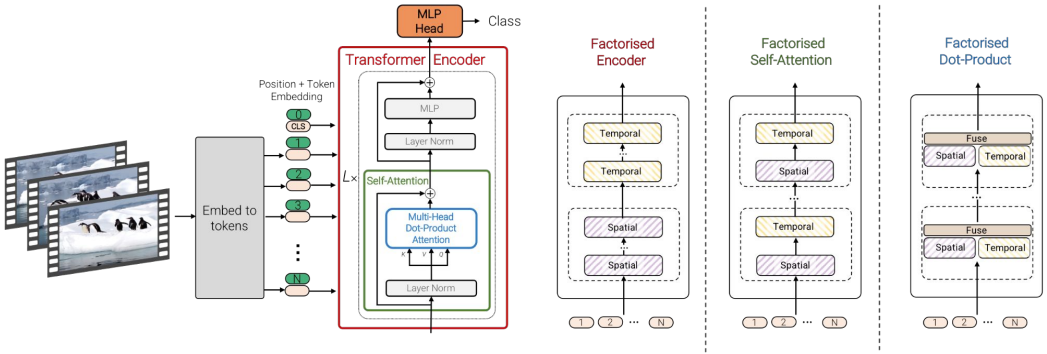
Google Research

Next steps

- Self-supervised pretraining
  - Not rely on ImageNet supervision
- Even more efficient transformer models
- Multimodal
  - Vision, audio, text ...
- Addressing more complex tasks beyond classification

Questions?

- A Arnab, C Sun, C Schmid. Unified Graph Structured Models for Video Understanding. *arXiv:2103.15662*
- A Arnab, M Dehghani, G Heigold, C Sun, M Lucic, C Schmid. ViViT: A Video Vision Transformer. *arXiv:2103.15691*

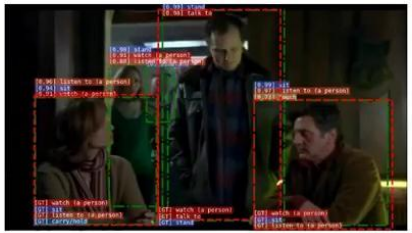


Google Research

Questions?

Input video

Predicted keyframes



Visualised Actor

Spatial messages received

Google Research