Large-Scale Video Understanding with Transformers

Anurag Arnab

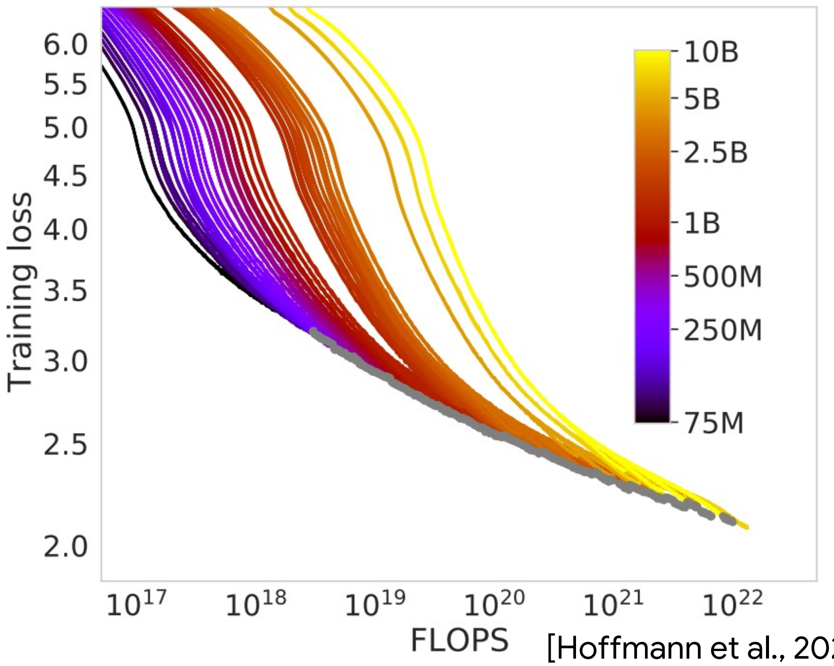
Google Research

Introduction

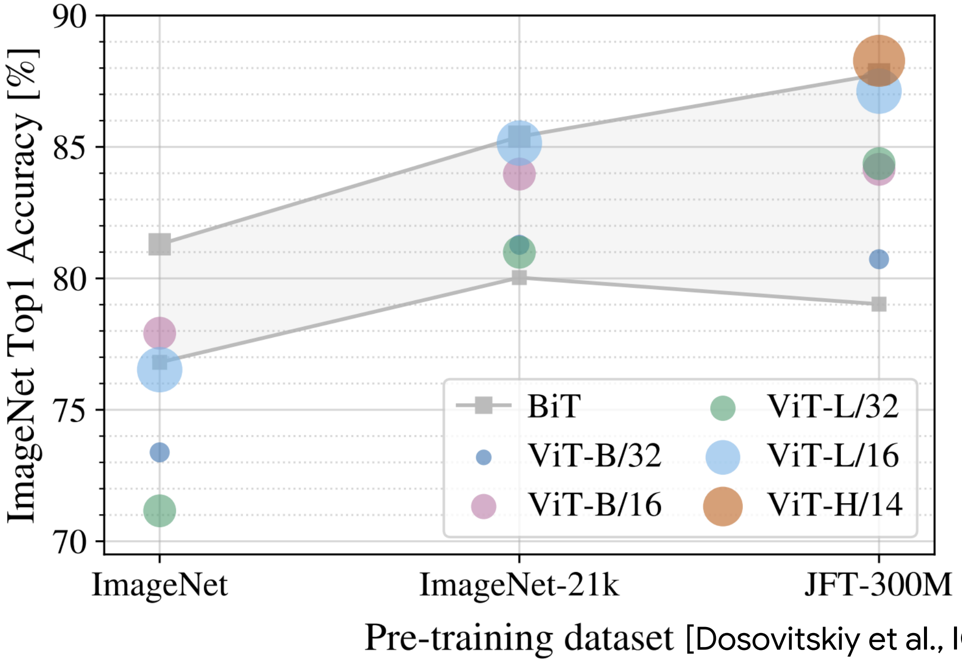
- Transformers achieve state-of-the-art performance in a wide range of domains.
- And that motivates us to develop transformer-based models for video understanding.

Transformers

- Scale with larger datasets, in a manner that convolutional networks cannot.
- Can naturally handle any input which can be "tokenized"



[Hoffmann et al., 2022]



[Dosovitskiy et al., ICLR 2021]

Transformers for video – Questions

1. How to develop transformer models for video?
2. Transformers have quadratic complexity with respect to the number of tokens
   - How do we make them more efficient for video?
3. Videos are inherently multimodal
   - How do we effectively leverage this information?
4. Transformers work well across a large range of domains
   - Can we train a single transformer model for everything?

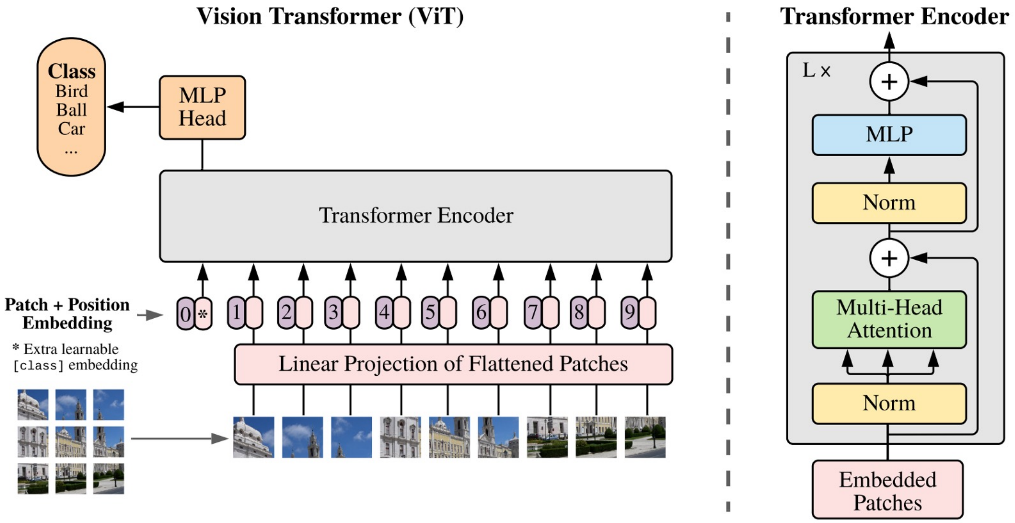
ViViT: A Video Vision Transformer

Anurag Arnab, Mostafa Dehghani,
Georg Heigold, Chen Sun,
Mario Lucic, Cordelia Schmid

Google Research

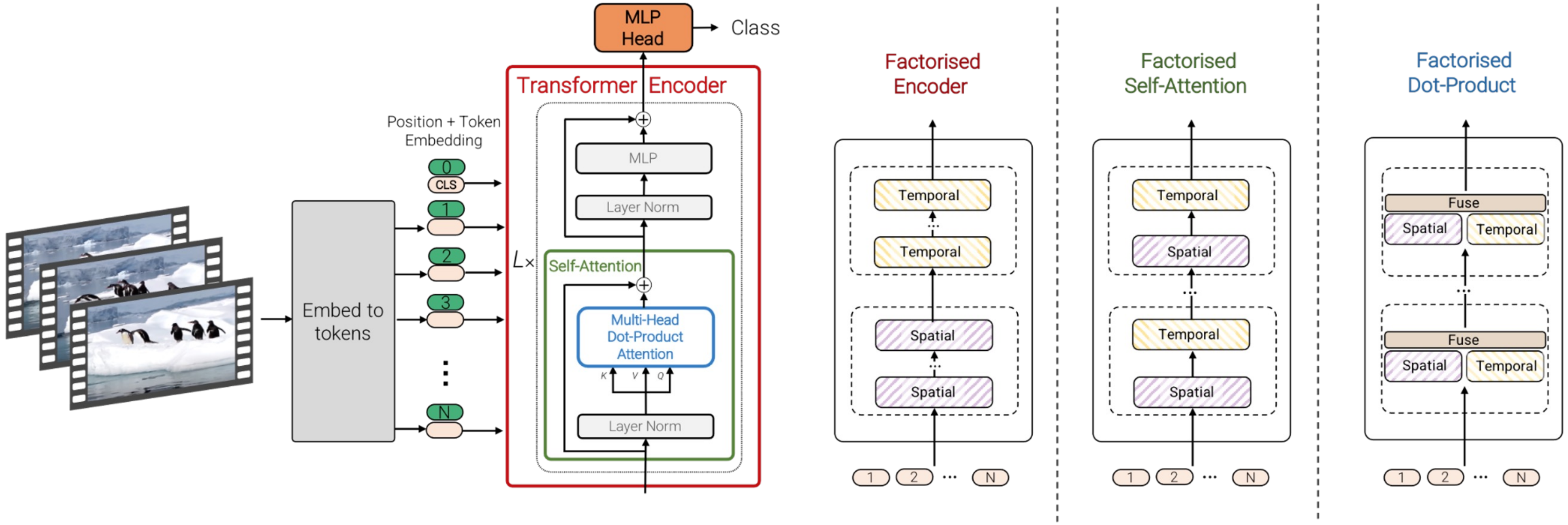
Introduction

- CNNs are architecture of choice in Vision ; Transformers are architecture of choice in Natural Language
- [Vision Transformers](): recent pure-transformer architecture for images
- Benefits of such architectures realised at large scale



ViViT: Video Vision Transformers

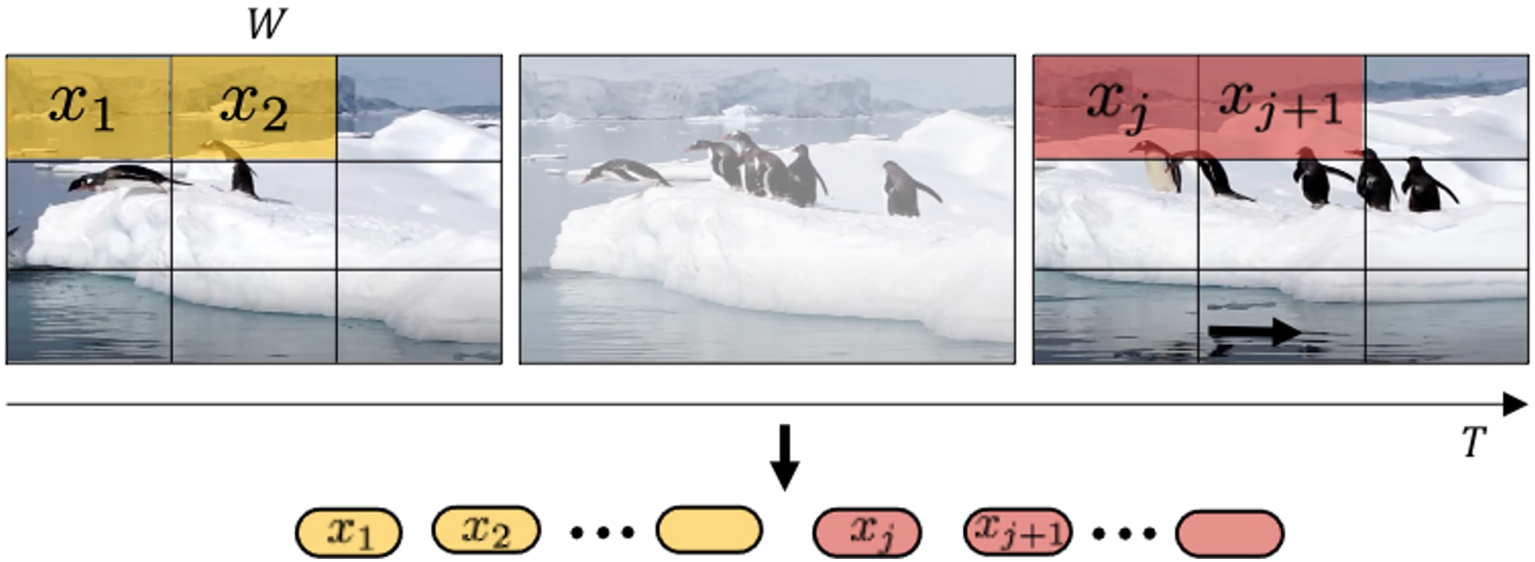
- Extend idea of ViT (static images) to videos
- To handle large number of tokens, explore more efficient factorised attention variants.
- Regularisation to train on comparatively small video datasets.



Google Research

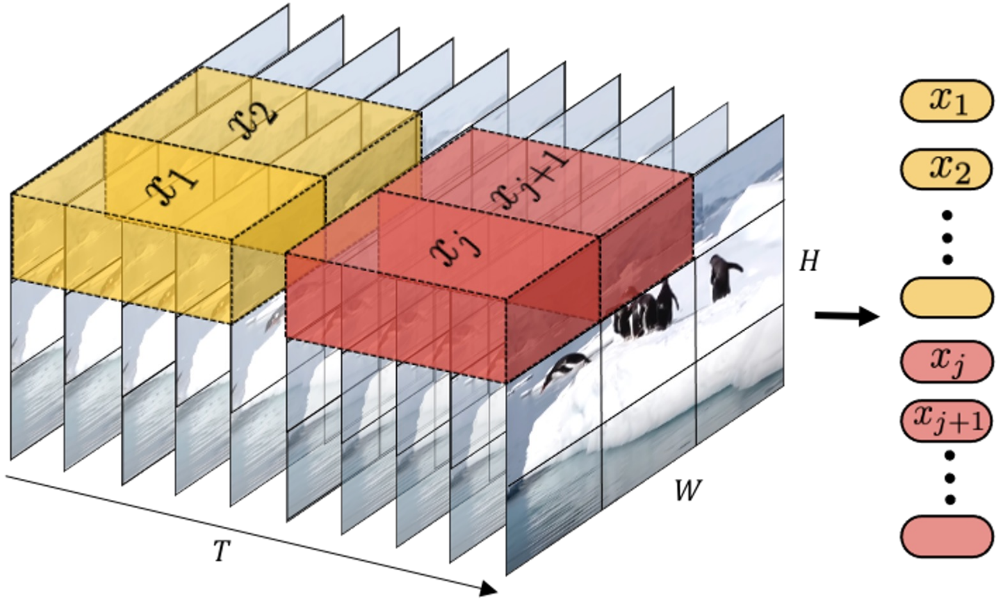
Input Encoding 1: Uniform Frame Sampling

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



Input Encoding 2: Tubelet embedding

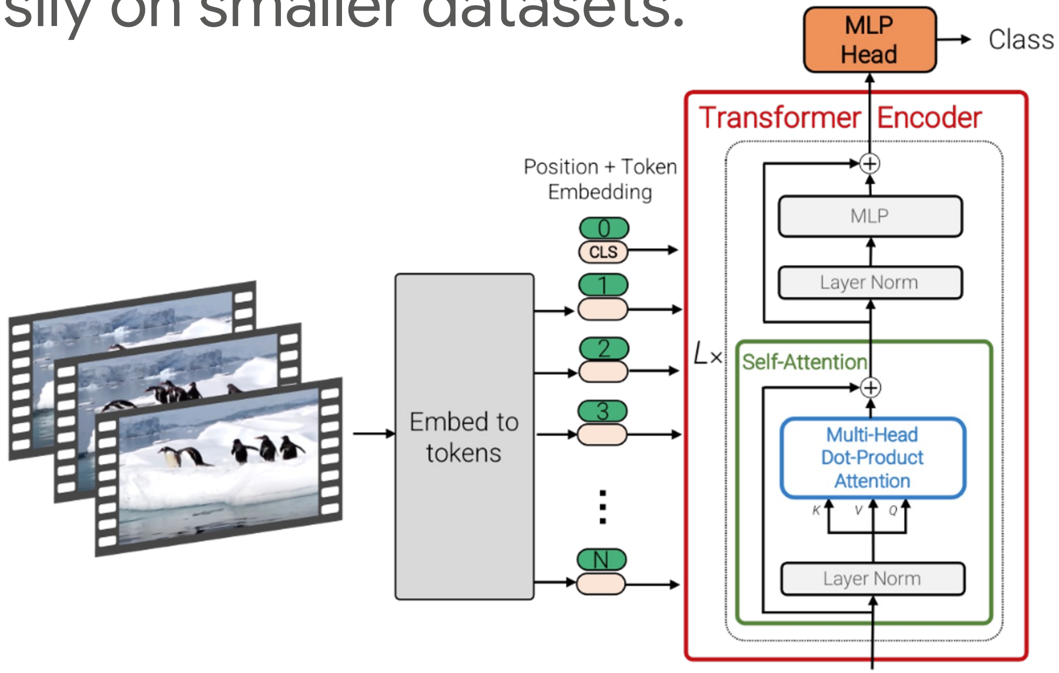
- Extract 3D tubelets to encode spatio-temporal “tubes” into tokens
- Temporal information included from the initial tokenisation stage.
- Works better when initialised appropriately.



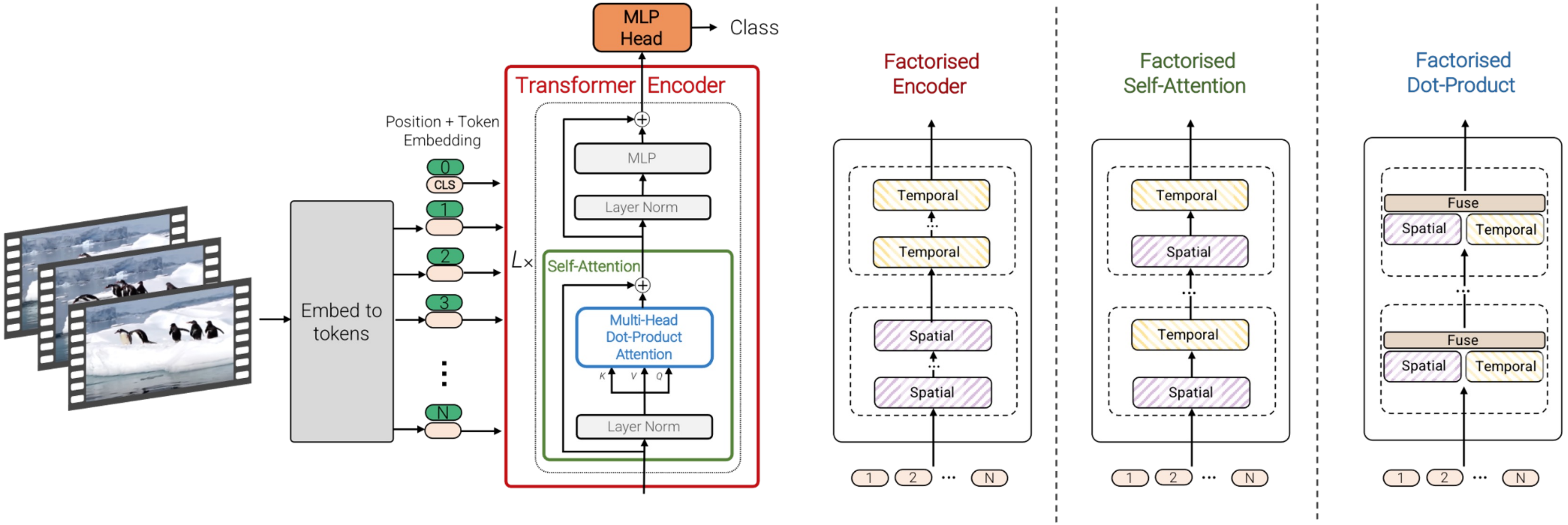
Google Research

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.



ViViT: Space/Time Factorisations



Alternative ways of mixing the temporal and spatial information

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

ViViT Factorisations

Factorised encoder

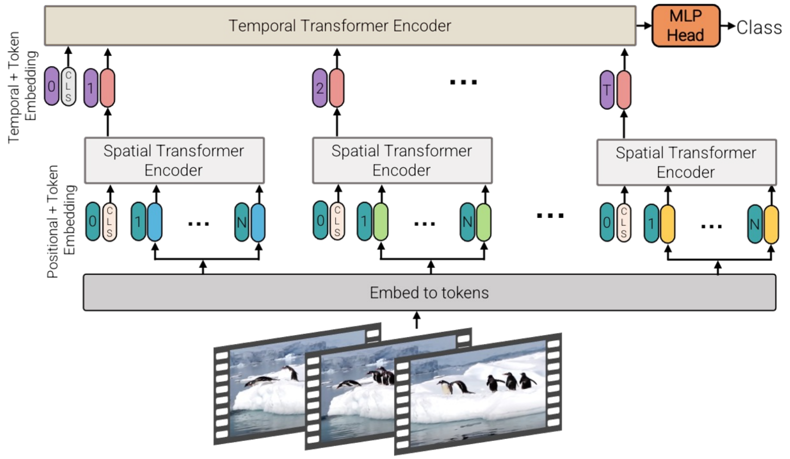
- "Late fusion" of spatial and temporal information

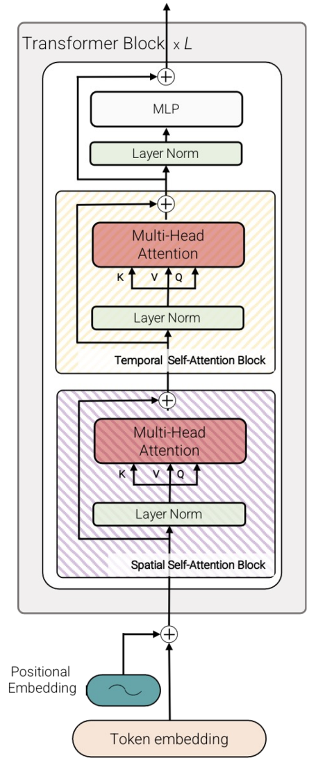
Factorised self-attention

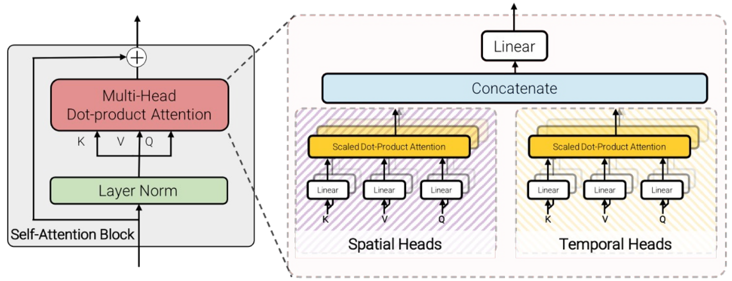
- Perform self-attention separately over space and time

Factorised dot-product

- Attention heads separated over space and time dimensions.







Google Research

Input Encoding

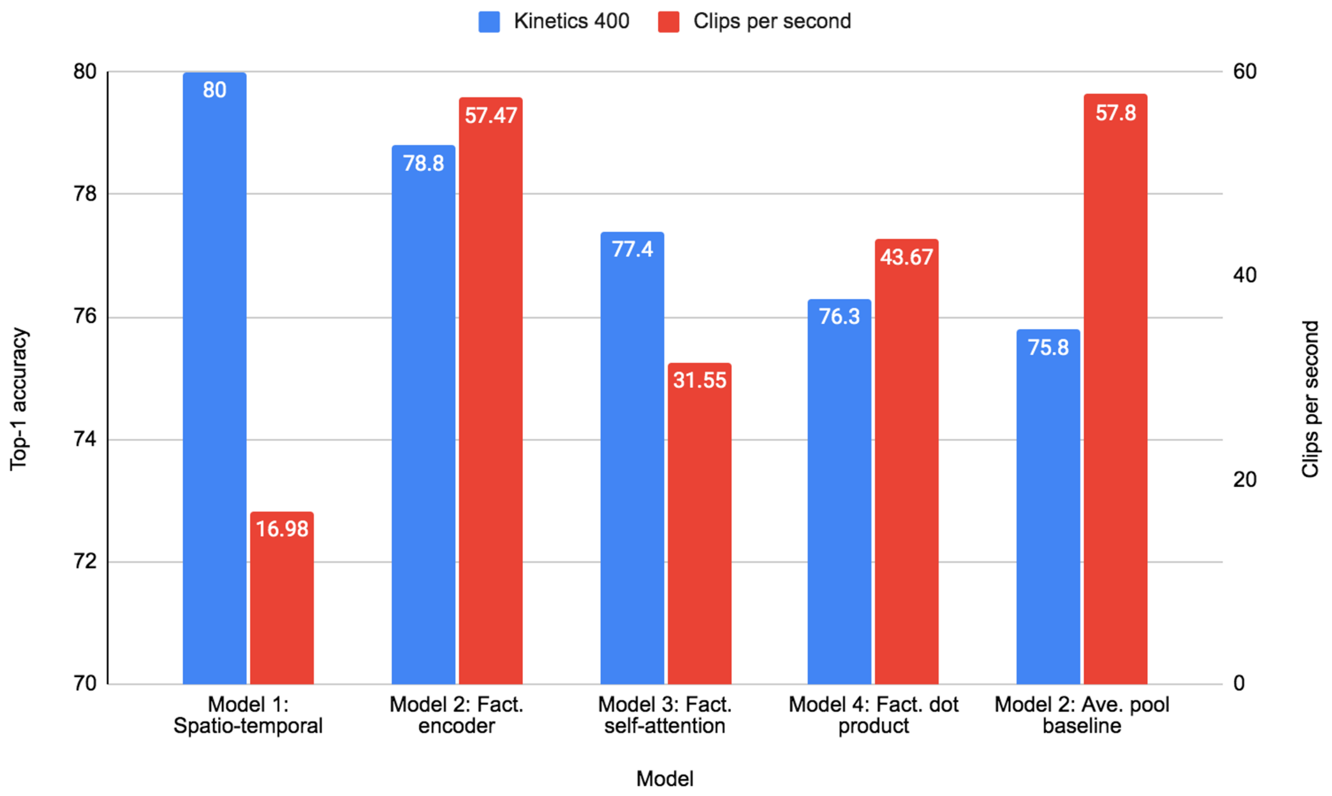
- Tubelet embedding works better if 3D filter is initialised appropriately.
  - Filter inflation [1, 2]: $\mathbf{E} = \frac{1}{t}[\mathbf{E}_{\text{image}}, \ldots, \mathbf{E}_{\text{image}}, \ldots, \mathbf{E}_{\text{image}}].$
  - Central frame initialiser: $\mathbf{E} = [\mathbf{0}, \ldots, \mathbf{E}_{\text{image}},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

Model Variants

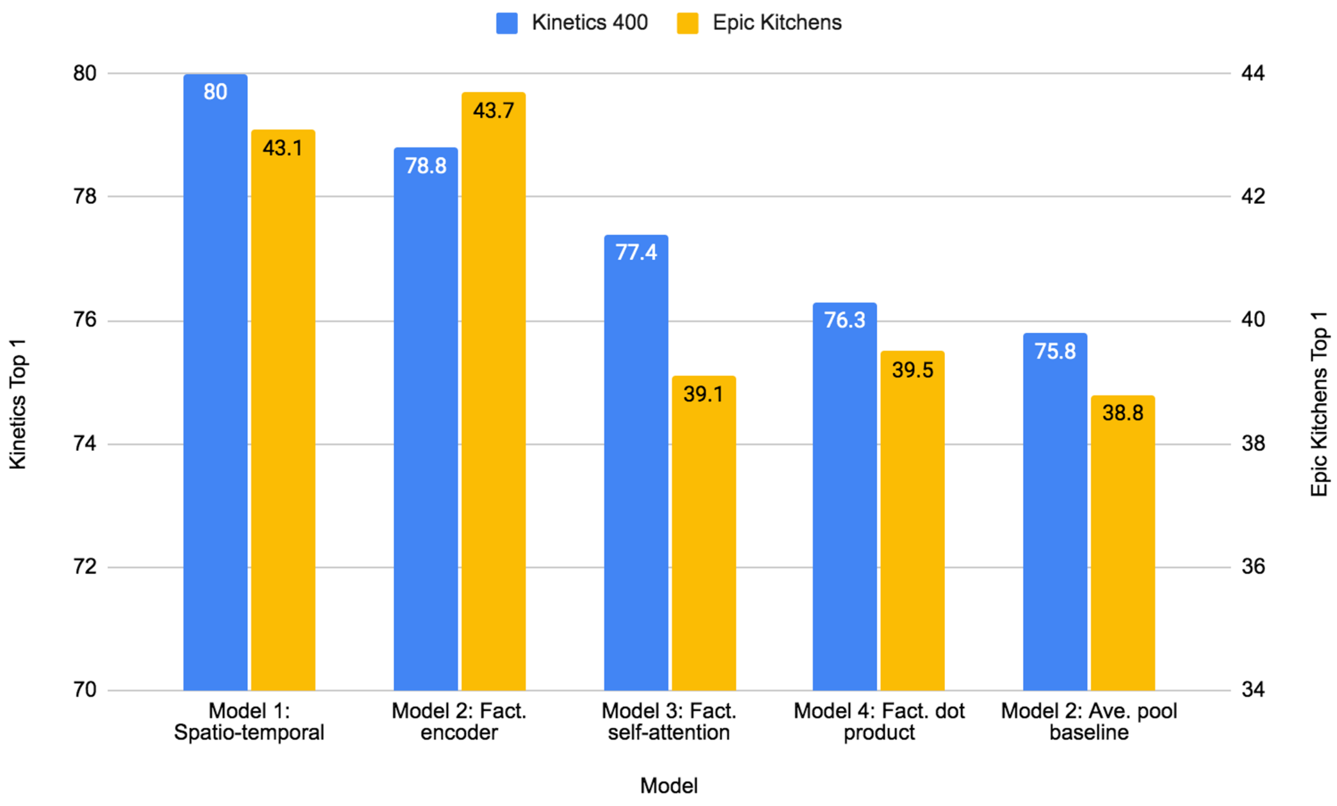
- Tokens fixed across models
- Unfactorised model works best on larger datasets (ie Kinetics), but slowest.



Google Research

Model Variants

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



Google Research

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.

	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

State-of-the-art Results on 5 Datasets

(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

(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**

(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**

(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

Google Research

Conclusion

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

- A Arnab *et al.* ViViT: A Video Vision Transformer. ICCV, 2021.
- [Paper], [Code]

Questions

- How can we model temporal dynamics more effectively?
- How can we make such models more efficient?
- How can we process and fuse multiple modalities?
- Can we train a single transformer model to perform multiple tasks across different modalities?

Questions

- How can we model temporal dynamics, and different modalities, more effectively?
    - Multiview Transformers for Video Recognition
- How can we make such models more efficient?
    - TokenLearner: What Can 8 Learned Tokens do for Images and Videos?
- Can we train a single transformer model to perform multiple tasks across different modalities?
    - PolyViT: Co-training Vision Transformers on Images, Video and Audio

Multiview Transformers for Video Recognition

Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, Cordelia Schmid

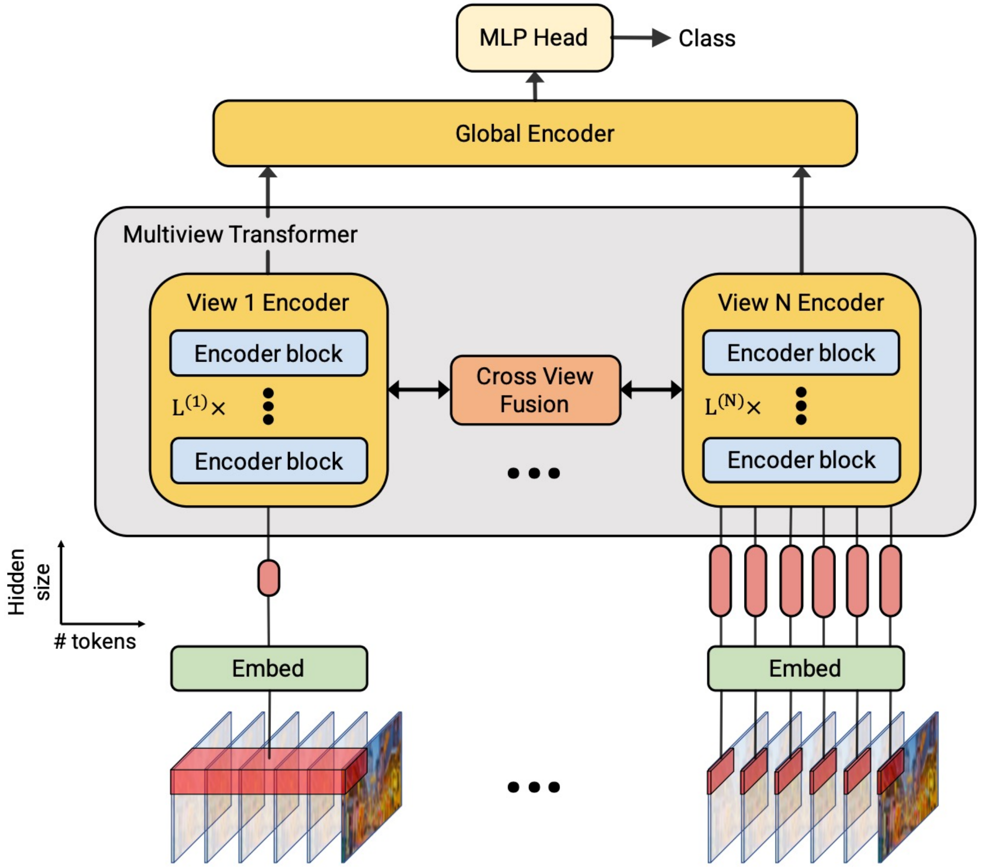
Google Research

Motivation

- Modelling inputs at multiple resolutions has been a central idea in Computer Vision, since handcrafted features (Burt and Adelson 1987, Dalal and Triggs 2005, Lazebnik et al 2006).
  - In space: detect objects of variable sizes
  - In time: detect events of different durations
- How to model multiple spatio-temporal resolutions with transformers?

Multiview Transformers

- Model multiscale, temporal information
- Create different “views” of the input
- Process these views in parallel, with lateral connections between transformer layers.
- Final global encoder aggregates tokens from each view encoder.

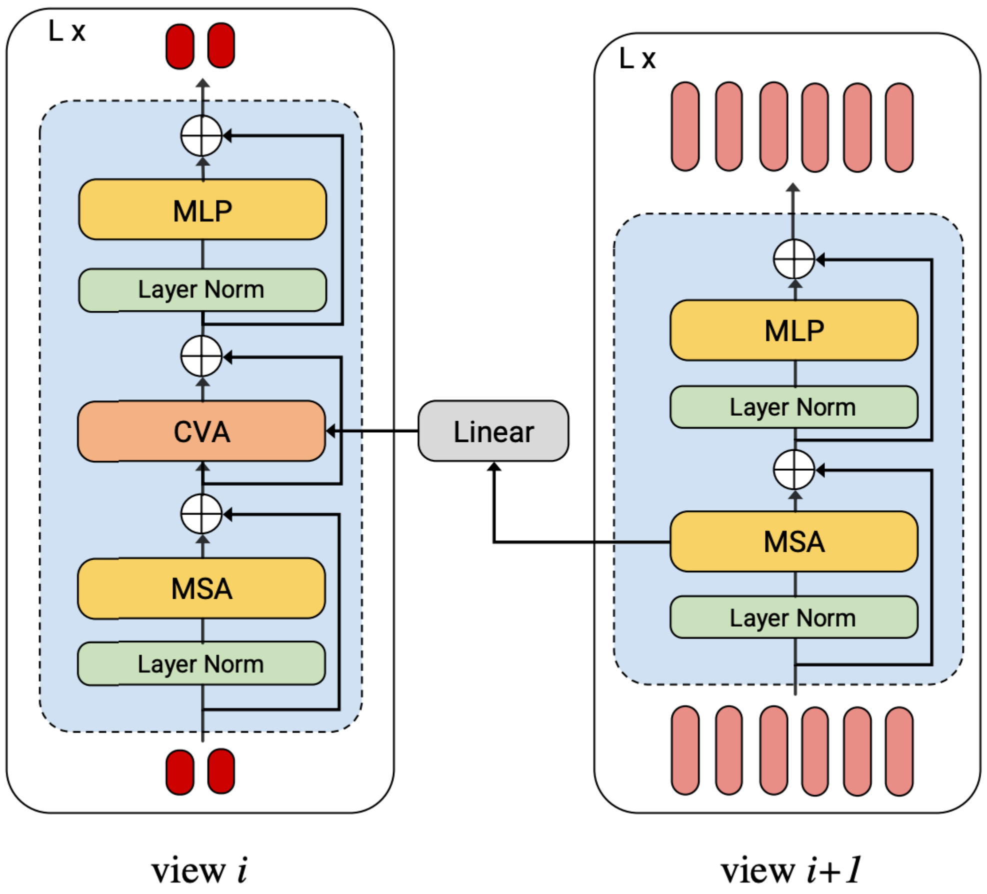


Multiview Transformers

- Our naming convention example
- B/2 + S/4 + Ti/8
  - Three views
  - “Base” transformer with tubelet size of 16x2
  - “Small” transformer with tubelet size of 16x4
  - “Tiny” transformer with tubelet size of 16x8
- Single view is the same as a ViViT Factorised Encoder

How to fuse different views?

- Paper considers multiple alternatives.
- The best was using cross-attention from view *i+1* to view *i*, where views are ordered by increasing numbers of tokens.



How to fuse different views?

- The best was using cross-attention from view *i+1* to view *i,* where views are ordered by increasing numbers of tokens.

Model variants	Method	GFLOPs	MParams	Top-1
B/4	N/A	145	173	78.3
S/8		20	60	74.1
Ti/16		3	13	67.6
B/4+S/8+Ti/16	Ensemble	168	246	77.7
	Late fusion	187	306	80.6
	MLP	202	323	80.6
	Bottleneck	188	306	81.0
	CVA	195	314	**81.1**

What encoder should we use for each view?

- The encoder for each ”view” does not have to be the same
- Better to use a deeper encoder for the view with more tokens.

Model variants	GFLOPs	MParams	Top-1
B/8+Ti/2	81	161	77.3
B/2+Ti/8	337	221	81.3
B/8+S/4+Ti/2	202	250	78.5
B/2+S/4+Ti/8	384	310	81.8
B/4+S/8+Ti/16	195	314	81.1

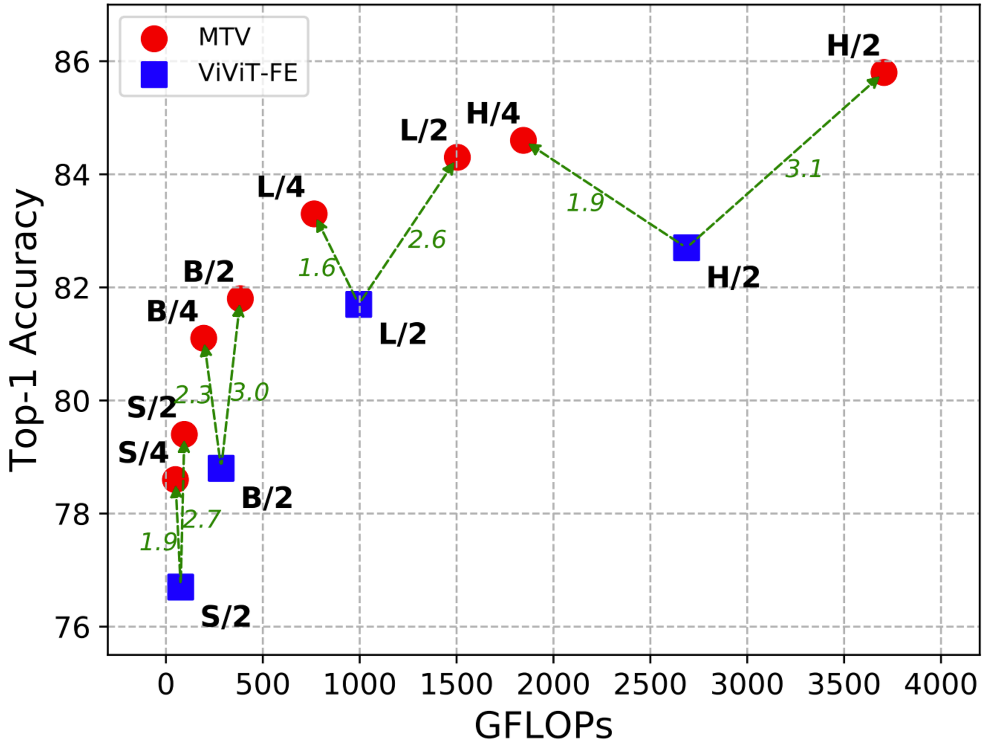
What encoder should we use for each view?

- The encoder for each ”view” does not have to be the same
- Using deeper encoder for other views does not help

Model variants	GFLOPs	MParams	Top-1
B/4+S/8+Ti/16	195	314	81.1
B/4+B/8+B/16	324	759	81.1
B/2+Ti/8	337	221	81.3
B/2+B/8	448	465	81.5
B/2+S/4+Ti/8	384	310	81.8
B/2+B/4+B/8	637	751	81.7

More views are better than deeper models

- It is better, in terms of accuracy and computational cost, to add multiple views in parallel, than to use a deeper, single-view model (ViViT).



State-of-the-art results

(a) Kinetics 400

Method	Top 1	Top 5	Views	TFLOPs
TEA [40]	76.1	92.5	10 × 3	2.10
TSM-ResNeXt-101 [41]	76.3	–	–	–
I3D NL [74]	77.7	93.3	10 × 3	10.77
VidTR-L [83]	79.1	93.9	10 × 3	10.53
LGD-3D R101 [52]	79.4	94.4	–	–
SlowFast R101-NL [23]	79.8	93.9	10 × 3	7.02
X3D-XXL [22]	80.4	94.6	10 × 3	5.82
OmniSource [20]	80.5	94.4	–	–
TimeSformer-L [6]	80.7	94.7	1 × 3	7.14
MFormer-HR [51]	81.1	95.2	10 × 3	28.76
MViT-B [21]	81.2	95.1	3 × 3	4.10
MoViNet-A6 [35]	81.5	**95.3**	1 × 1	0.39
ViViT-L FE [3]	81.7	93.8	1 × 3	11.94
MTV-B	**81.8**	95.0	4 × 3	4.79
MTV-B (320p)	**82.4**	95.2	4 × 3	11.16
Methods with web-scale pretraining				
VATT-L [2] (HowTo100M)	82.1	95.5	4 × 3	29.80
ip-CSN-152 [69] (IG)	82.5	95.3	10 × 3	3.27
R3D-RS (WTS) [19]	83.5	–	10 × 3	9.21
OmniSource [20] (IG)	83.6	96.0	–	–
ViViT-H [3] (JFT)	84.9	95.8	4 × 3	47.77
TokenLearner-L/10 [55] (JFT)	85.4	96.3	4 × 3	48.91
Florence [79] (FLD-900M)	86.5	97.3	4 × 3	–
CoVeR (JFT-3B) [81]	87.2	–	1 × 3	–
MTV-L (JFT)	84.3	96.3	4 × 3	18.05
MTV-H (JFT)	85.8	96.6	4 × 3	44.47
MTV-H (WTS)	**89.1**	**98.2**	4 × 3	44.47
MTV-H (WTS 280p)	**89.9**	**98.3**	4 × 3	73.57

(b) Kinetics 600

Method	Top 1	Top 5
SlowFast R101-NL [23]	81.8	95.1
X3D-XL [22]	81.9	95.5
TimeSformer-L [6]	82.2	95.6
MFormer-HR [51]	82.7	96.1
ViViT-L FE [3]	82.9	94.6
MViT-B [21]	83.8	96.3
MoViNet-A6 [35]	**84.8**	**96.5**
MTV-B	83.6	96.1
MTV-B (320p)	84.0	96.2
R3D-RS (WTS) [19]	84.3	–
ViViT-H [3] (JFT)	85.8	96.5
TokenLearner-L/10 [55] (JFT)	86.3	97.0
Florence [79] (FLD-900M)	87.8	97.8
CoVeR (JFT-3B) [81]	87.9	–
MTV-L (JFT)	85.4	96.7
MTV-H (JFT)	86.5	97.3
MTV-H (WTS)	**89.6**	**98.3**
MTV-H (WTS 280p)	**90.3**	**98.5**

(c) Something-Something v2

Method	Top 1	Top 5
SlowFast R50 [23, 77]	61.7	–
TimeSformer-HR [6]	62.5	–
VidTR [83]	63.0	–
ViViT-L FE [3]	65.9	89.9
MViT [21]	67.7	90.9
MFormer-L [51]	68.1	**91.2**
MTV-B	67.6	90.1
MTV-B (320p)	**68.5**	90.4

(d) Kinetics 700

	Top 1	Top 5
VidTR-L [83]	70.2	–
SlowFast R101 [23]	71.0	89.6
MoViNet-A6 [35]	72.3	–
MTV-L	**75.2**	**91.7**
CoVeR (JFT-3B) [81]	79.8	–
MTV-H (JFT)	78.0	93.3
MTV-H (WTS)	**82.2**	**95.7**
MTV-H (WTS 280p)	**83.4**	**96.2**

(e) Epic-Kitchens-100 Top 1 accuracy

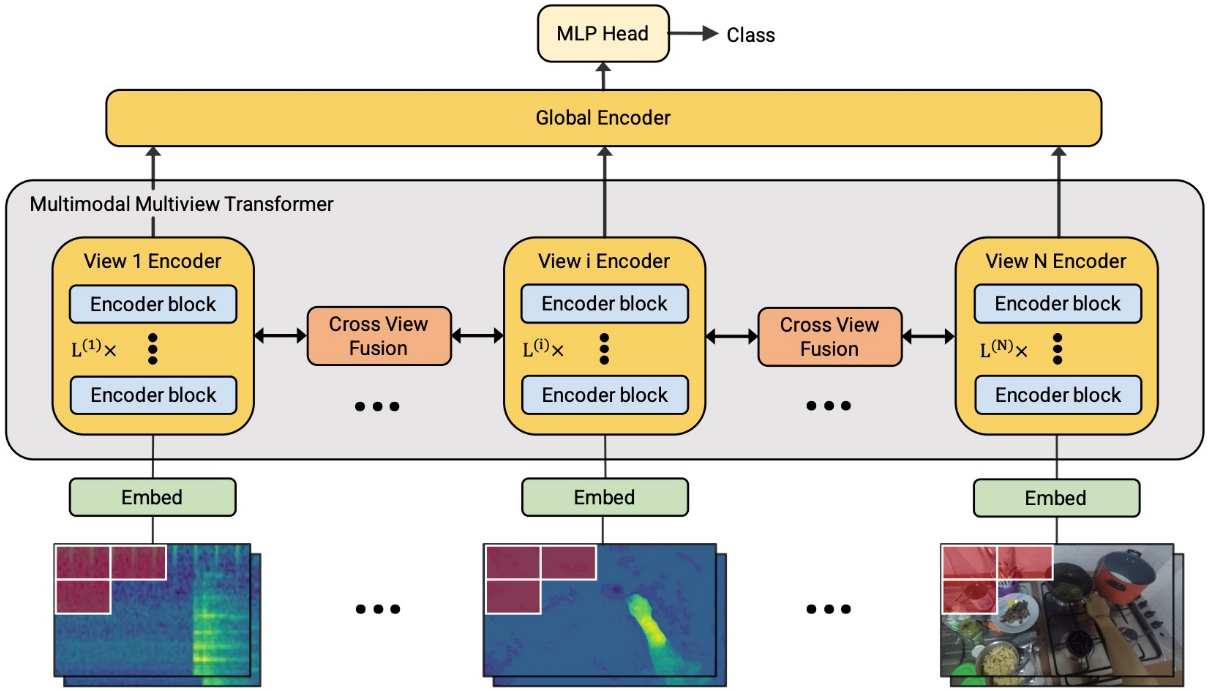
Method	Action	Verb	Noun
ViViT-L FE [3]	44.0	66.4	56.8
MFormer-HR [51]	44.5	67.0	58.5
MoViNet-A6 [35]	47.7	**72.2**	57.3
MTV-B	46.7	67.8	**60.5**
MTV-B (320p)	**48.6**	68.0	**63.1**
MTV-B (WTS 280p)	**50.5**	69.9	**63.9**

(f) Moments in Time

	Top 1	Top 5
AssembleNet-101 [56]	34.3	62.7
ViViT-L FE [3]	38.5	64.1
MoViNet-A6 [35]	40.2	–
MTV-L	**41.7**	**69.7**
VATT-L (HT100M) [2]	41.1	67.7
MTV-H (JFT)	**44.0**	**70.2**
MTV-H (WTS)	**45.6**	**74.7**
MTV-H (WTS 280p)	**47.2**	**75.7**

Multimodal MTV

- Recent extension of MTV to multiple modalities
- Each “view” is now a different modality
- Won the Epic-Kitchens action recognition challenge



Conclusion

- Processing multiple “views” in parallel allows us to achieve superior accuracy-speed trade-offs for video classification
- State-of-the-art results across 6 datasets.

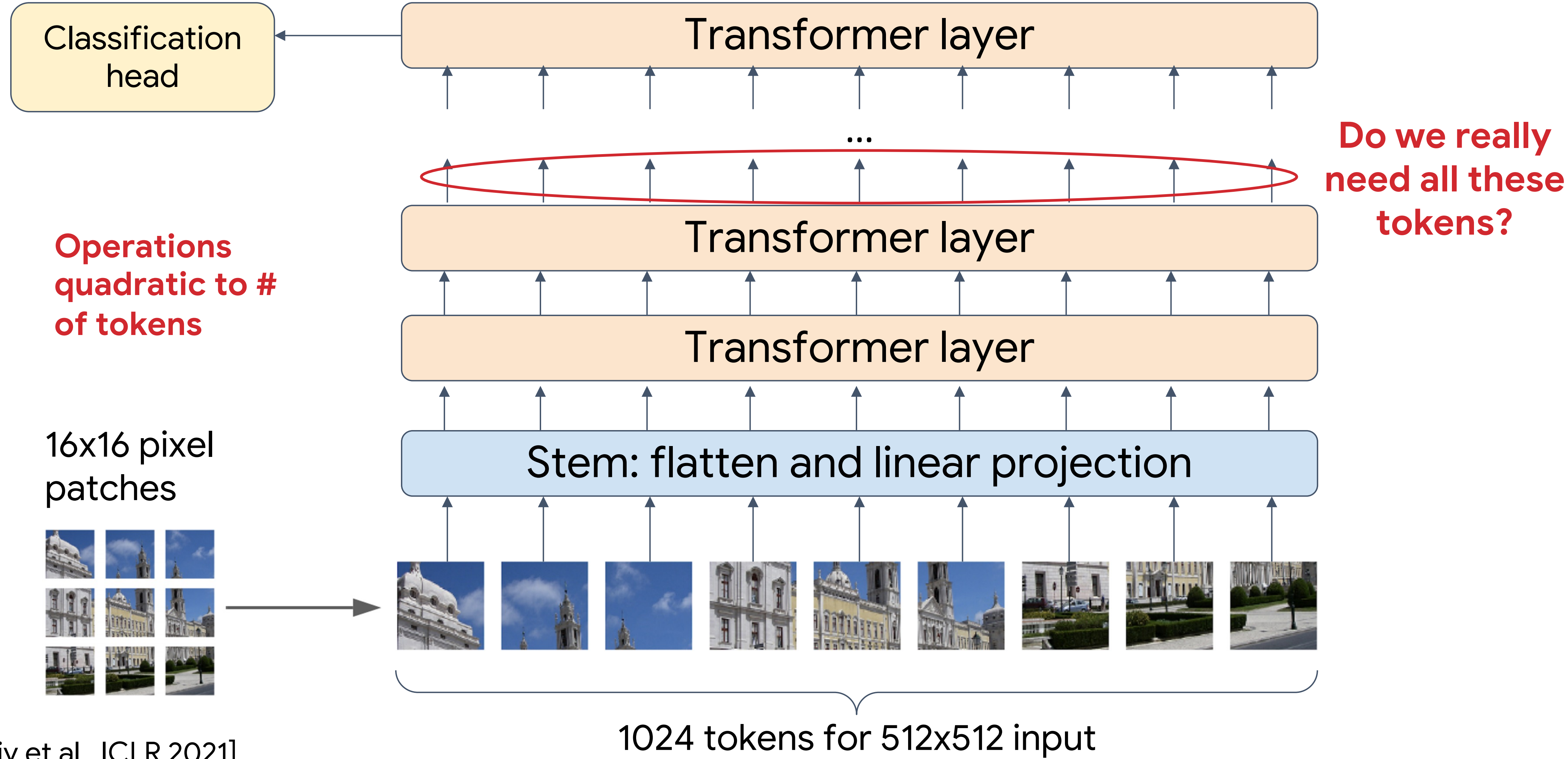
- Poster session on Tuesday afternoon, 75b
- [Paper], [Code]

TokenLearner: What Can 8 Learned Tokens do for Images and Video

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, Anelia Angelova

Google Research

Vision Transformers

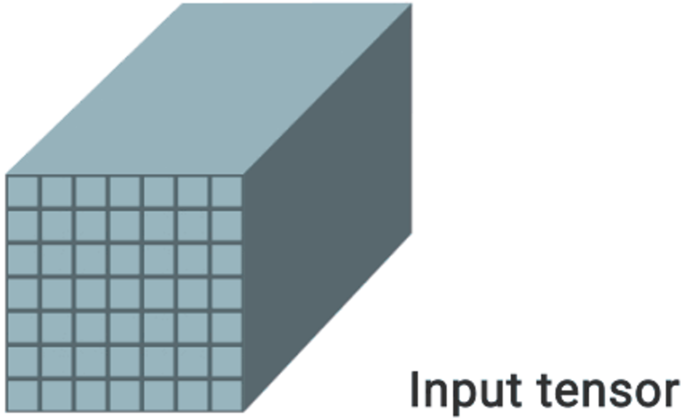


[Dosovitskiy et al., ICLR 2021]

Motivation

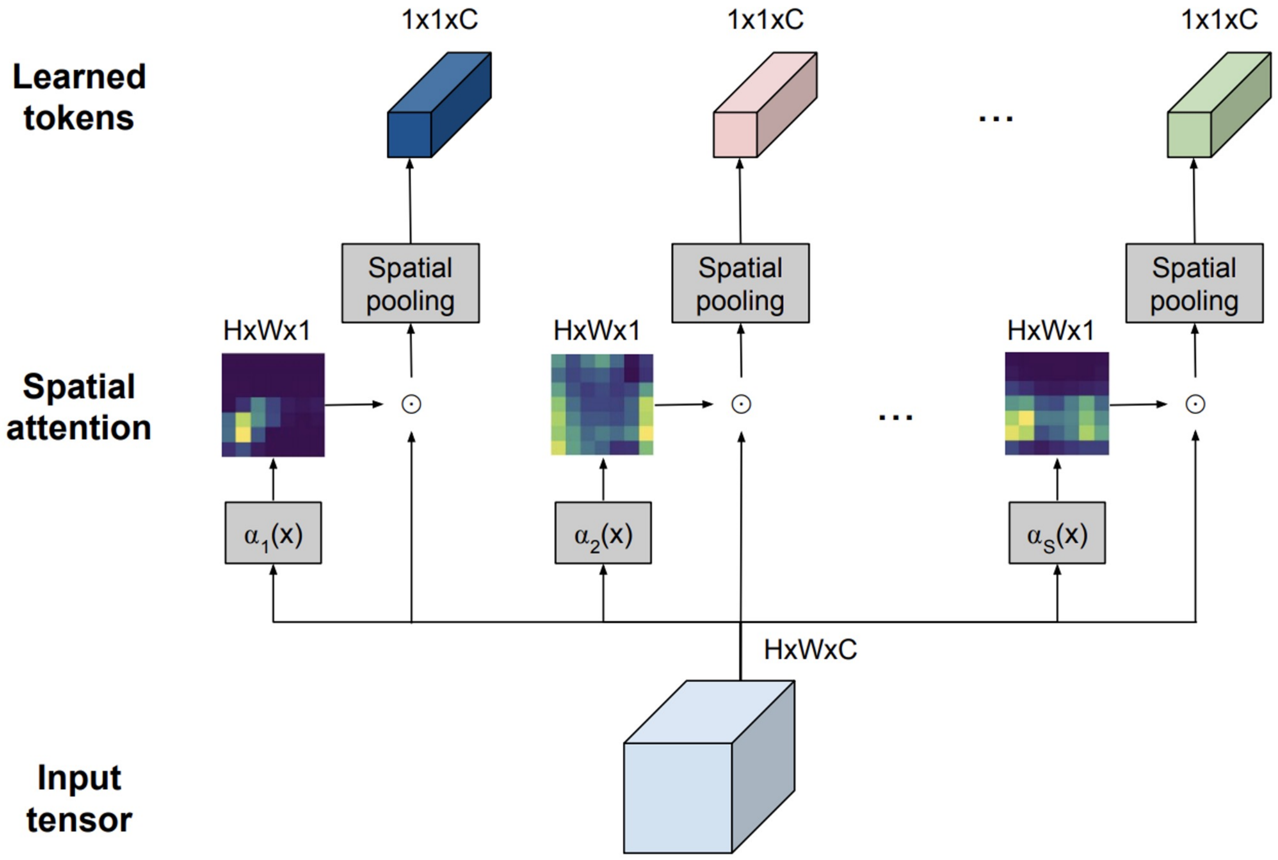
- Transformers have quadratic complexity with respect to the number of tokens.
- Do we really need that many tokens and process them all at every layer?
- Can we not ‘learn’ to adaptively obtain much fewer tokens instead, and focus on processing them?

TokenLearner



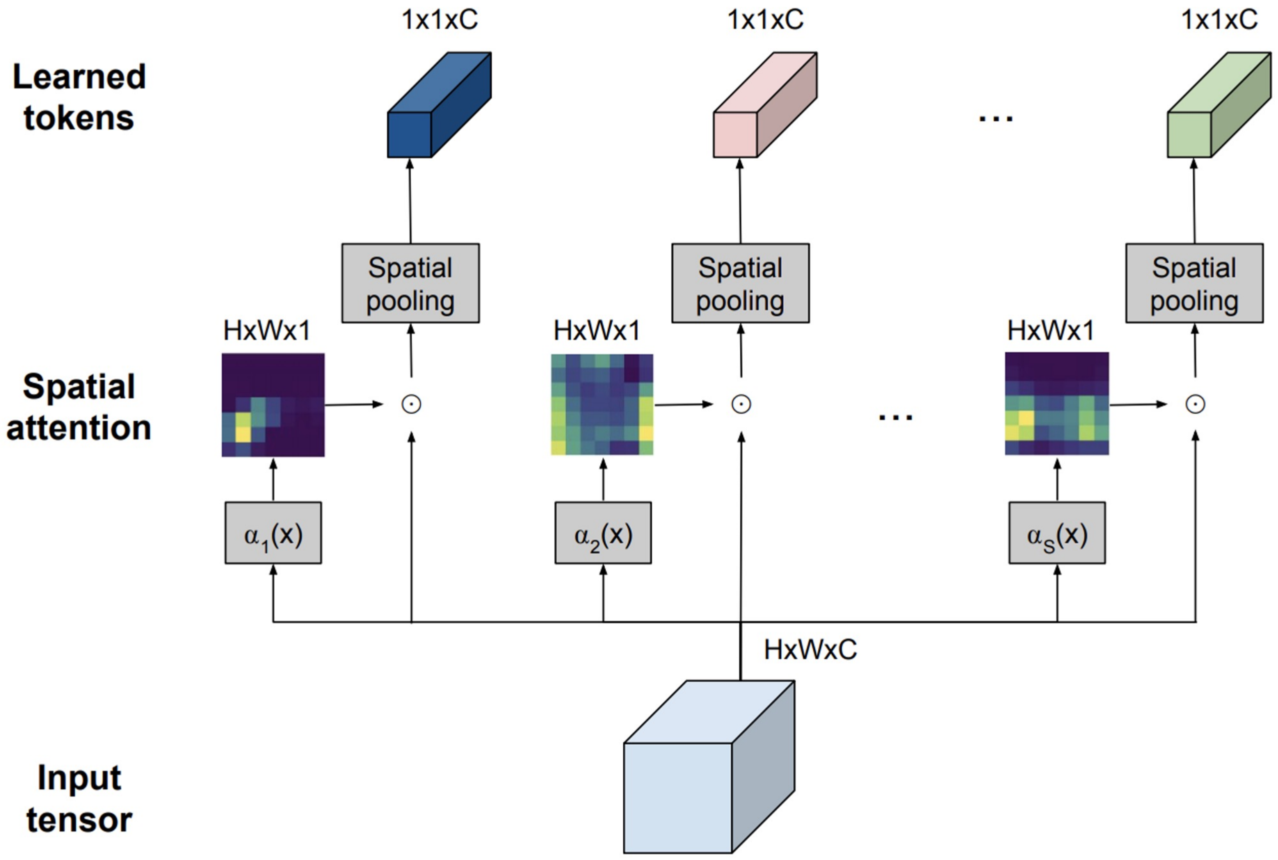
TokenLearner

- TokenLearner has a form of spatial attention mechanism
- Given an image-like tensor, it
  - Weights each pixel differently (i.e., focuses on a subset of pixels)
  - Summarizes them as a token.
- Could be applied to intermediate tensors
- Works well with a small number of tokens! Example: 8 or 16



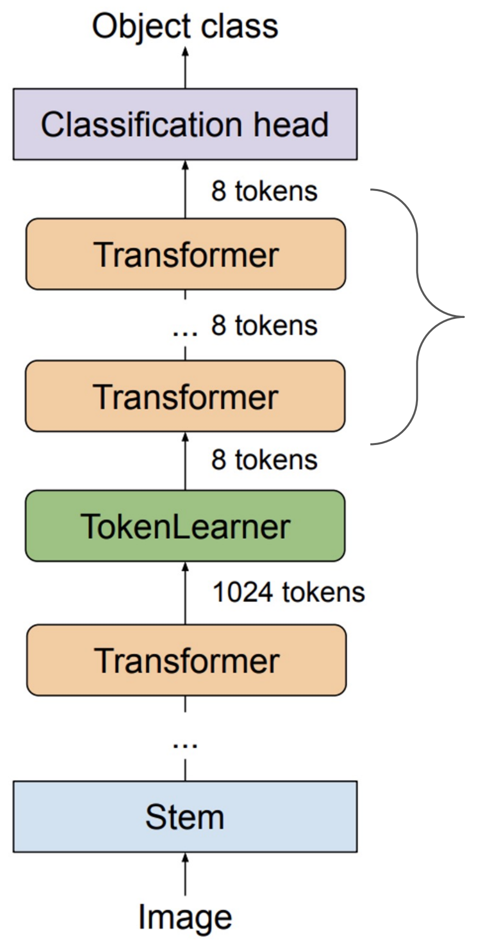
TokenLearner

- The $\alpha(\cdot)$ function can be anything
- Examples
  - Conv layers
  - MLP
  - Attention (equivalent to Perceiver)
- When implementing, $\alpha_{1:S}(\cdot)$ is a single function with S output channels.



TokenLearner within ViT

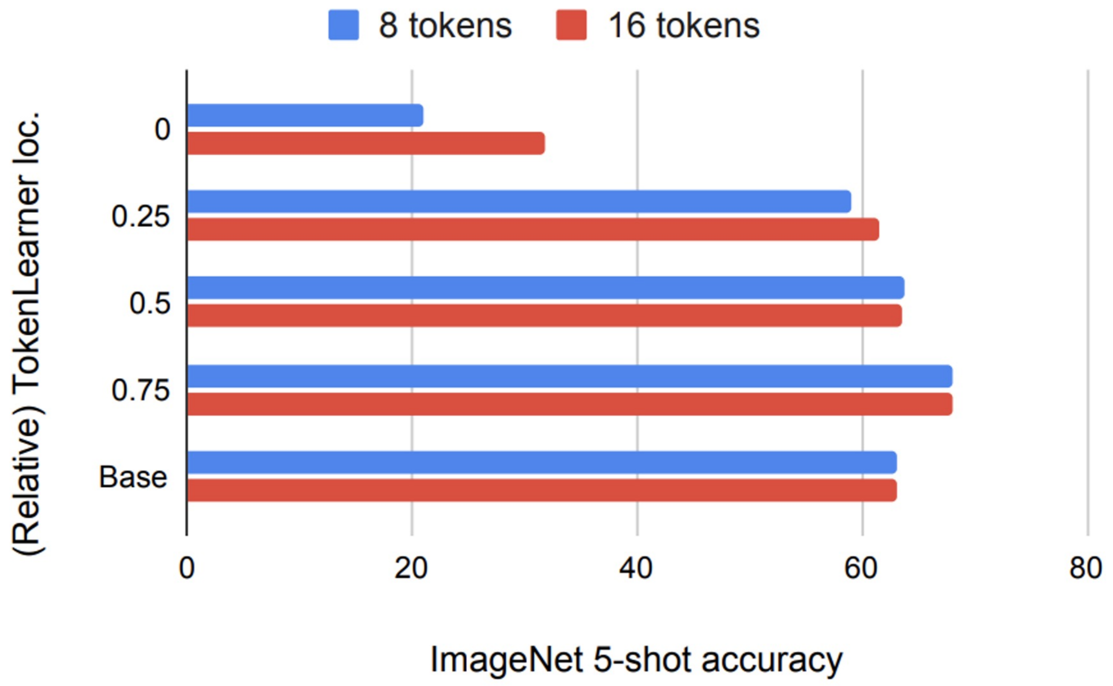
- TokenLearner module inserted in the middle of Transformer architecture
- The computation after the TokenLearner module becomes negligible.

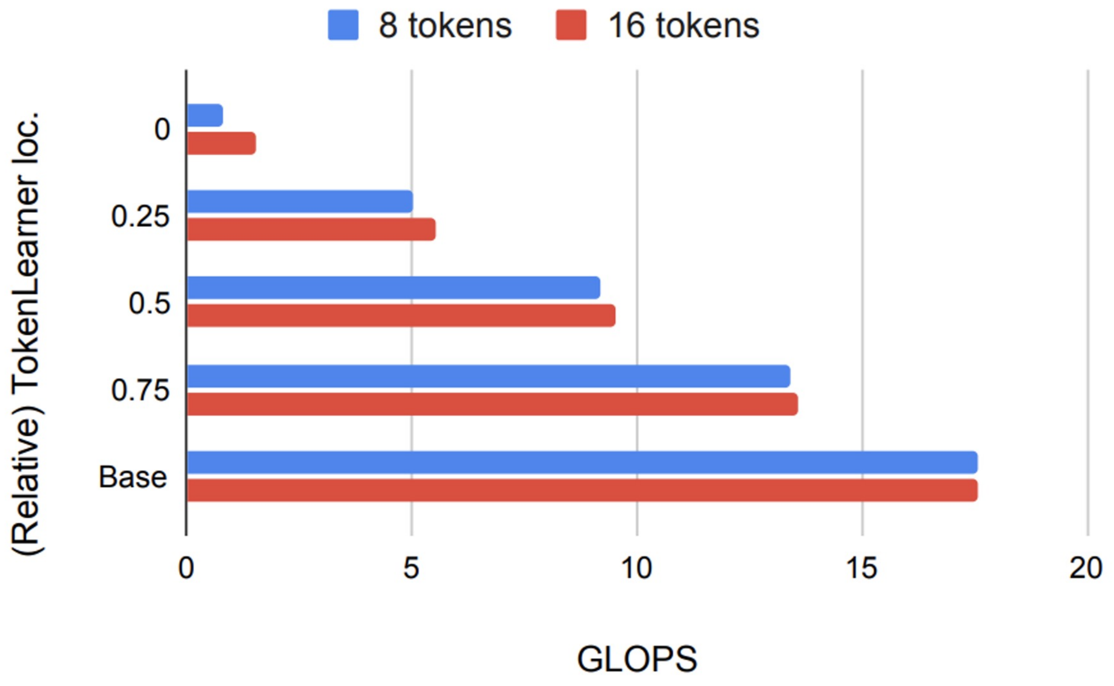


Computation negligible

Where do we place TokenLearner?

- Interestingly, TokenLearner performs better, while being faster. Adaptiveness!
- Experiment using ViT-B, pretraining on JFT and doing ImageNet few-shot evaluation (same setting as original ViT paper).





Scaling up TokenLearner

- By using TokenLearner, we can now
  - Process more initial tokens (use smaller patch sizes)
  - Use more transformer layers.
- Results using ViT-L with 512x512 inputs, and 16 learned tokens.

Base	# layers	TokenLearner	GFLOPS	ImageNet Top1
ViT L/16	24	-	363.1	87.35
ViT L/16	24	16-TL at 12	178.1	87.68
ViT L/16	24+11	16-TL at 12	186.8	87.47
ViT L/14	24+11	16-TL at 18	361.6	88.37

Scaling up TokenLearner

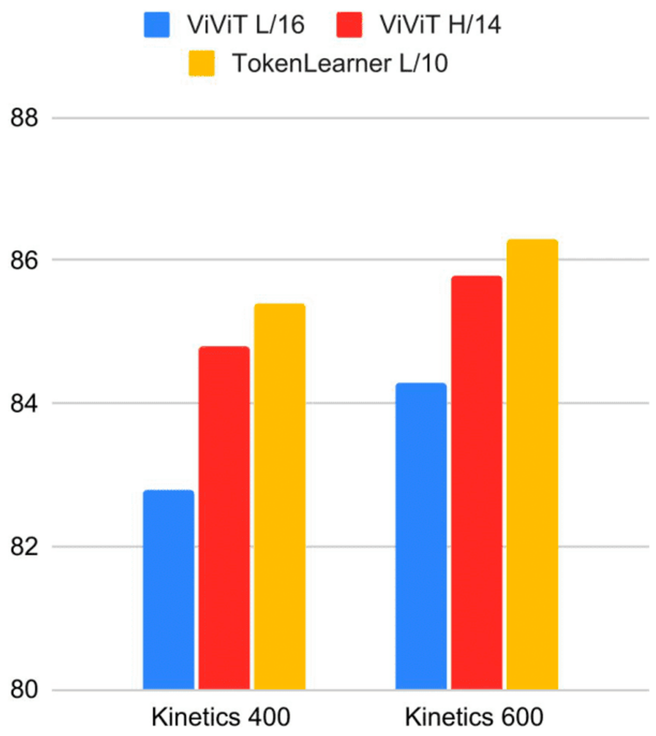
- By using TokenLearner, we can now
  - Process more initial tokens (use smaller patch sizes)
  - Use more transformer layers.
- Results using ViT-L with 512x512 inputs, and 16 learned tokens.

Method	# params.	ImageNet	ImageNet ReaL
BiT-L	928M	87.54	90.54
ViT-H/14	654M	88.55	90.72
ViT-G/14	1843M	**90.45**	90.81
TokenLearner L/10 (24+11)	**460M**	88.5	90.75
TokenLearner L/8 (24+11)	**460M**	88.87	**91.05**

TokenLearner on video

- Once again, we can use the higher efficiency of TokenLearner to process more tokens and achieve state-of-the-art results.
- Results from inserting TokenLearner into ViViT-L, at time of publication:

	TokenLearner	Previous SOTA
Kinetics-400	85.4	84.9
Kinetics-600	86.3	86.1
Charades	66.3	63.2
AViD	53.8	50.9



Conclusion

- There are lots of redundant tokens in images and video.
- We can learn to summarise them into a smaller subset of tokens, and process only those.
- With more efficient models, we can process more tokens to improve accuracy.

- M Ryoo et al. TokenLearner: What Can 8 Learned Tokens Do for Images and Video. *NeurIPS* 2021.
- [Paper], [Code], [Blog]

Google Research

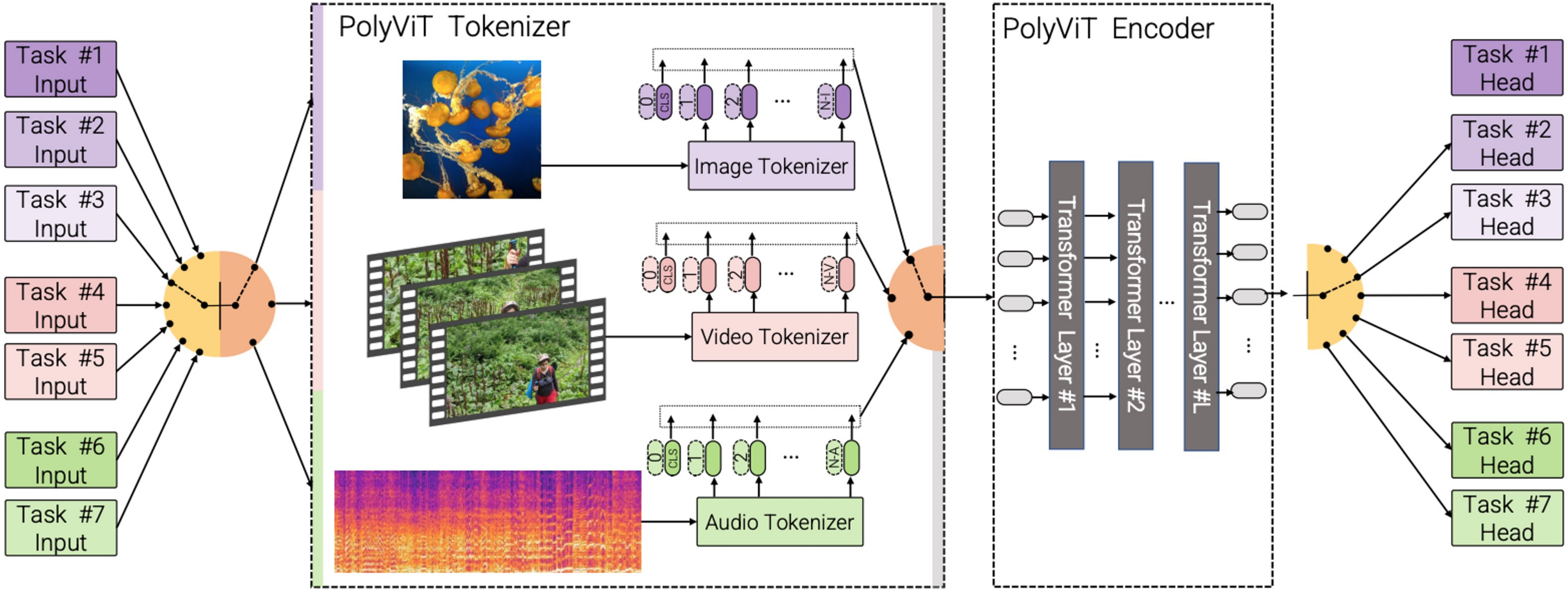
PolyViT: Co-Training Vision Transformers on Images, Video and Audio

Valerii Likhosherstov*, Anurag Arnab*, Yi Tay, Mario Lucic, Krzysztof Choromanski, Mostafa Dehghani*

Google Research

Motivation

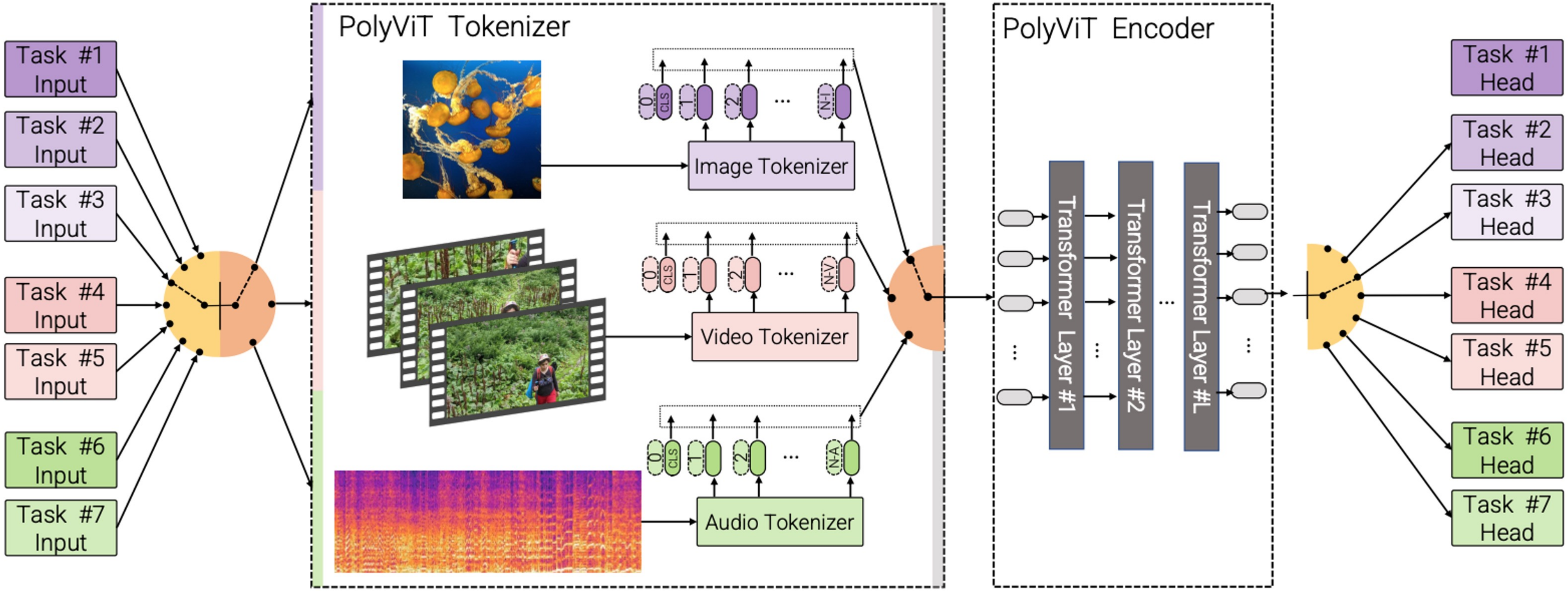
- Transformers are used for wide range of perception tasks
- Unified architecture, with shared parameters, for wide range of tasks?



earch

PolyViT

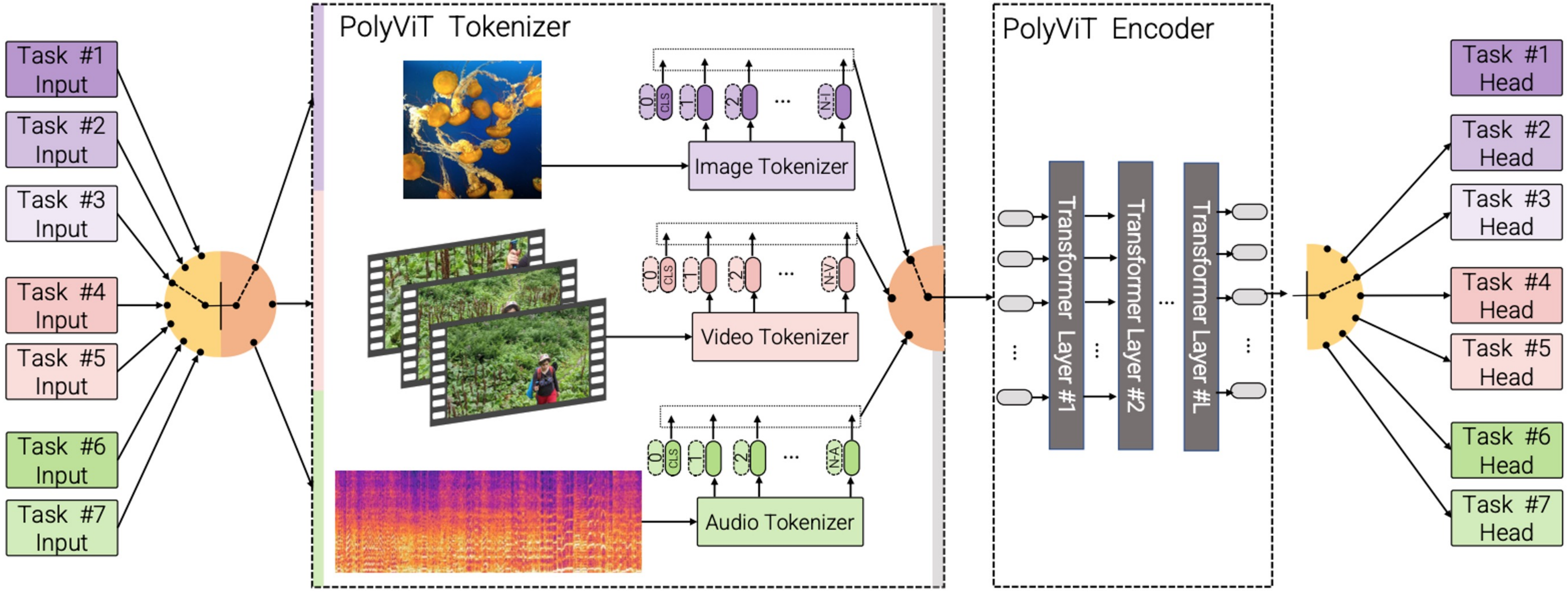
- Only task-specific parameters: output linear head
- Tokenizer is modality-specific



earch

PolyViT

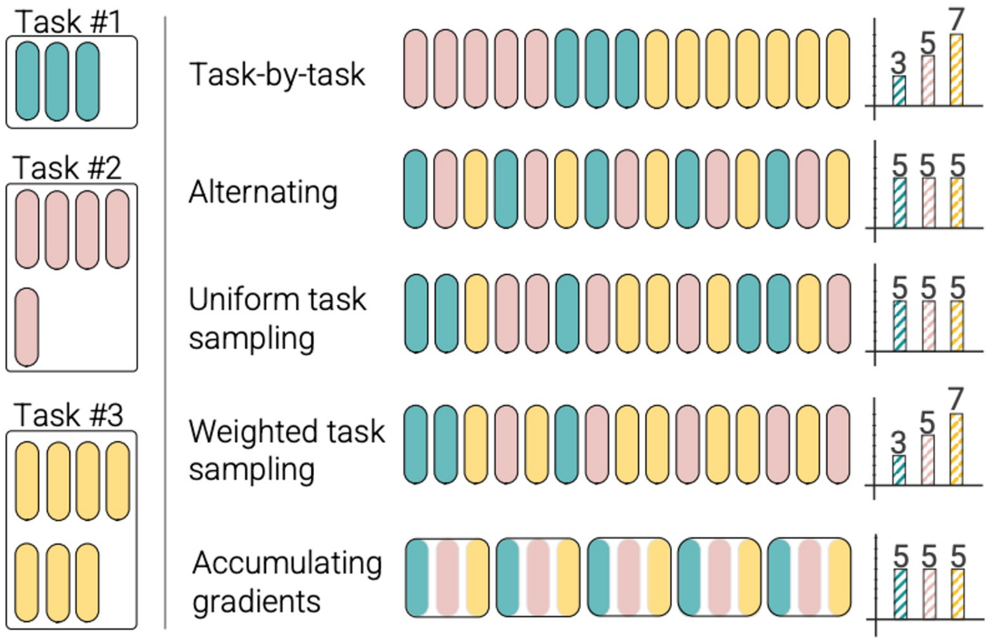
- Model can process multiple tasks and modalities. Performs a single task at a time.
- Model is very parameter efficient. No gains in FLOPs or runtime.



le Research

How to co-train?

- When training, sample batches from a single task
    - Ablate different sampling schedules
- We can reuse training hyperparameters from a single-task baseline.
- No additional tuning necessary!
- Total number of training steps does not increase either



How to co-train?

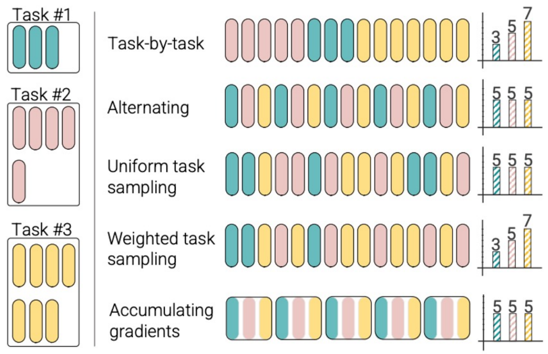


	Image					Video		Audio	
Schedule	Im1K	C100	C10	Pets	R45	K400	MiT	MiniAS	VGG
Task-by-task	0.3	0.8	11.7	1.9	2.0	0.3	0.3	1.6	37.2
Accumulated	**88.1**	90.0	98.8	94.0	96.1	58.0	22.5	22.9	27.3
Alternating	86.0	89.4	99.2	94.0	95.8	69.7	30.0	31.4	44.6
Uniform	85.8	89.3	98.6	94.6	96.1	68.8	29.3	30.6	44.1
Weighted	86.9	**90.4**	**99.3**	**96.5**	**97.0**	**71.6**	**32.5**	**33.5**	**49.2**

How to solve 9 tasks across 3 modalities

Model	#Models	#Params	Image					Video		Audio	
			Im1K	C100	C10	Pets	R45	K400	MiT	MiniAS	VGG
ViT-Im21K Linear probe	1	**93M**	80.7	76.2	91.7	91.8	81.7	64.0	25.5	11.3	15.7
Single-task baseline	9	773M	83.1	92.0	99.0	94.5	**96.7**	78.7	33.8	29.3	**51.7**
PolyViT, 1 modality	3	263M	**84.3**	**93.3**	**99.1**	**95.1**	96.4	**80.2**	**36.5**	**36.7**	51.6
PolyViT, $L_{adapt} = 0$	1	**93M**	83.1	91.2	99.0	95.0	**96.7**	77.5	33.2	32.3	50.6
PolyViT, $L_{adapt} = L/2$	1	178M	82.8	91.5	99.0	95.0	96.6	79.4	35.3	33.1	51.5

- Baselines
  - Train a model for each classification task ⟶ lots of parameters
  - Linear probe on ImageNet-21K pretrained model ⟶ parameter-efficient
- Co-train a model for each modality
  - Best accuracy whilst saving parameters
- Co-train a model across all tasks and modalities
  - 8.3x parameter reduction whilst losing maximum of 1.2% accuracy

Google Research

Evaluating learned feature representations

Model	Finetuning	Image: C-ch101	Image: SUN397	Image: Dmlab	Image: DTD	Image: KITTI	Image: PCAM	Video: Epic K.	Video: S-S v2	Video: K600	Audio: MiT-A	Audio: K400-A
ViT-Im21K pretrained	–	88.9	75.7	41.0	72.1	46.9	80.2	10.0	17.8	66.6	4.9	10.8
ViT	ImageNet-1K	**91.0**	79.3	45.6	71.9	52.5	80.7	12.2	18.5	67.9	5.3	12.0
PolyViT	Image tasks	90.7	**80.0**	45.2	**72.5**	53.8	81.2	12.1	17.9	67.9	5.3	11.9
ViViT	MiT	85.2	73.8	43.0	69.9	**54.9**	81.7	14.9	26.3	74.2	5.1	11.9
PolyViT	Video tasks	89.2	77.5	**45.9**	71.1	53.5	**83.8**	17.2	27.9	**79.7**	5.3	12.2
AST	VGGSound	29.0	7.6	29.8	34.7	45.1	79.5	2.9	4.6	10.6	9.7	21.7
PolyViT	Audio tasks	38.8	14.7	31.4	40.1	43.2	78.4	3.0	5.8	14.5	**10.3**	**22.0**
PolyViT $L_{adapt}=0$	All	**91.0**	78.2	45.8	71.8	52.3	81.9	16.8	27.9	77.8	9.6	20.6
PolyViT $L_{adapt}=L/2$	All	90.7	77.8	45.1	72.1	52.5	82.3	**18.0**	**28.7**	79.4	9.9	21.1

- Linear evaluation (like self-supervised learning) on new datasets
- Multi-modal PolyViT generalizes to all new tasks.
- Image models transfer well to video and vice versa.
- Audio models do not generalize at all (and vice versa).

Google Research

State-of-the-art by co-training on one modality

- Comparison to ViViT unfactorized model
- Largest improvements on smaller datasets (Kinetics 400)
  - Co-training has a regularising effect.

Model	#Models	#Params	Kinetics 400 Top 1	Kinetics 400 Top 5	Kinetics 600 Top 1	Kinetics 600 Top 5	Moments in Time Top 1	Moments in Time Top 5
ViViT	3	913M	80.6	94.7	82.5	**95.6**	38.0	64.9
PolyViT	1	**308M**	**82.4**	**95.0**	**82.9**	95.5	**38.6**	**65.5**

State-of-the-art by co-training on one modality

- Comparison to MBT, audio-only
- Largest improvements on smaller datasets.
  - Co-training has a regularising effect.

Model	#Models	#Params	AudioSet mAP	VGGSound Top 1	VGGSound Top 5
MBT (audio-only)	2	172M	44.3	52.3	78.1
PolyViT	1	**87M**	**44.5**	**55.1**	**80.4**

Conclusion

- Co-training on one modality
  - Improves accuracy on all tasks. Has a regularising effect
- Co-training on multiple modalities and tasks
  - Even more parameter-efficient.
  - Learns universal features that are useful for a wide range of tasks.
- Co-training is simple and practical to do.
  - Does not require additional hyperparameter tuning over single-task baselines.

Questions?

- A Arnab et al. ViViT: A Video Vision Transformer. *ICCV* 2021.
- S Yan et al. Multiview Transformers for Video Recognition. *CVPR* 2022.
- M Ryoo et al. TokenLearner: What Can 8 Learned Tokens Do for Images and Video. *NeurIPS* 2021.
- V Likhosherstov et al. PolyViT: Co-training Vision Transformers on Images, Video and Audio. arXiv:2111.12993