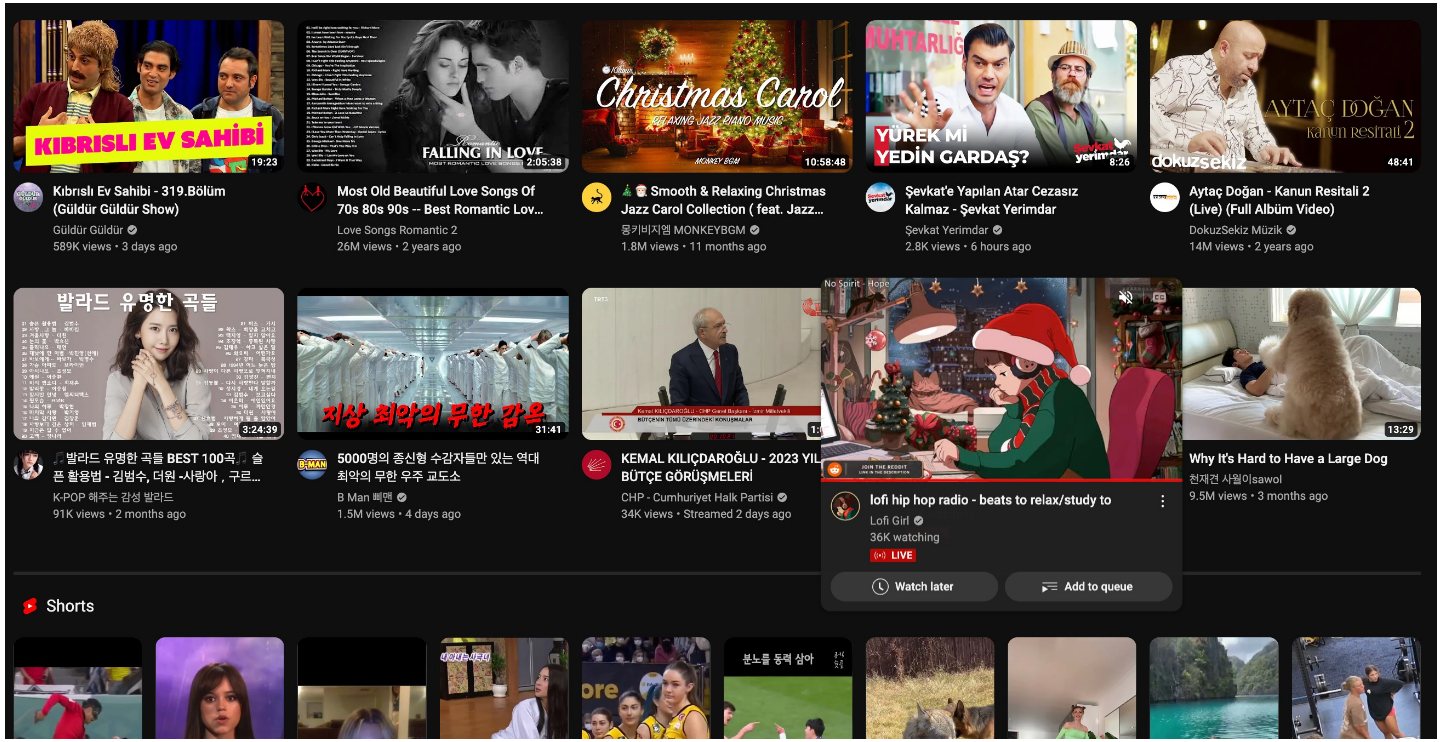
Large-Scale Video Understanding with Transformers

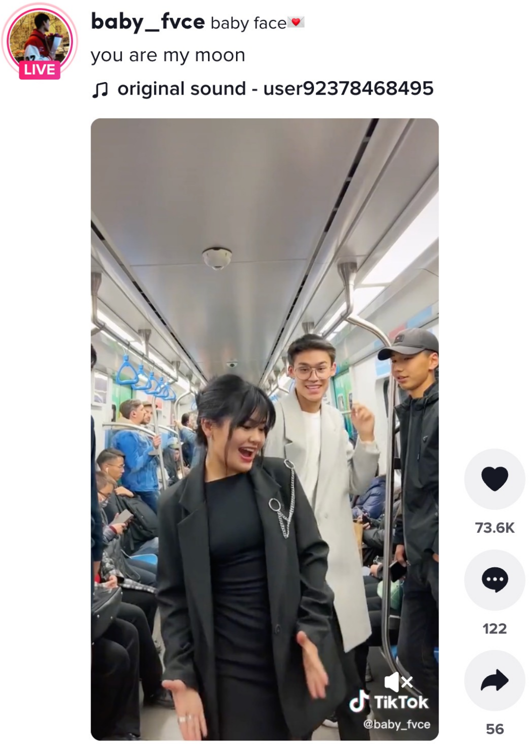
Anurag Arnab

Google Research

Introduction

- About 270 000 hours of videos uploaded every day on YouTube alone!
- How can we make sense of all the uploaded content



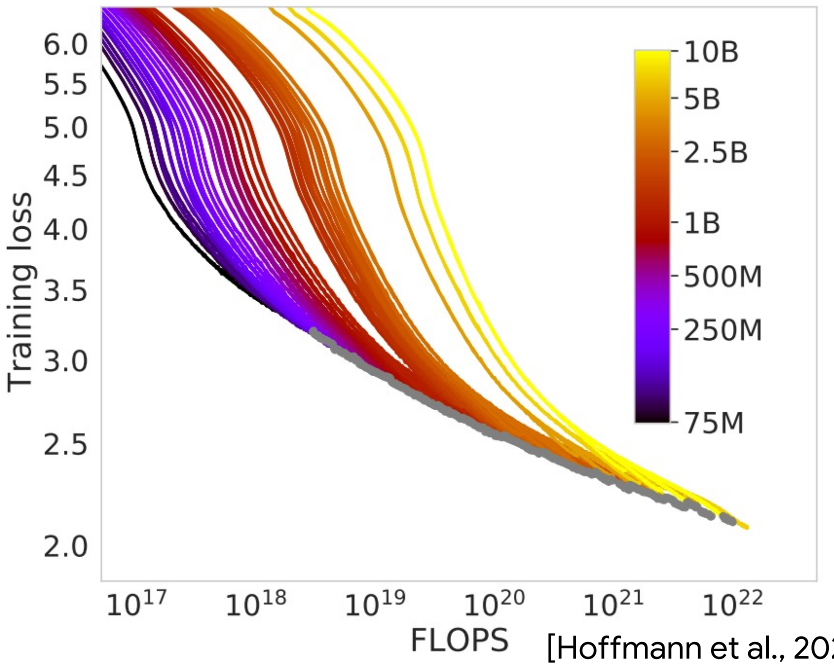


Introduction

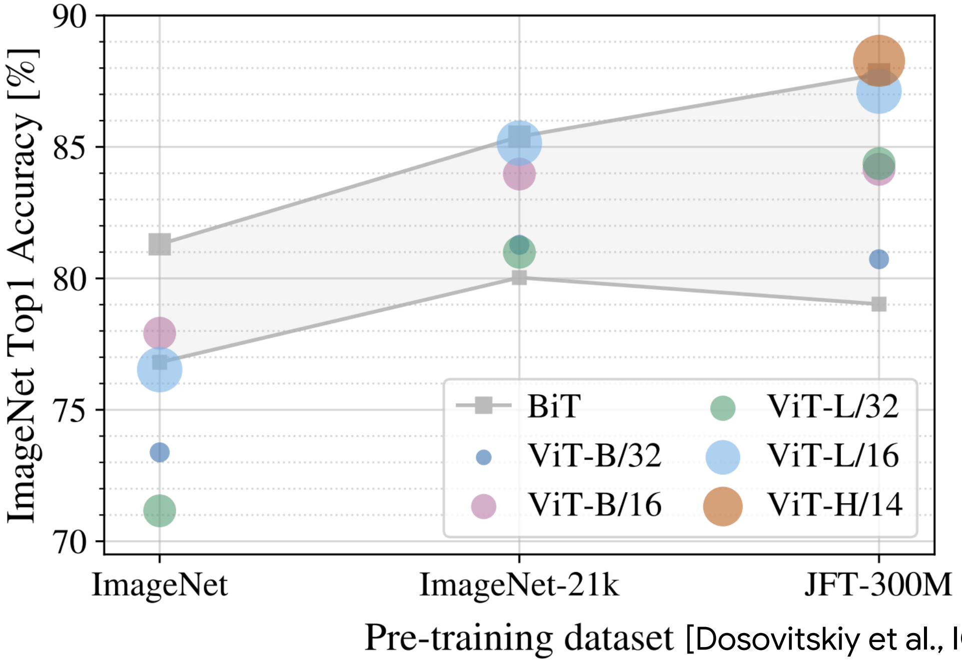
- Transformers achieve state-of-the-art performance in a wide range of domains.
- And that motivates us to develop transformer-based models for video understanding.

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 shine when training on large datasets
   - How can we pretrain them in a data-efficient way?

ViViT: A Video Vision Transformer

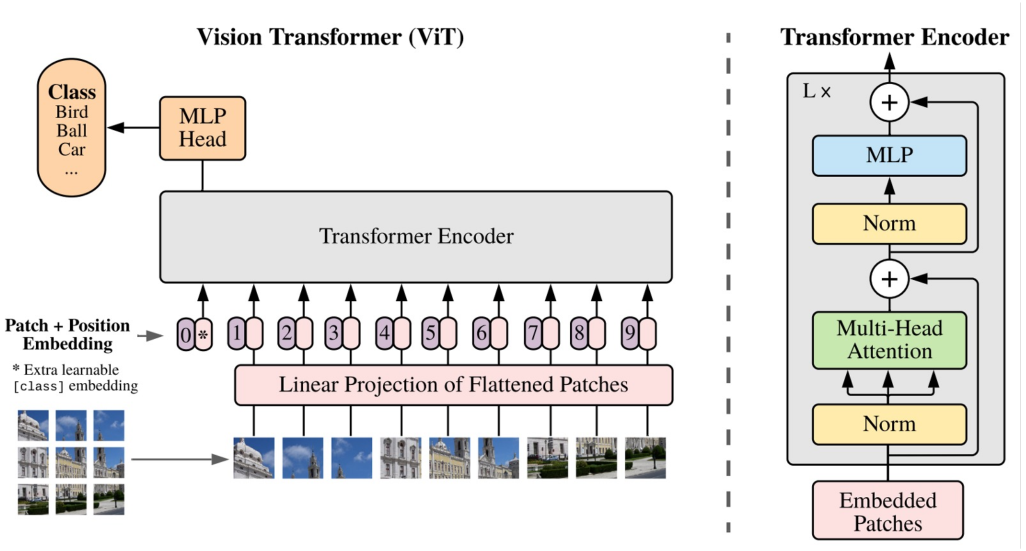
Anurag Arnab, Mostafa Dehghani,
Georg Heigold, Chen Sun,
Mario Lucic, Cordelia Schmid

ICCV 2021

Google Research

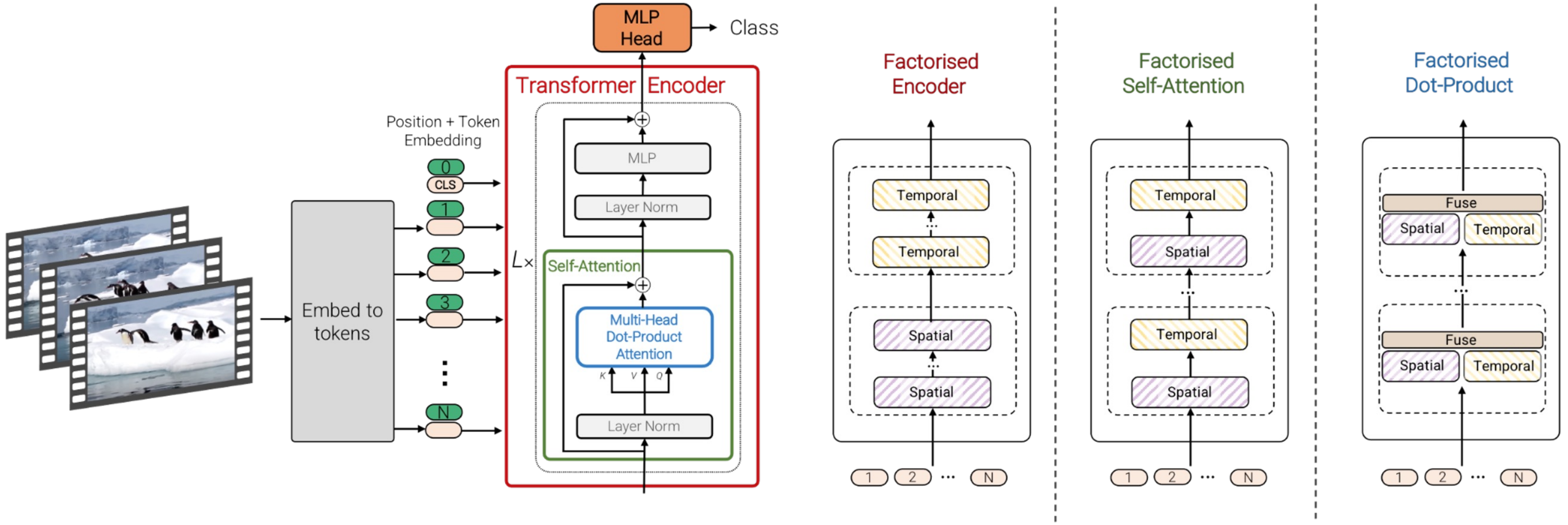
Introduction

- 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

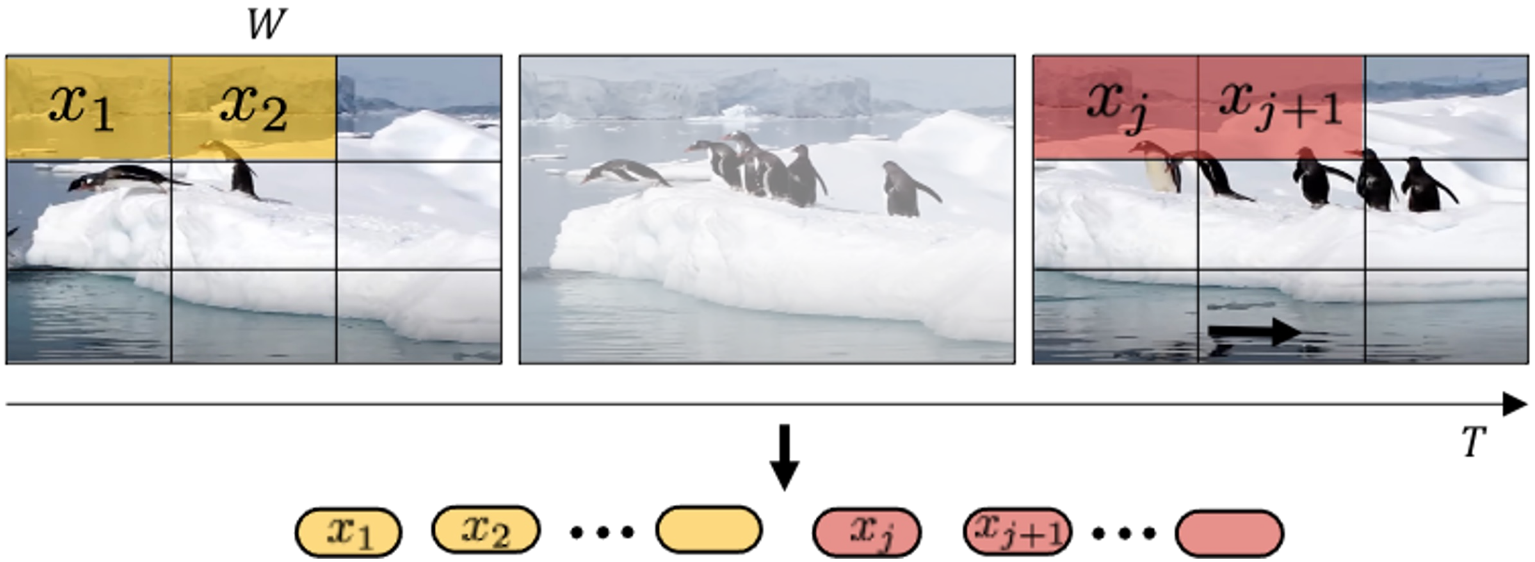
- Extend idea of ViT (static images) to videos
- To handle large number of tokens, explore more efficient factorised attention variants.
- Regularisation to train on comparatively small video datasets.



Google Research

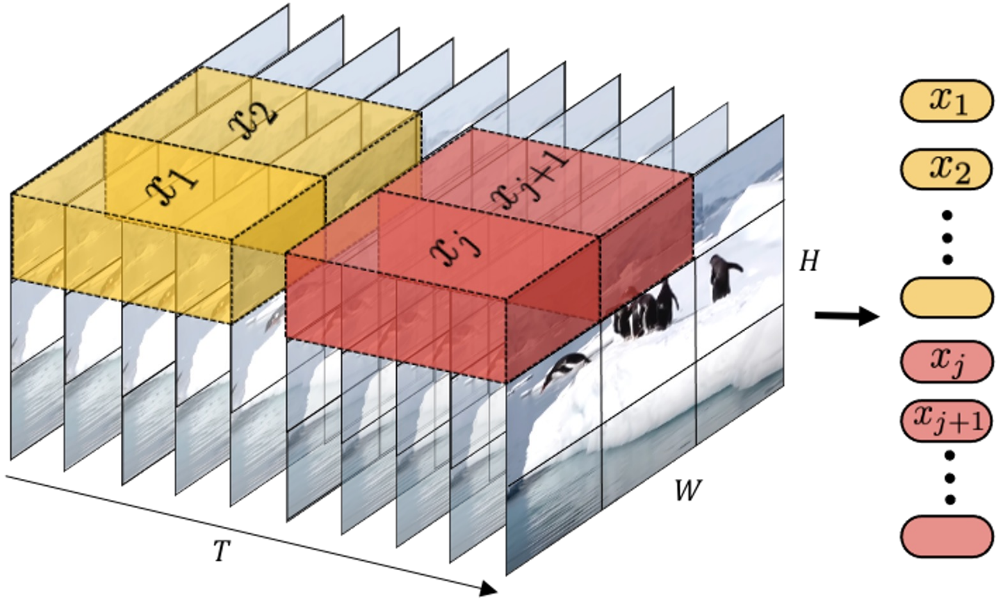
Input Encoding 1: Uniform Frame Sampling

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



Input Encoding 2: Tubelet embedding

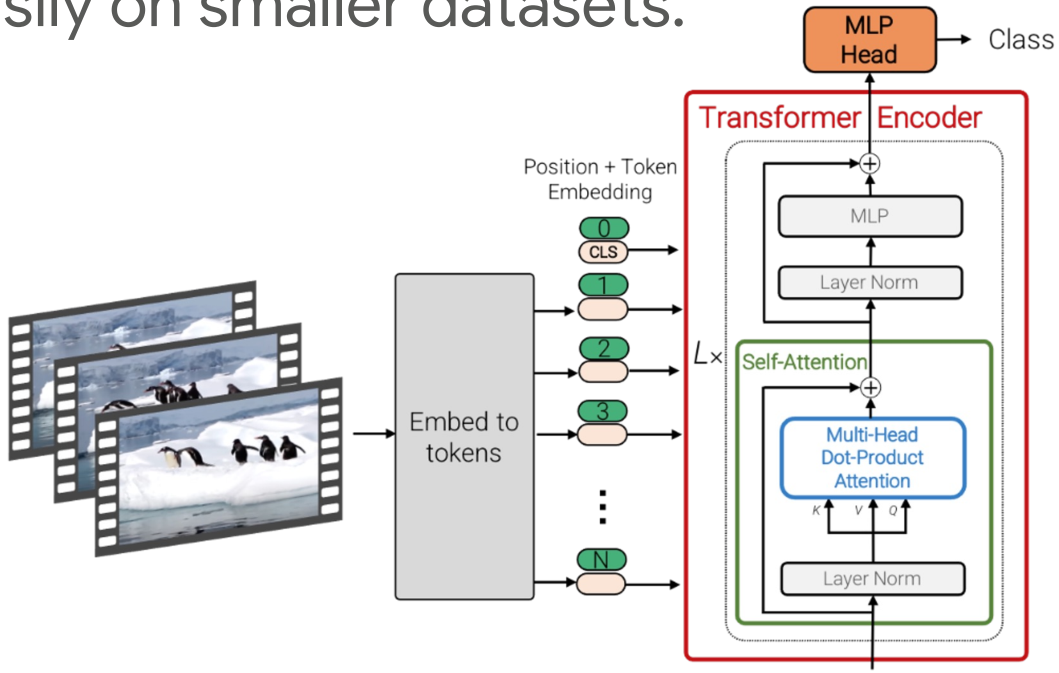
- Extract 3D tubelets to encode spatio-temporal “tubes” into tokens
- Temporal information included from the initial tokenisation stage.
- Works better when initialised appropriately.



Google Research

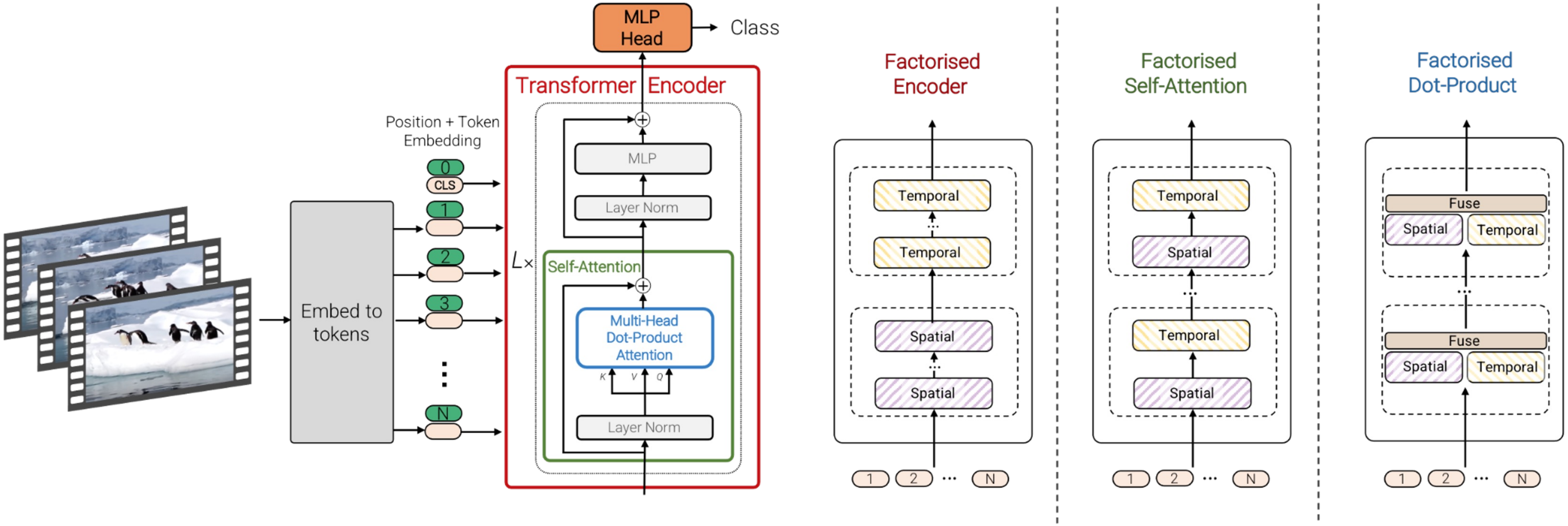
ViViT: Joint Spatio-Temporal Attention

- 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.



Google Research

ViViT: Space/Time Factorisations



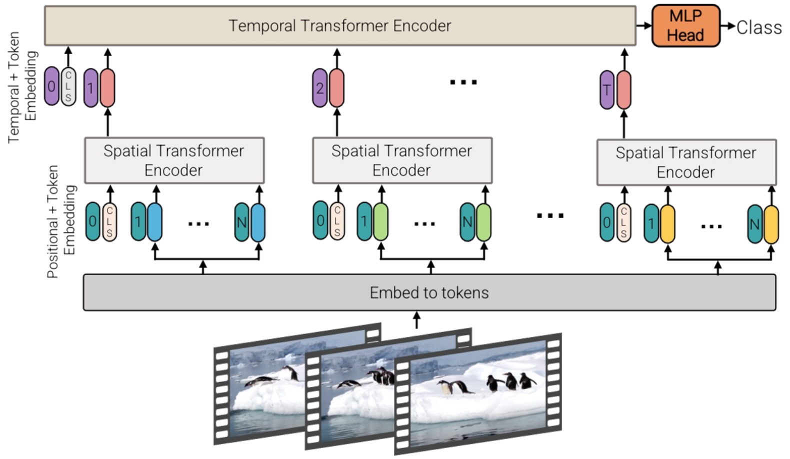
Alternative ways of mixing the temporal and spatial information

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

ViViT Factorisations

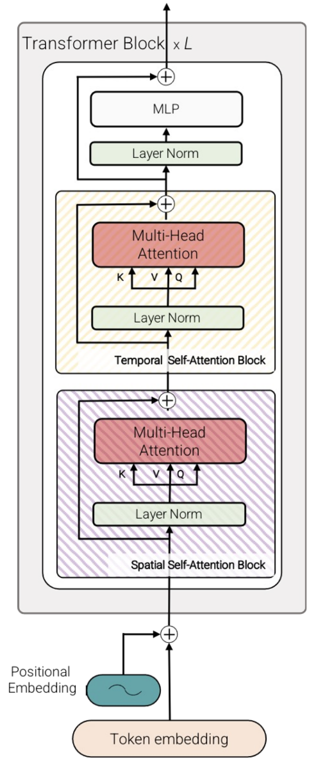
Factorised encoder

- “Late fusion” of spatial and temporal information



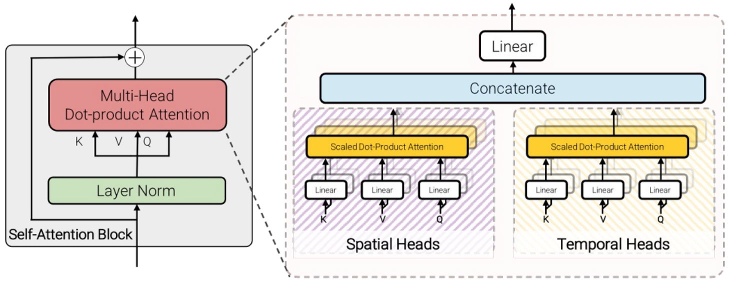
Factorised self-attention

- 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.

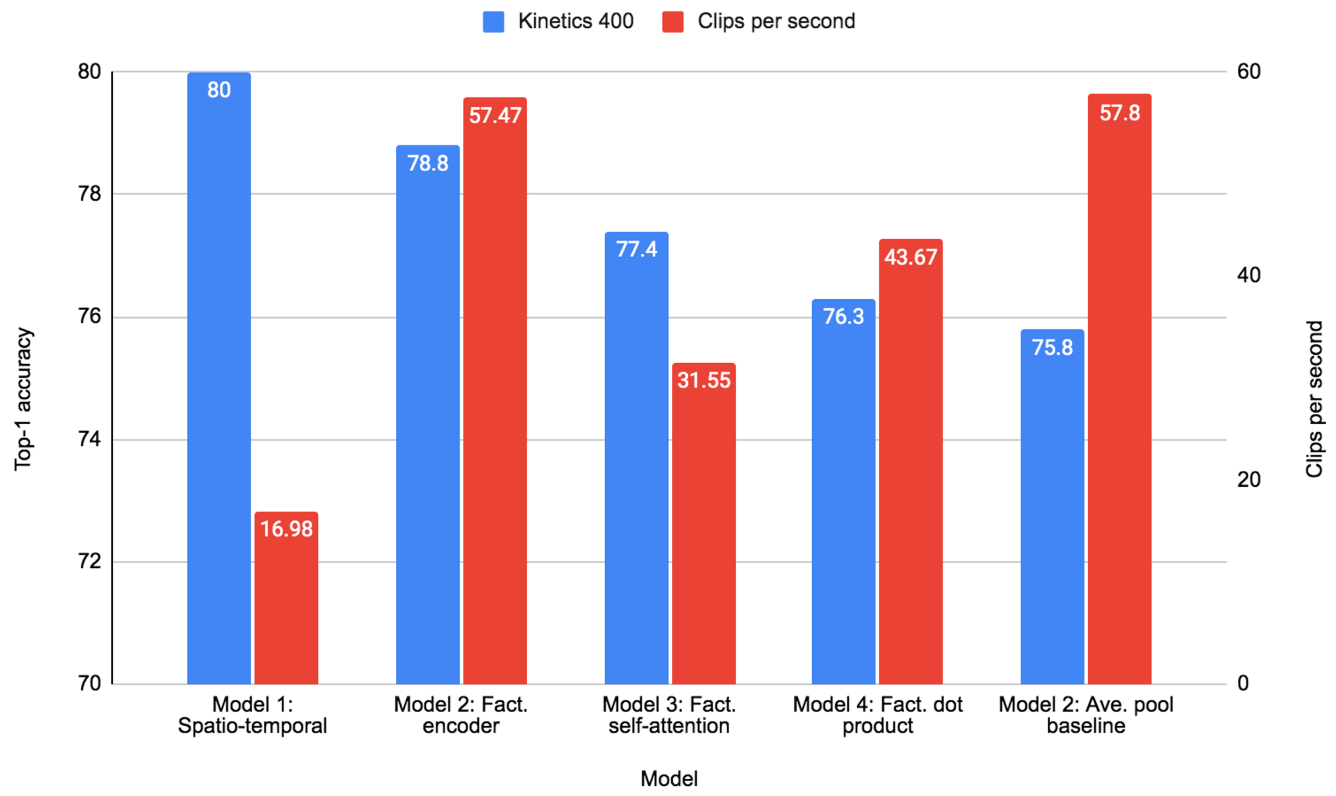
	Top-1 accuracy
Uniform frame sampling	78.5
Tubelet embedding	
Random initialisation [22]	73.2
Filter inflation [6]	77.6
Central frame	79.2

Google Research

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

Model Variants

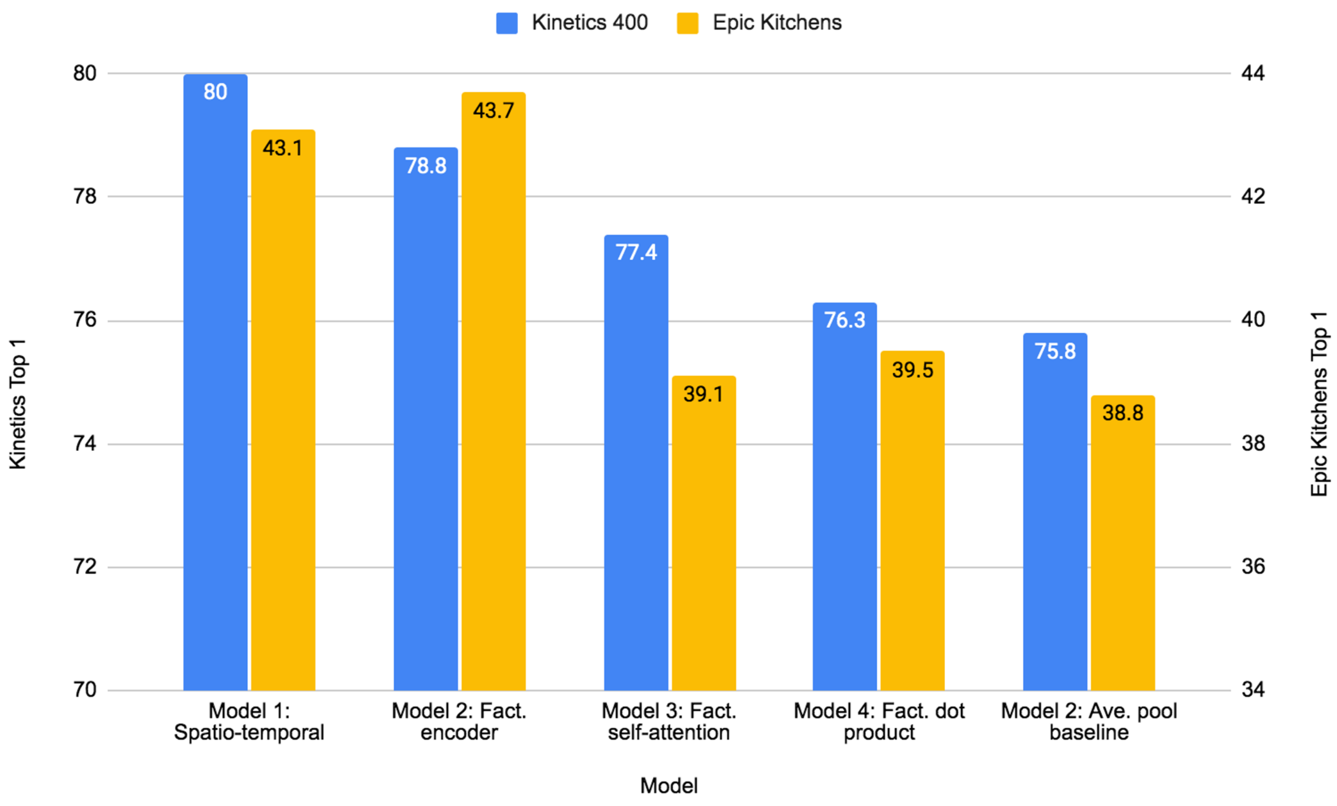
- Tokens fixed across models
- Unfactorised model works best on larger datasets (ie Kinetics), but slowest.



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 at time

(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 at time.

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

Multiview Transformers for Video Recognition

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

CVPR 2022

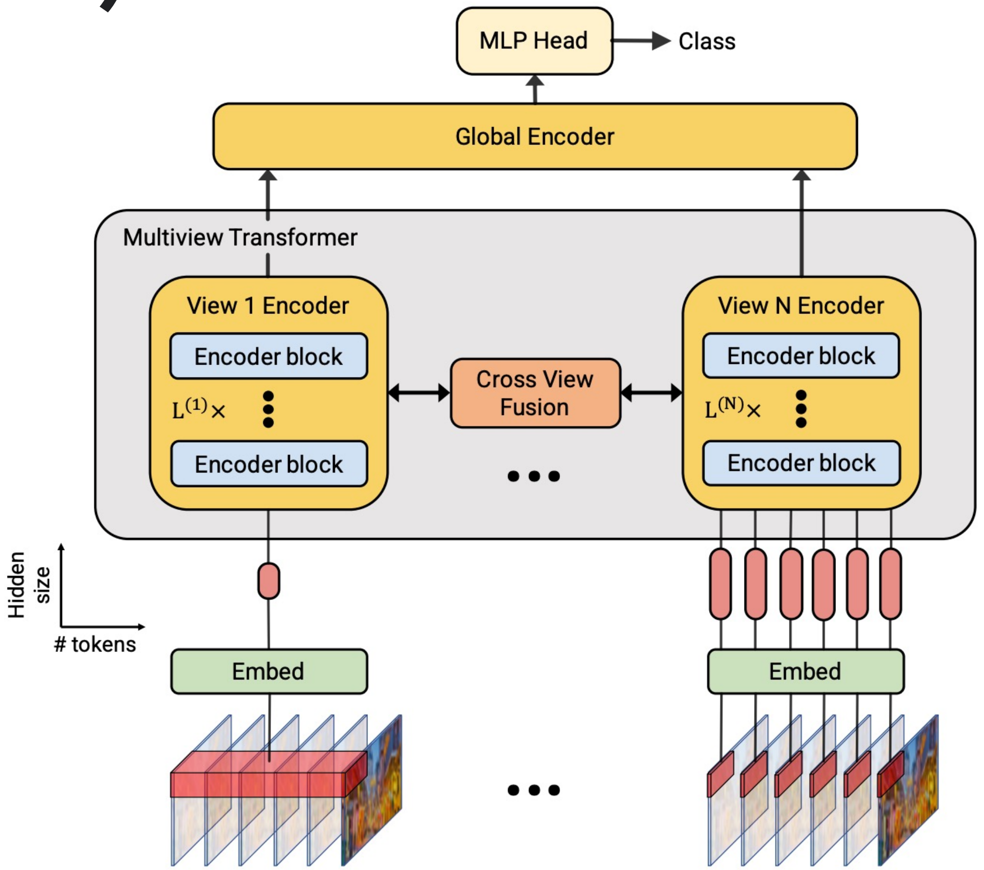
Google Research

Motivation

- Transformers have a “global receptive field” and model long-range interactions.
- 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?

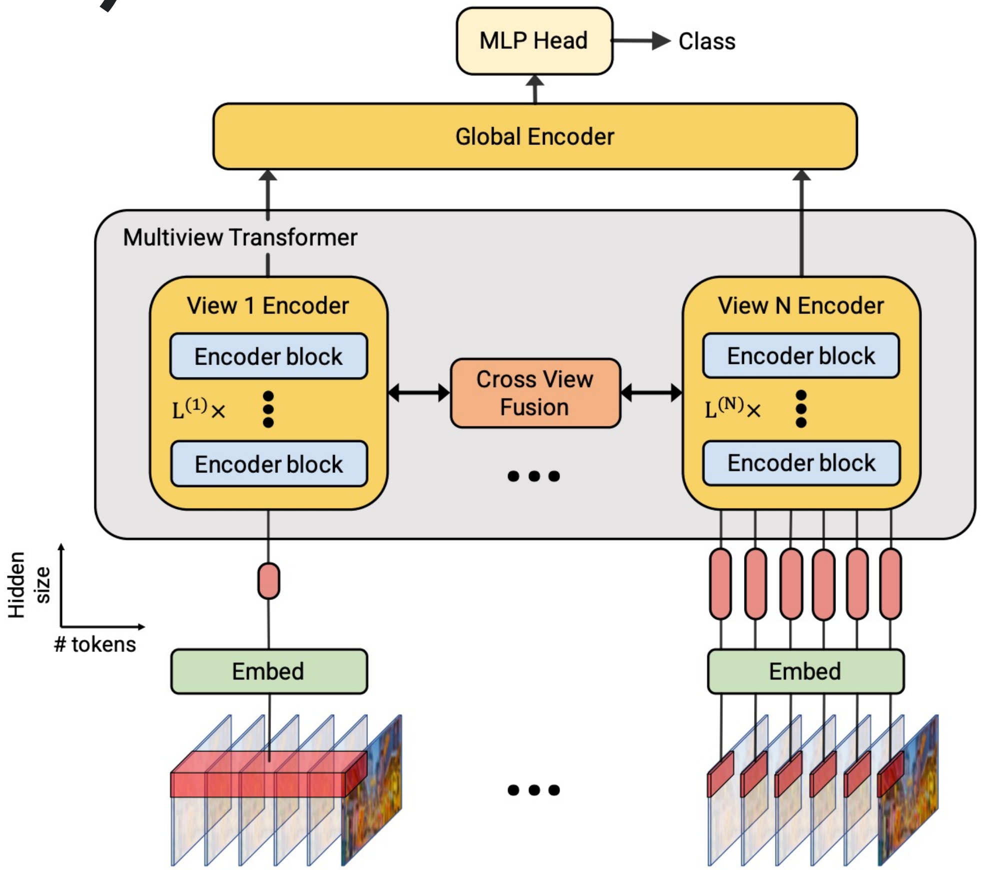
Multiview Transformers (MTV)

- 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.
- Views are constructed by tokenisations of the same input.



Multiview Transformers (MTV)

- Views are constructed by tokenisations of the same input.
- View with small tubelets
  - Many tokens
  - Fine temporal details
- View with large tubelets
  - Few tokens
  - Overall context of the scene.

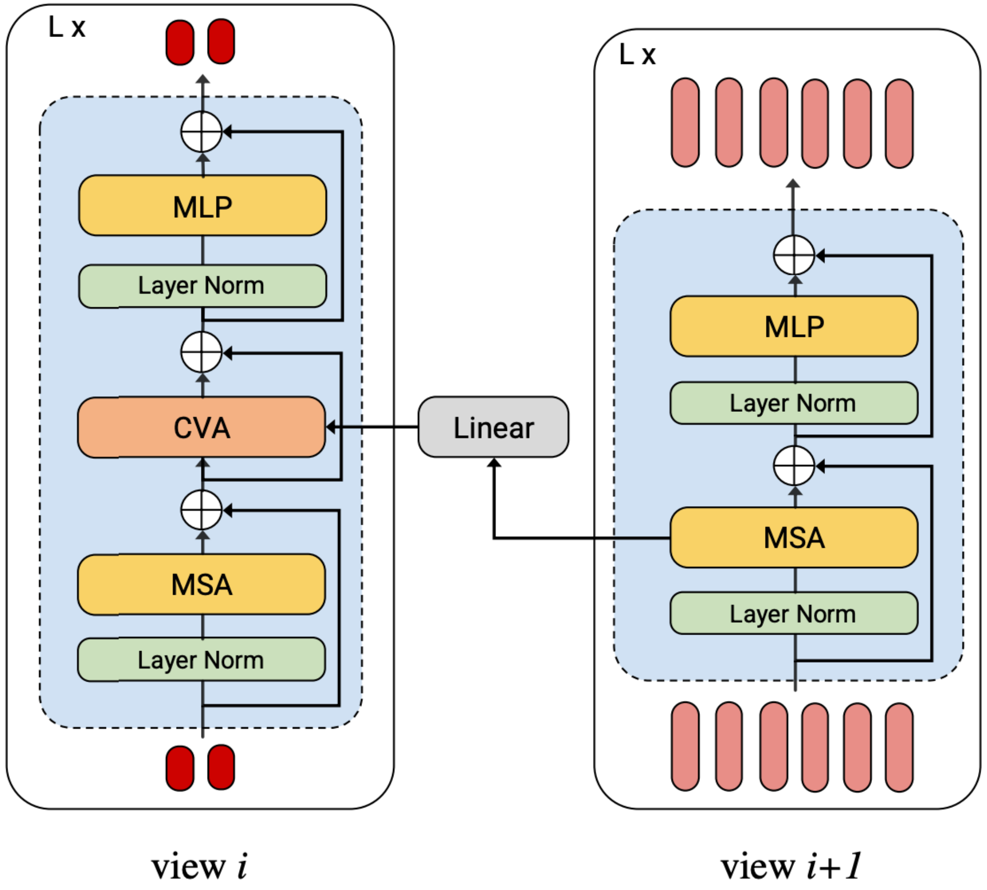


Multiview Transformers

- 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		145	173	78.3
S/8	N/A	20	60	74.1
Ti/16		3	13	67.6
	Ensemble	168	246	77.7
	Late fusion	187	306	80.6
B/4+S/8+Ti/16	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

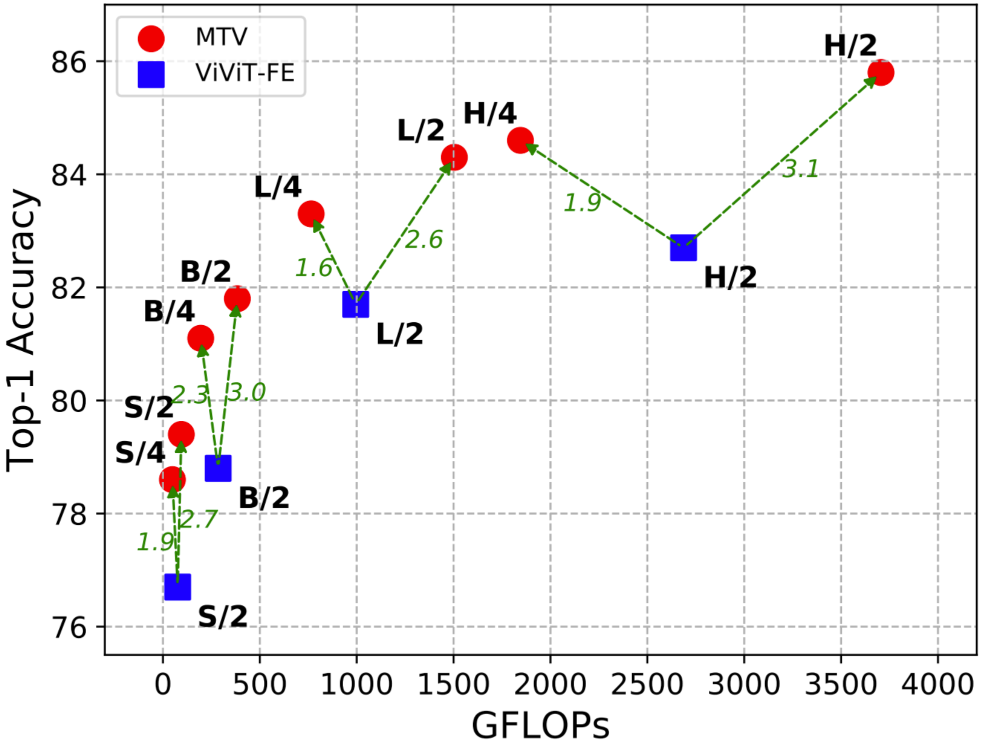
What encoder should we use for each view?

- 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

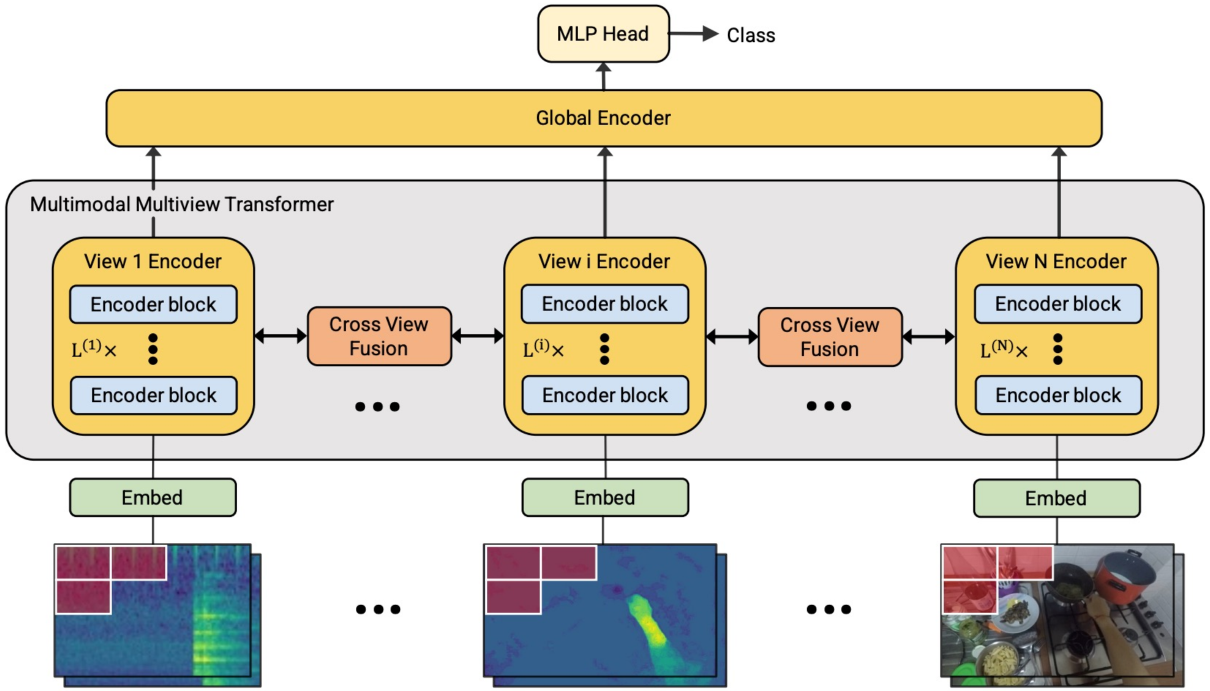
Method	Action	Verb	Noun
ViViT-L FE [3]	44.0	66.4	56.8
MFormer-HR [51]	44.5	67.0	58.5
MoViNet-A6 [35]	47.7	**72.2**	57.3
MTV-B	46.7	67.8	**60.5**
MTV-B (320p)	**48.6**	68.0	**63.1**
MTV-B (WTS 280p)	**50.5**	69.9	**63.9**

(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
  - Audio as spectrograms
  - Optical flow



Multimodal MTV

- Use the deepest encoder for RGB – the most discriminative modality.
- Won this year's Epic Kitchens Action Recognition challenge.

View 1	View 2	View 3	Accuracy
Base: RGB	Small: RGB	Tiny: RGB	52.7
Base: RGB	Small: Audio	Tiny: RGB	53.4
Base: RGB	Small: Flow	Tiny: RGB	53.2
Base: RGB	Small: Audio	Tiny: Flow	**53.6**

Conclusion

- Processing multiple “views” in parallel allows us to achieve superior accuracy-speed trade-offs for video classification.
- Easy to extend this to leverage multiple modalities.
- State-of-the-art results across 6 datasets ; winner of Epic Kitchens challenge.

- [Paper], [Epic Kitchens challenge], [Code]

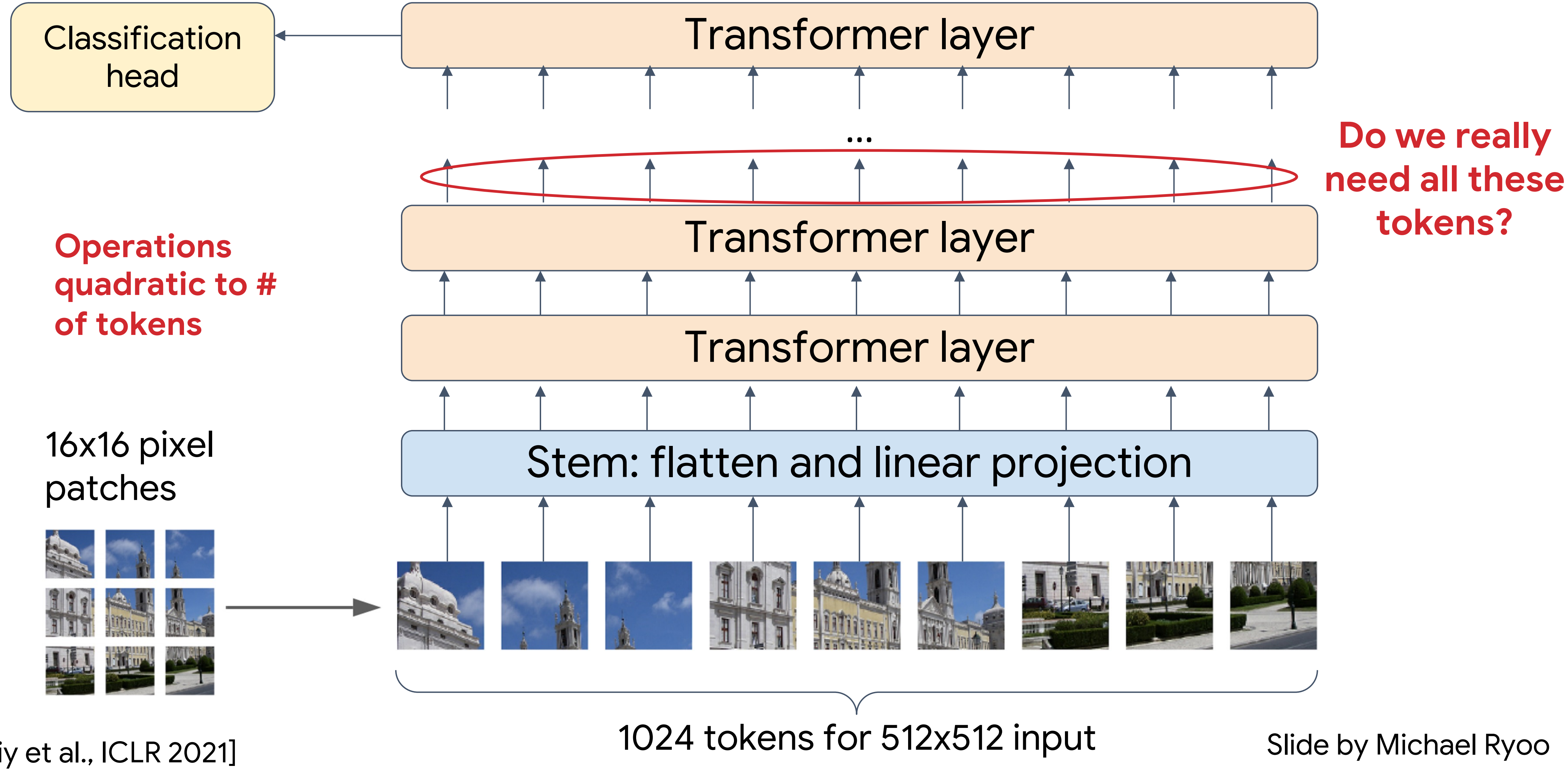
TokenLearner: What Can 8 Learned Tokens do for Images and Video

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

NeurIPS 2021

Google Research

Vision Transformers



[Dosovitskiy et al., ICLR 2021]

Slide by Michael Ryoo

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

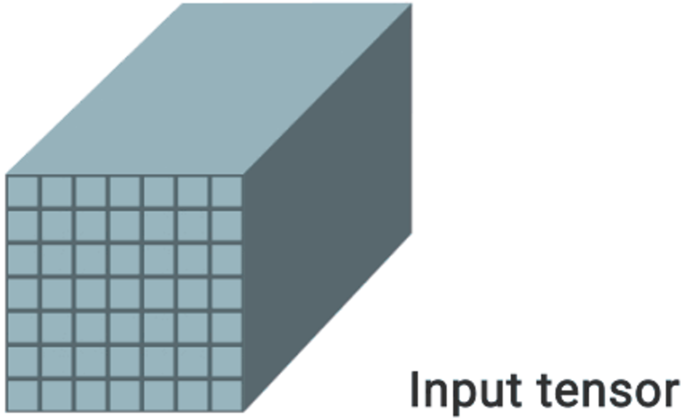
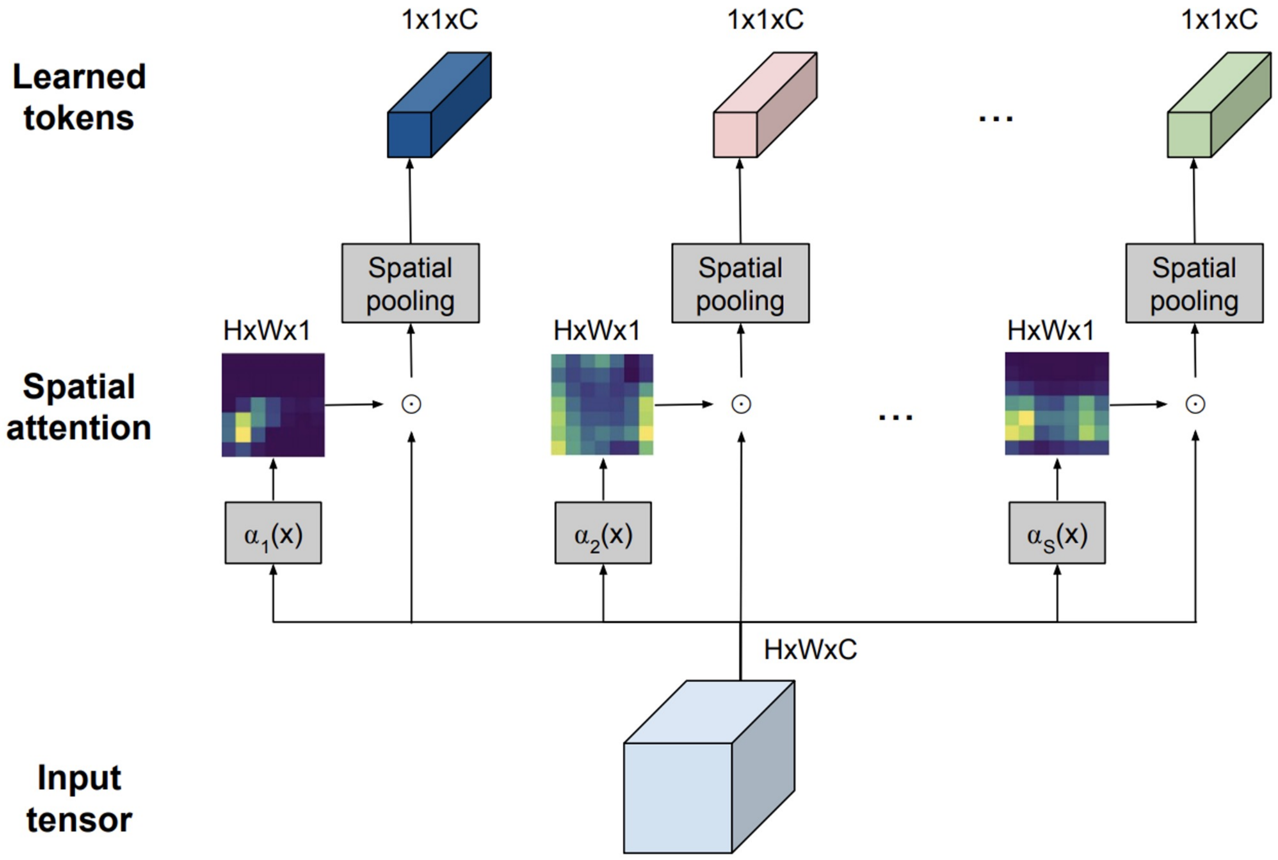


Figure by Tom Small

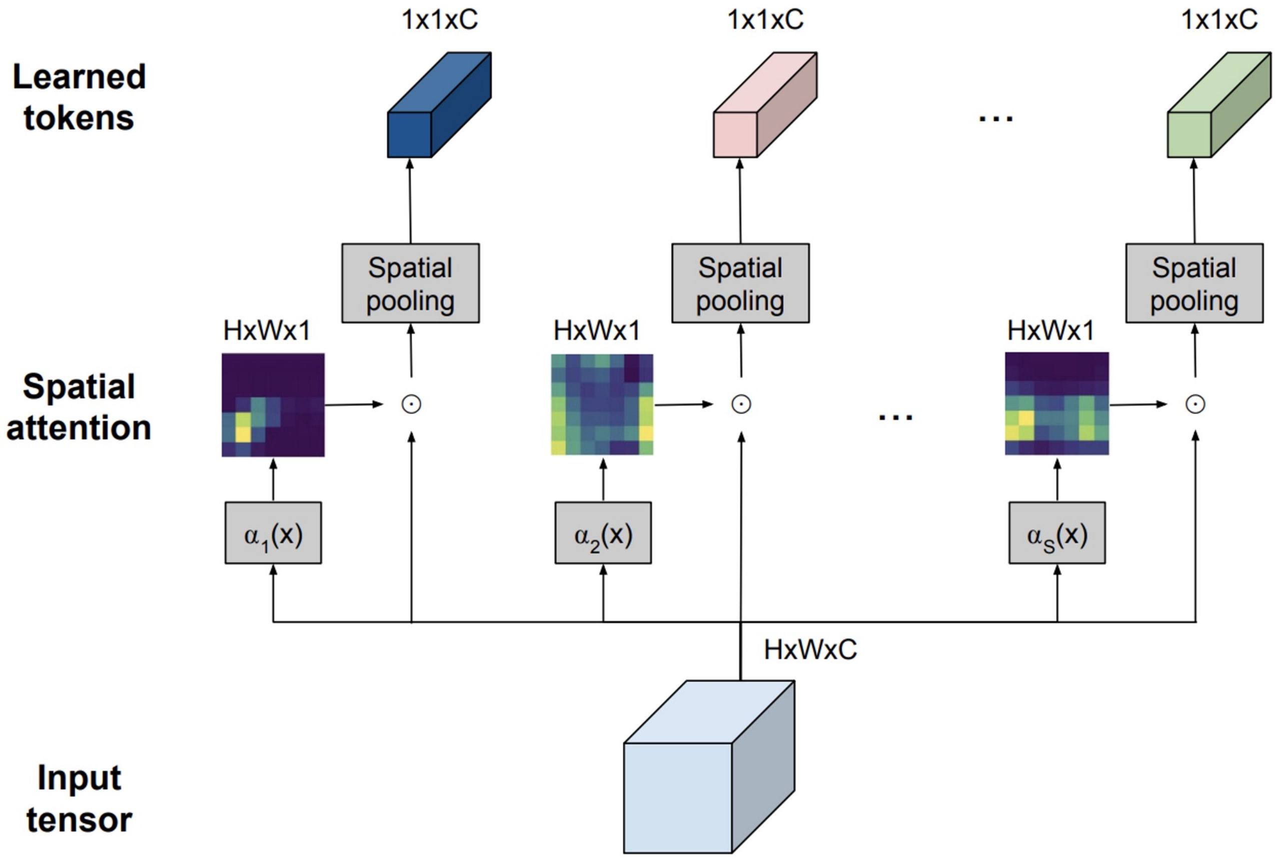
TokenLearner

- TokenLearner is 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



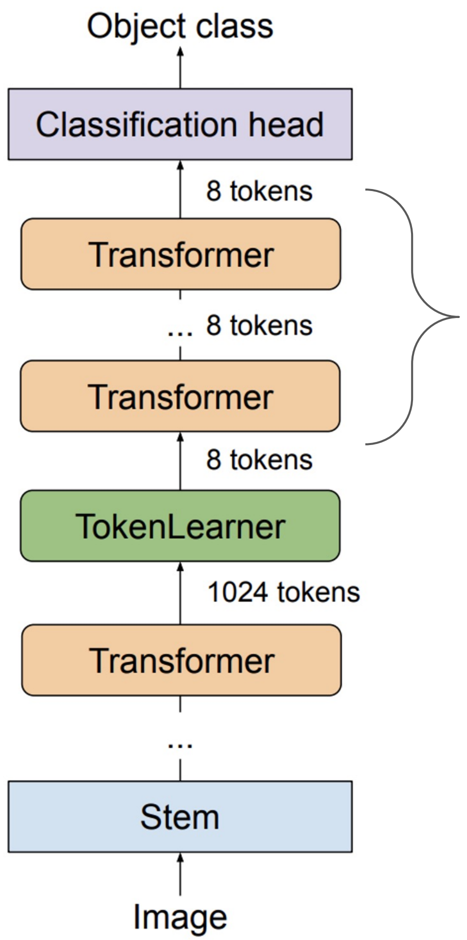
TokenLearner

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



TokenLearner within ViT

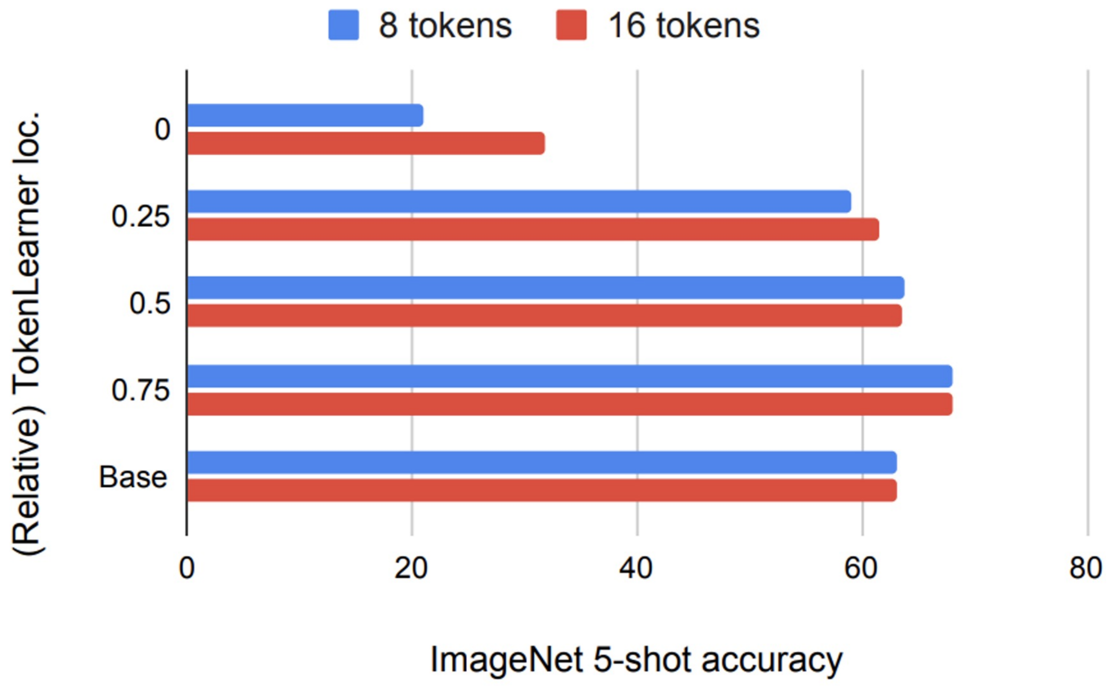
- TokenLearner module inserted in the middle of Transformer architecture
- The computation after the TokenLearner module becomes negligible.

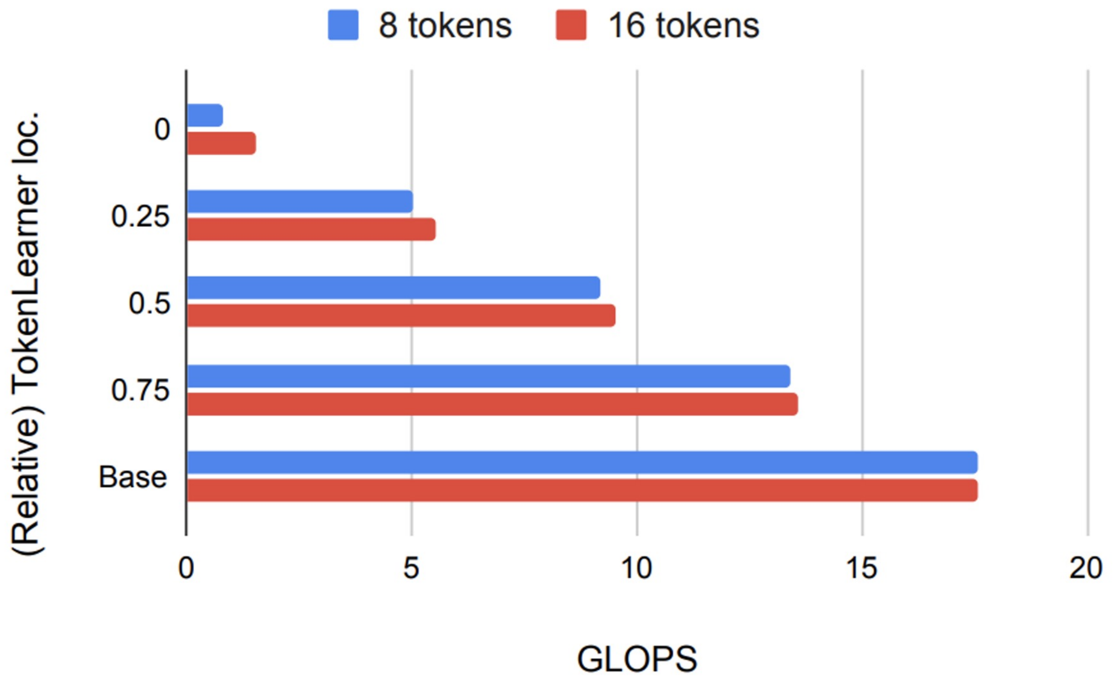


Computation negligible

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

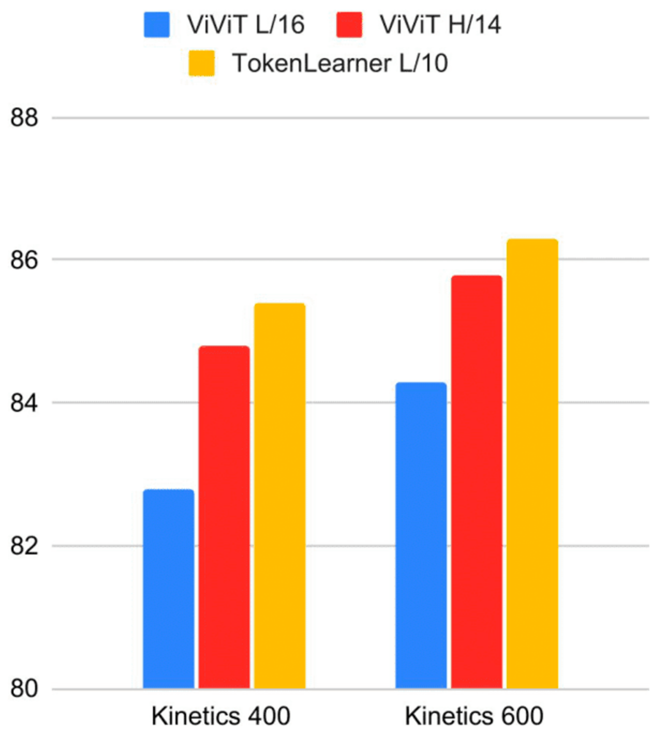
- By using TokenLearner, we can now
  - Process more initial tokens (use smaller patch sizes)
  - Use more transformer layers.
- Results using ViT-L with 512x512 inputs, and 16 learned tokens.

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

Audiovisual Masked Autoencoders

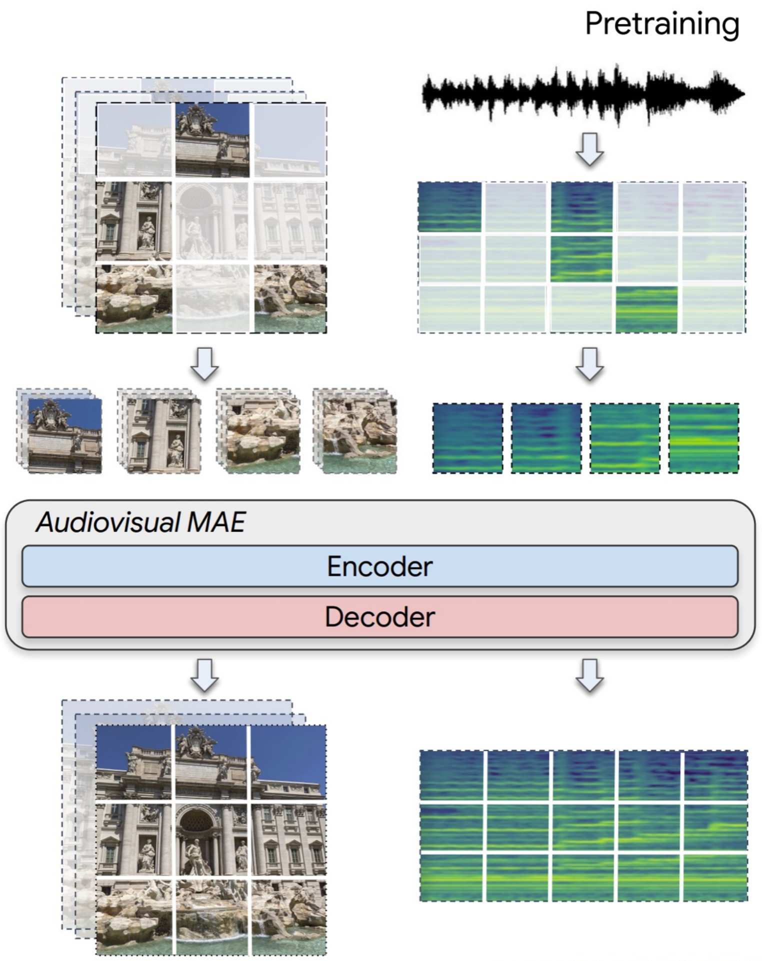
Lili Georgescu, Eduardo Fonseca,
Radu Ionescu, Mario Lucic, Cordelia Schmid,
Anurag Arnab

Google Research

Introduction

- Video models rely on pretrained image models for initialisation
- Masked Autoencoders present a self-supervised alternative

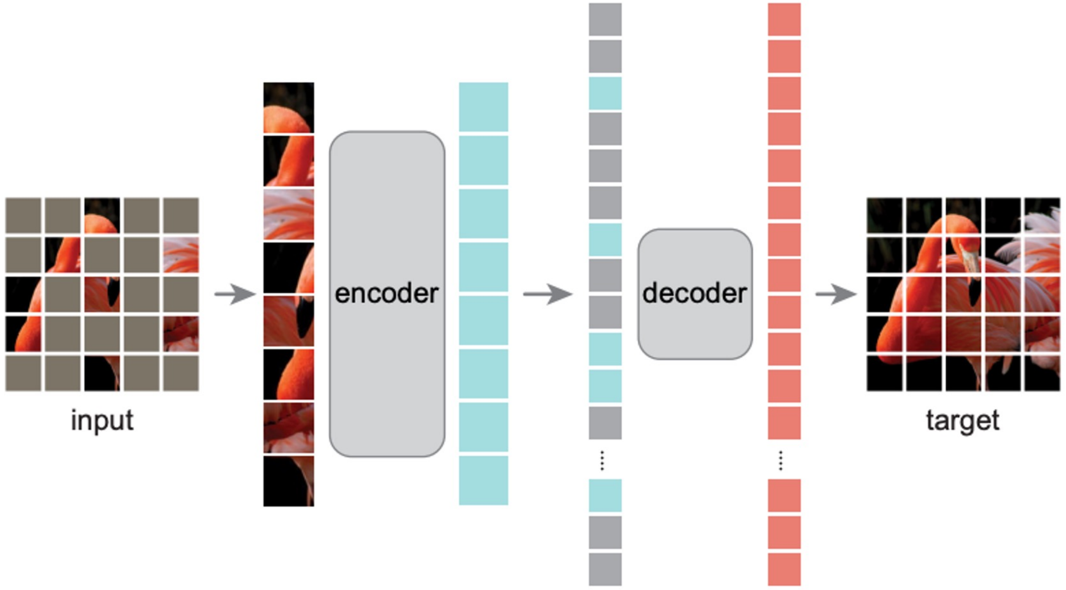
- Can we leverage multiple modalities for stronger representation learning?
  - For multimodal downstream tasks?
  - For unimodal downstream tasks?



Google Research

Masked Autoencoders

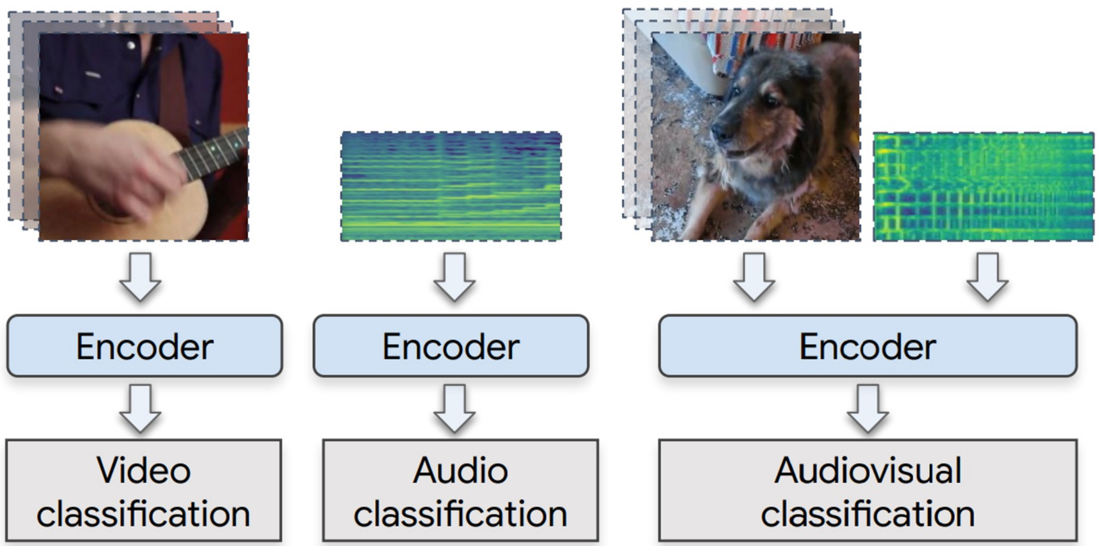
- Tokenise the input
- Remove $\alpha\%$ of the tokens
- Encode these unmasked tokens.
- Add mask tokens back into the sequence.
- Decode the tokens, and reconstruct the original inputs.
- Inspired by BERT for NLP.



Google Research

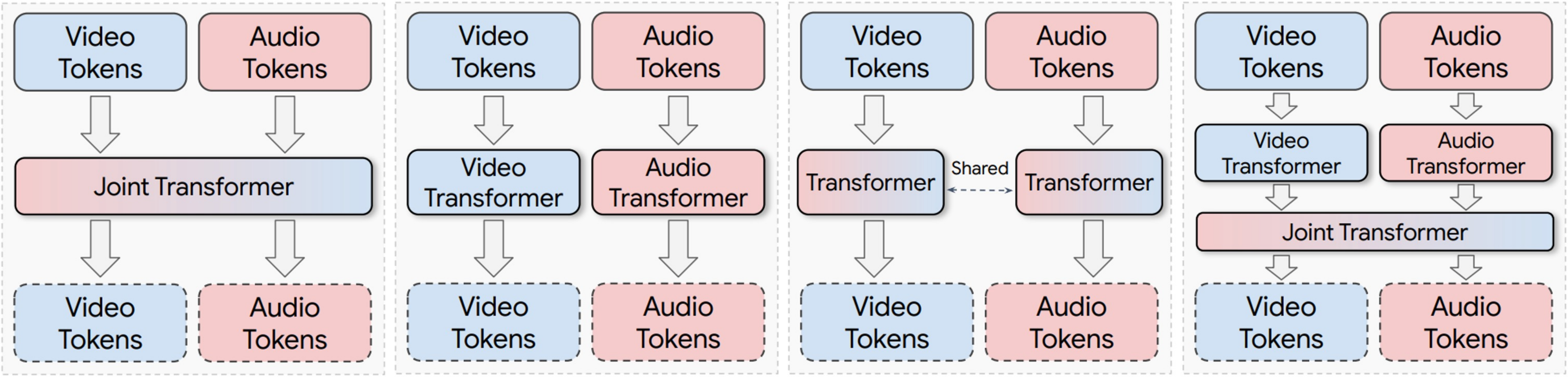
Masked Autoencoders

- Representation is learned by the encoder.
- After pretraining, we discard the decoder, and finetune the encoder on downstream tasks.



Architecture

- Different alternatives for combining audio and visual information at different stages of the encoder and the decoder.
- Early-, mid- or late-fusion. Parameter sharing instead.

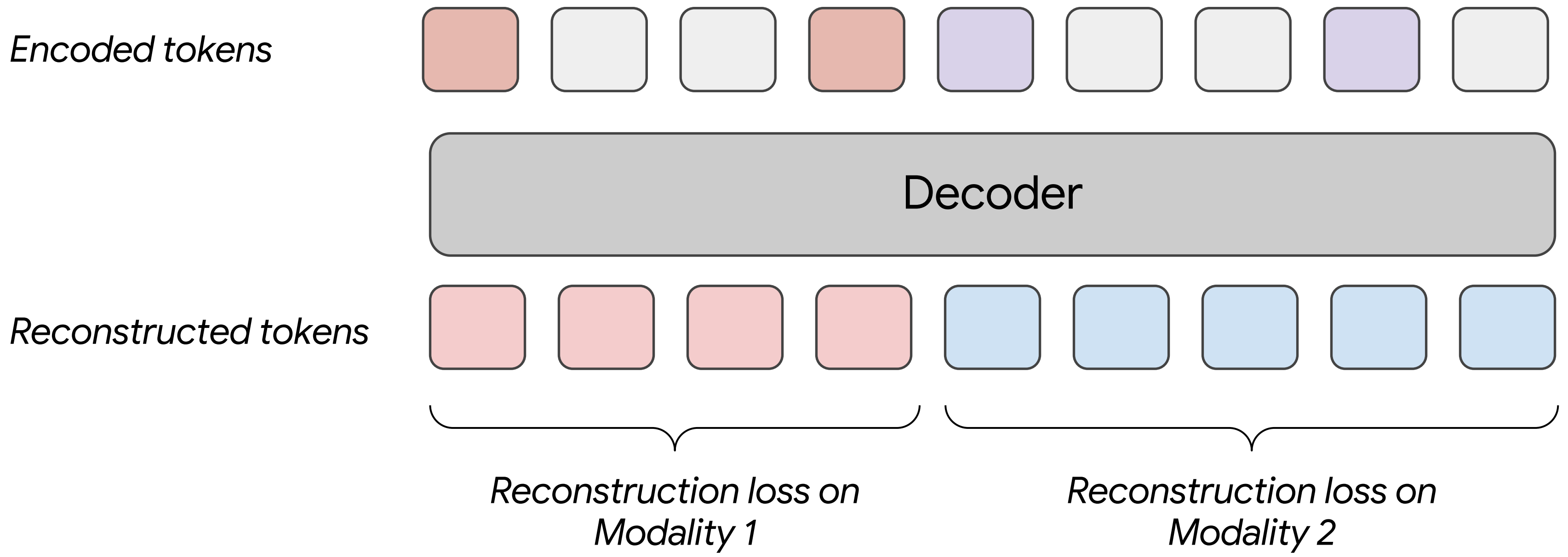


a) "Early fusion" b) Separate c) Shared d) Mid-fusion

Google Research

Reconstruction Objective

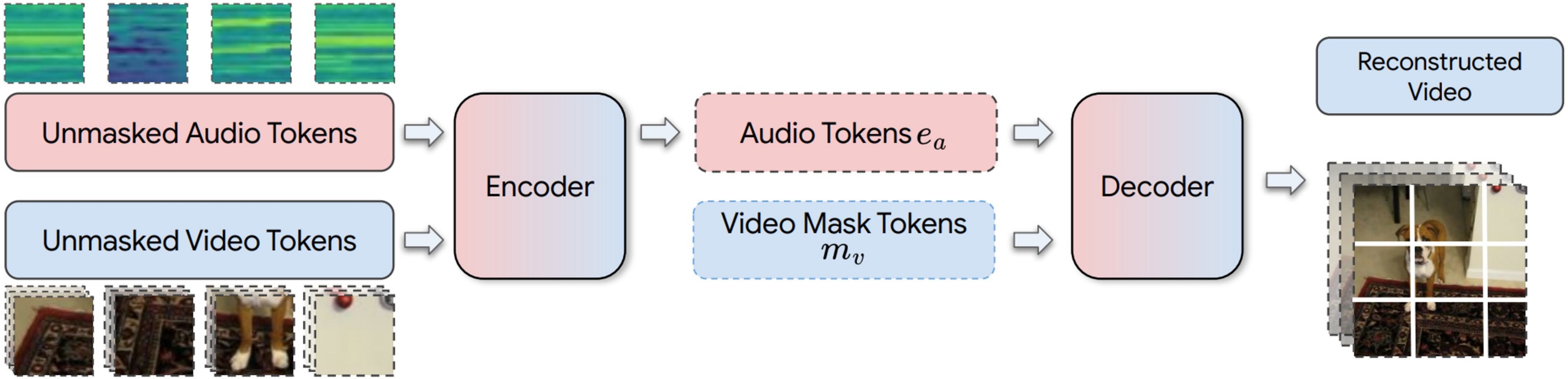
1. Joint Reconstruction
    - Simply encode both modalities and reconstruct both modalities.
    - Equal loss weights on each modality.
    - Normal MAE training, but with more tokens from more modalities.



Google Research

Modality Inpainting

- Reconstruct audio tokens from encoded video tokens and audio mask tokens (and vice versa)
- Requires video tokens to encode the audio to be able to reconstruct the audio from video alone.



Datasets for experiments

- VGGSound
  - 200K examples. Object making the sound is always present in the video. Videos from YouTube.
- AudioSet
  - 2M examples. Videos from YouTube. Weaker correlation between audio and video
- Epic Kitchens
  - 80K examples.
  - Egocentric videos from head-mounted cameras. For evaluating transfer performance, as it presents a challenging domain shift.

Which architecture?

- “Separate” and “Mid-fusion” consistently best for the encoder
- Encoding strategy matters for audiovisual tasks.
- Weight-sharing in the decoder is consistently better.
- Experiments on VGGSound

Encoder	Decoder	Audio-only	Video-only	Audiovisual
Early fusion	Shared	55.5	46.5	62.2
Early fusion	Separate	55.7	43.6	61.1
Separate	Shared	55.4	48.9	63.0
Shared	Separate	55.4	45.9	61.3
Mid-fusion	Shared	55.8	48.5	63.5
Mid-fusion	Early	55.5	48.5	63.3

Which objective?

- The vanilla joint reconstruction performs the best
- Modality inpainting is harder to train

Objective	Audio-only	Video-only	Audiovisual
Joint reconstruction	55.5	46.5	62.2
Inpainting (video from audio)	51.5	39.9	58.4
Inpainting (audio from video)	52.5	38.1	58.2
Inpainting (both modalities)	54.1	38.6	58.4

What about training separate MAEs?

- An alternative is to train separate unimodal MAEs
- Audiovisual MAE improves substantially for audiovisual finetuning
- On par for audio-only or video-only finetuning
- Means we can pretrain a single model, and use for different downstream tasks

Pretraining	Audio only	Video only	Audiovisual
AudioMAE	55.7	42.1	58.3
VideoMAE	52.8	49.3	62.1
Audiovisual MAE	55.8	48.5	63.5

Iterations matter more than the dataset size

- AudioSet is 10x the size of VGGSound
- But when we pretrain on both datasets for the same number of iterations, performance is similar.

Pretrain \ Finetune	VGGSound	AudioSet	Epic Kitchens
VGGSound	**65.0**	51.2	**45.5**
AudioSet	64.7	**51.3**	43.5

More iterations are consistently better

- Accuracy consistently improves as we pretrain for longer
- We always pretrain on VGGSound, and accuracy plateaus when finetuning on VGGSound
- But we continually improve when transferring on Epic Kitchens

Epochs	200	400	800	1200
VGGSound	63.2	63.9	**65.0**	64.9
Epic Kitchens	41.8	42.5	45.5	**46.0**

Comparison to state-of-the-art

- Our model is a simple. Encoder is
  - Standard vision transformer for single-modal tasks
  - MBT for multimodal tasks
- Other methods use modality-specific architectures
- We only perform self-supervised pretraining.
  - Other methods use supervised pretraining on multiple datasets.
- Can still achieve state-of-the-art results
- Shows promise of self-supervised pretraining instead of supervised.

Comparison to state-of-the-art

(a) VGGSound. We report Top-1 accuracy.

Epochs	Pretraining	A	V	AV
Kazakos *et al.* [42]	Sup. Im1K	52.5	–	–
PlayItBack [63]	Sup. Im21K	53.7	–	–
PolyViT [45]	Sup. Im21K, AS	55.1	–	–
MBT [49]	Sup. Im21K	52.3	**51.2**	64.1
Ours	SSL VGGSound	**57.2**	50.3	**65.0**

(b) AudioSet. We report the mAP for audiovisual fusion models.

Epochs	Pretraining	Training set	A	V	AV
GBlend [71]	Im1K	AS-2M	32.4	18.8	41.8
Perceiver [40]	None	AS-2M	38.4	25.8	44.2
PerceiverIO [39]	None	AS-2M	–	–	44.9
Fayek *et al.* [24]	Im1K	AS-2M	38.4	25.7	46.2
MBT [49]	Im21K	AS-500K	41.5	**31.3**	49.7
Ours	SSL AS-2M	AS-500K	**45.7**	30.6	**51.3**

(c) Epic Kitchens. We report Top-1 accuracies for verbs, nouns and actions (pairs of verbs and nouns).

		Audio			Video			Audiovisual		
Method	Pretraining	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Damen *et al.* [19]	Sup. Im1K	42.6	22.4	14.5	–	–	–	–	–	–
Kazakos *et al.* [42]	Sup. VGGSound	46.1	23.0	15.2	–	–	–	–	–	–
PlayItBack [63]	Sup. Im21K	47.0	23.1	15.9	–	–	–	–	–	–
TSM [46]	Sup. Im1K + K400	–	–	–	67.9	49.0	38.3	–	–	–
ViViT-L Fact. Encoder [6]	Sup. Im21K + K400	–	–	–	66.4	56.8	44.0	–	–	–
MotionFormer [54]	Sup. Im21K + K400	–	–	–	67.0	58.5	44.5	–	–	–
MTV [74]	Sup. Im21K + K400	–	–	–	67.8	**60.5**	**46.7**	–	–	–
MBT [49]	Sup. Im21K	44.3	22.4	13.0	62.0	56.4	40.7	64.8	**58.0**	43.4
Ours	SSL VGGSound	**52.7**	**27.2**	**19.7**	**70.8**	55.9	45.8	**71.4**	56.4	**46.0**

Comparison to state-of-the-art

(a) VGGSound. We report Top-1 accuracy.

Epochs	Pretraining	A	V	AV
Kazakos *et al*. [42]	Sup. Im1K	52.5	–	–
PlayItBack [63]	Sup. Im21K	53.7	–	–
PolyViT [45]	Sup. Im21K, AS	55.1	–	–
MBT [49]	Sup. Im21K	52.3	**51.2**	64.1
Ours	SSL VGGSound	**57.2**	50.3	**65.0**

(b) AudioSet. We report the mAP for audiovisual fusion models.

Epochs	Pretraining	Training set	A	V	AV
GBlend [71]	Im1K	AS-2M	32.4	18.8	41.8
Perceiver [40]	None	AS-2M	38.4	25.8	44.2
PerceiverIO [39]	None	AS-2M	–	–	44.9
Fayek *et al*. [24]	Im1K	AS-2M	38.4	25.7	46.2
MBT [49]	Im21K	AS-500K	41.5	**31.3**	49.7
Ours	SSL AS-2M	AS-500K	**45.7**	30.6	**51.3**

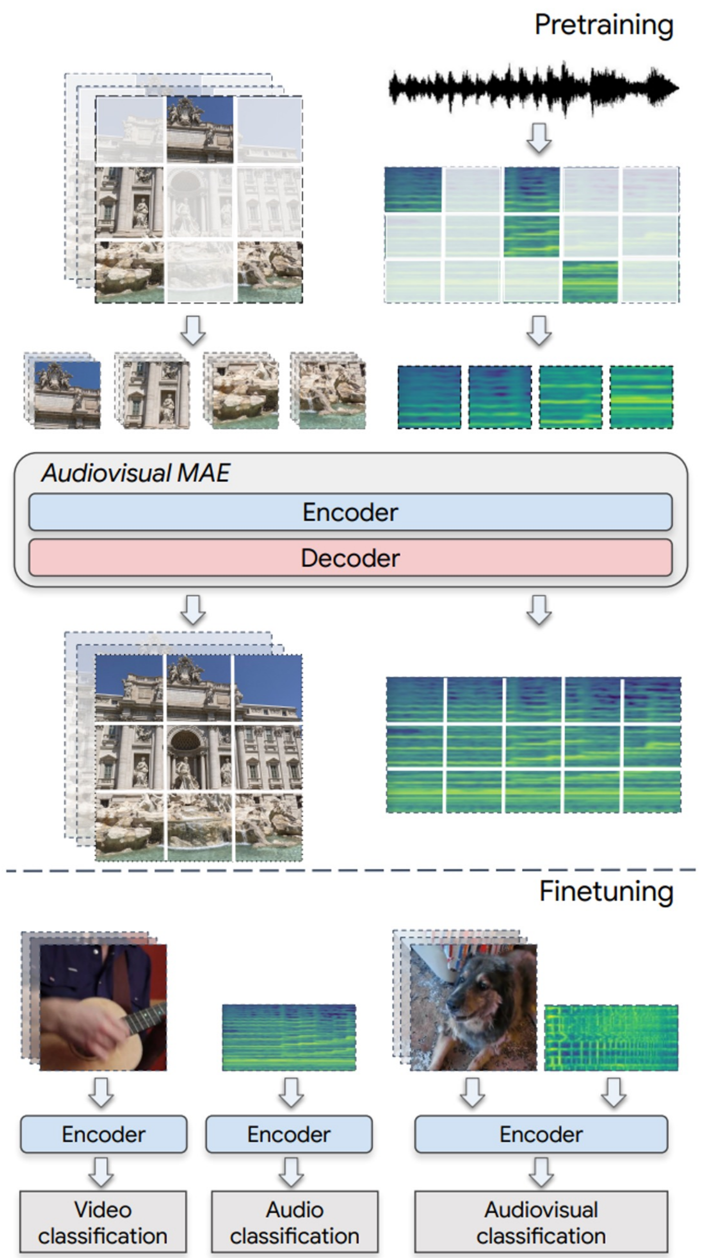
(c) Epic Kitchens. We report Top-1 accuracies for verbs, nouns and actions (pairs of verbs and nouns).

		Audio			Video			Audiovisual		
Method	Pretraining	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Damen *et al*. [19]	Sup. Im1K	42.6	22.4	14.5	–	–	–	–	–	–
Kazakos *et al*. [42]	Sup. VGGSound	46.1	23.0	15.2	–	–	–	–	–	–
PlayItBack [63]	Sup. Im21K	47.0	23.1	15.9	–	–	–	–	–	–
TSM [46]	Sup. Im1K + K400	–	–	–	67.9	49.0	38.3	–	–	–
ViViT-L Fact. Encoder [6]	Sup. Im21K + K400	–	–	–	66.4	56.8	44.0	–	–	–
MotionFormer [54]	Sup. Im21K + K400	–	–	–	67.0	58.5	44.5	–	–	–
MTV [74]	Sup. Im21K + K400	–	–	–	67.8	**60.5**	**46.7**	–	–	–
MBT [49]	Sup. Im21K	44.3	22.4	13.0	62.0	56.4	40.7	64.8	**58.0**	43.4
Ours	SSL VGGSound	**52.7**	**27.2**	**19.7**	**70.8**	55.9	45.8	**71.4**	56.4	**46.0**

Conclusion

- Leverage multiple modalities present in video for pretraining.
- Effective for unimodal and multimodal downstream tasks.

- L Georgescu et al. Audiovisual Masked Autoencoders. *Arxiv* 2022.
- [Paper]



Collaborators

- Shen Yan, Michael Ryoo, Lili Georgescu, Xuehan Xiong, Zhichao Lu, Mi Zhang, Chen Sun, Cordelia Schmid, Mario Lucic, Mostafa Dehghani, Georg Heigold, AJ Piergiovanni, Anelia Angelova

Google BROWN Stony Brook University MICHIGAN STATE UNIVERSITY UNIVERSITY OF BUCHAREST

Questions?

- A Arnab et al. ViViT: A Video Vision Transformer. *ICCV* 2021.
- S Yan et al. Multiview Transformers for Video Recognition. *CVPR* 2022.
- X Xiong et al. M&M Mix: A Multimodal Multiview Transformer Ensemble. *arXiv* 2022
- M Ryoo et al. TokenLearner: What Can 8 Learned Tokens Do for Images and Video. *NeurIPS* 2021.
- L Georgescu et al. Audiovisual Masked Autoencoders. *arXiv* 2022

Google Research