ViViT : A Video Vision Transformer

OVERVIEW

- Introduction
- Related Work
- Embeddings
- Models Proposed
- Ablation Study
- Conclusion

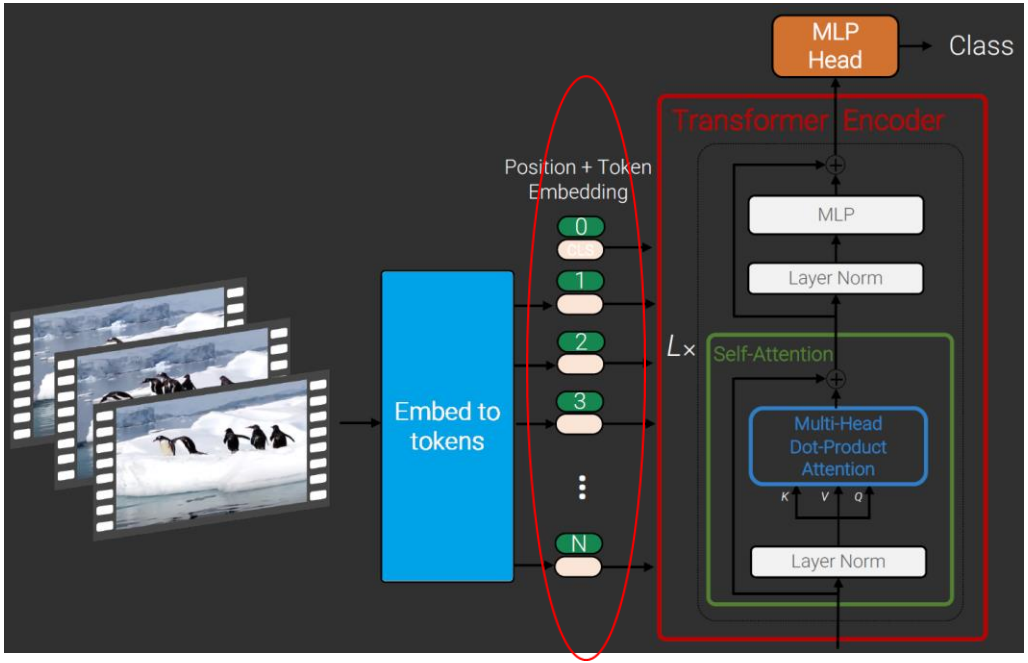
Introduction

- Transformers: Just a tool for NLP? NO
- Local vs Global Context
- Spatial & Temporal Dependencies

Introduction contd.

Video Vision Transformer (ViViT)

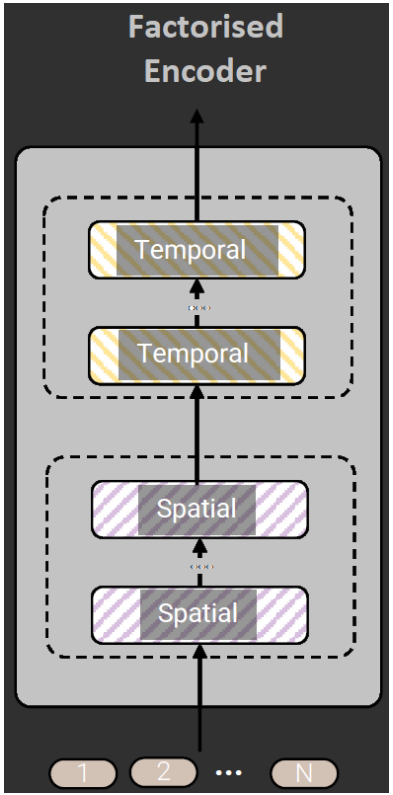
- Basic Idea
- Issue? Lot of tokens

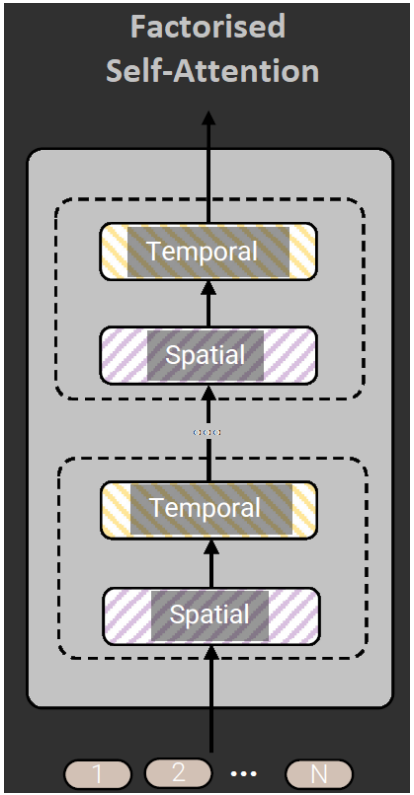


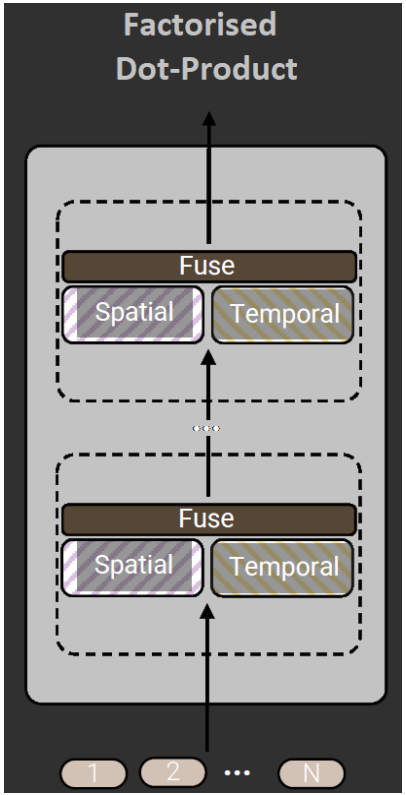
Related Work

- NLP transformer to image applications
- Joint Space Time Model
- Factorization of Spatial & Temporal domain

ViViT cont. (Variants)







The sequence of tokens input to the following transformer encoder is:

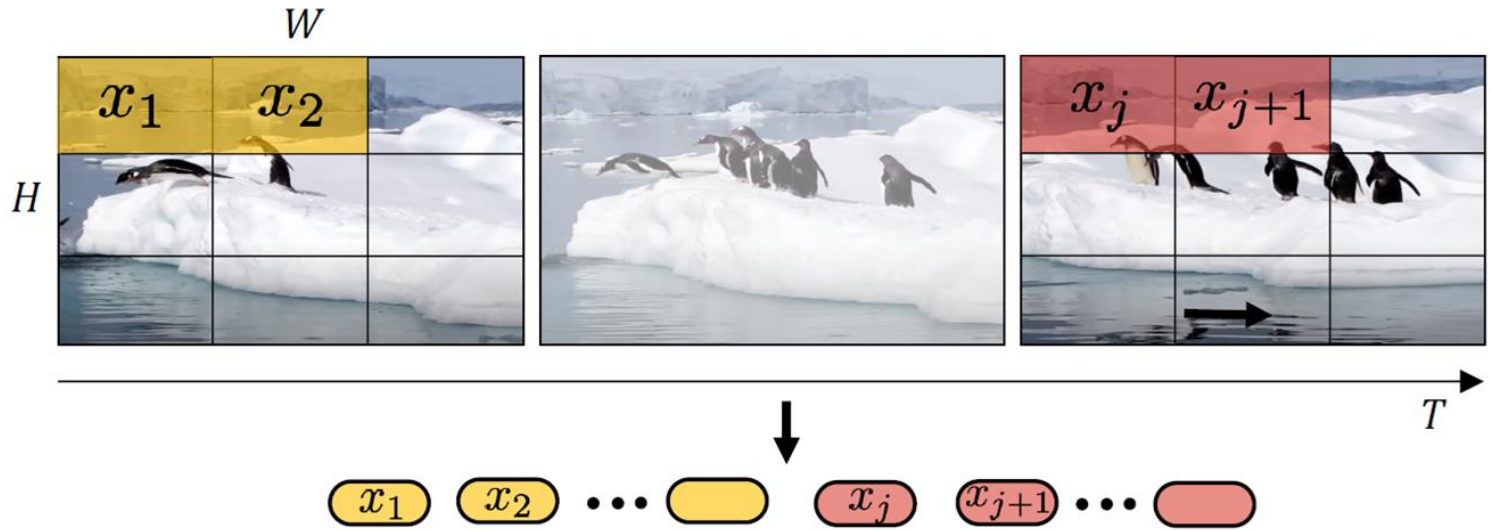
$$\mathbf{z} = [z_{cls}, \mathbf{E}x_1, \mathbf{E}x_2, \ldots, \mathbf{E}x_N] + \mathbf{p}$$

MLP blocks as follows:

$$\mathbf{y}^{\ell} = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}^{\ell})) + \mathbf{z}^{\ell}$$
$$\mathbf{z}^{\ell+1} = \mathrm{MLP}(\mathrm{LN}(\mathbf{y}^{\ell})) + \mathbf{y}^{\ell}.$$

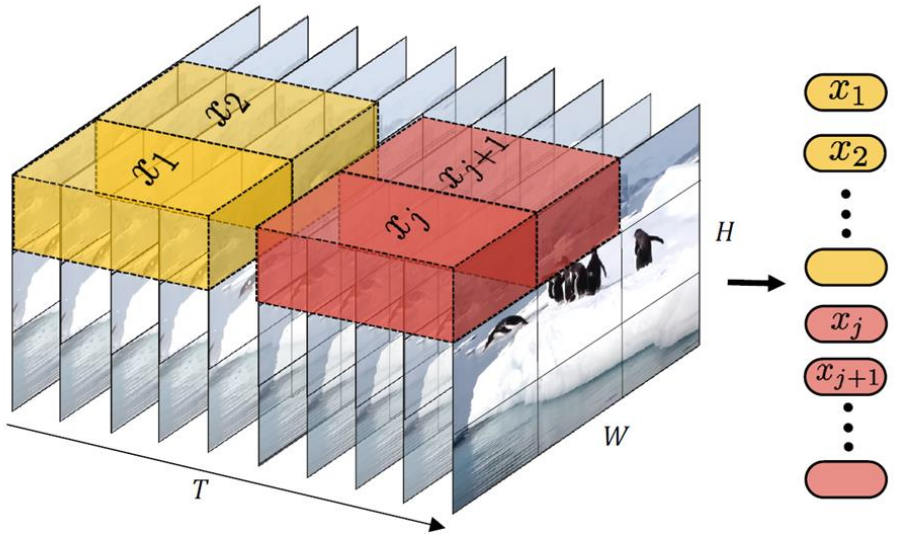
They proposed two simple methods for mapping a video

Uniform frame sampling



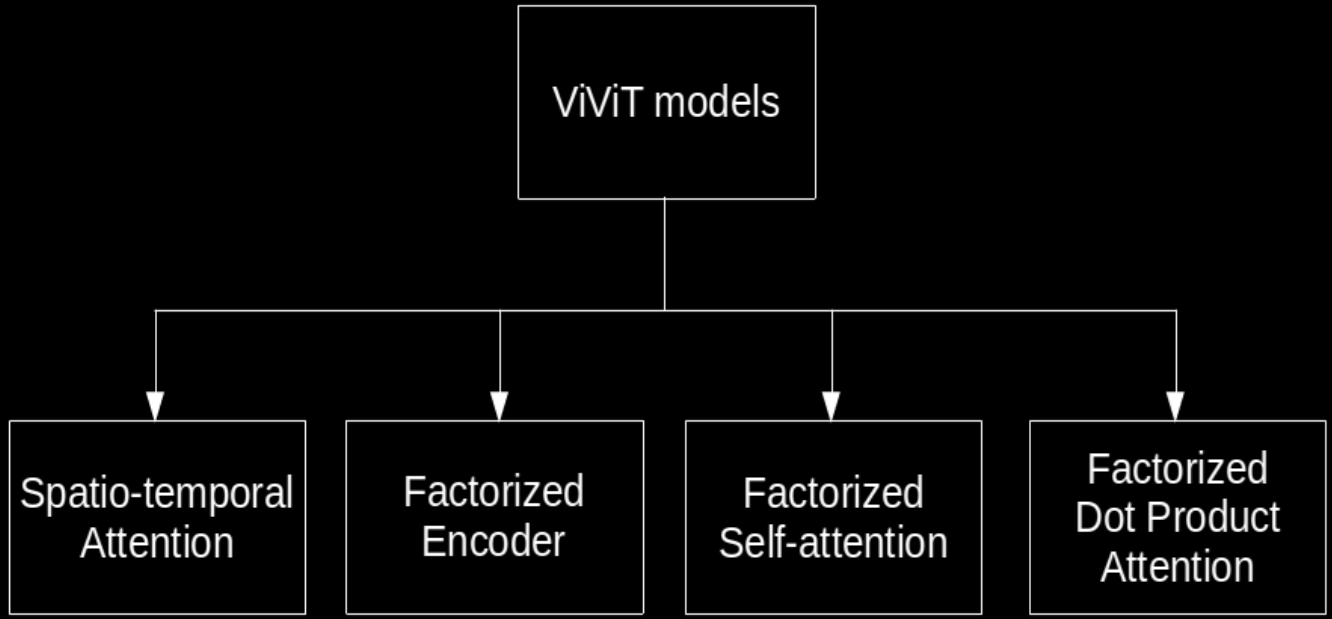
For a tubelet of dimension

- This, fuses spatio-temporal information during tokenisation.
- “Uniform frame sampling” where temporal information from different frames is fused by the transformer.
- For a tubelet of dimension the equation is as given below:



$$t \times h \times w, n_t = \lfloor \frac{T}{t} \rfloor, n_h = \lfloor \frac{H}{h} \rfloor \text{ and } n_w = \lfloor \frac{W}{w} \rfloor$$

Transformer Models for Video

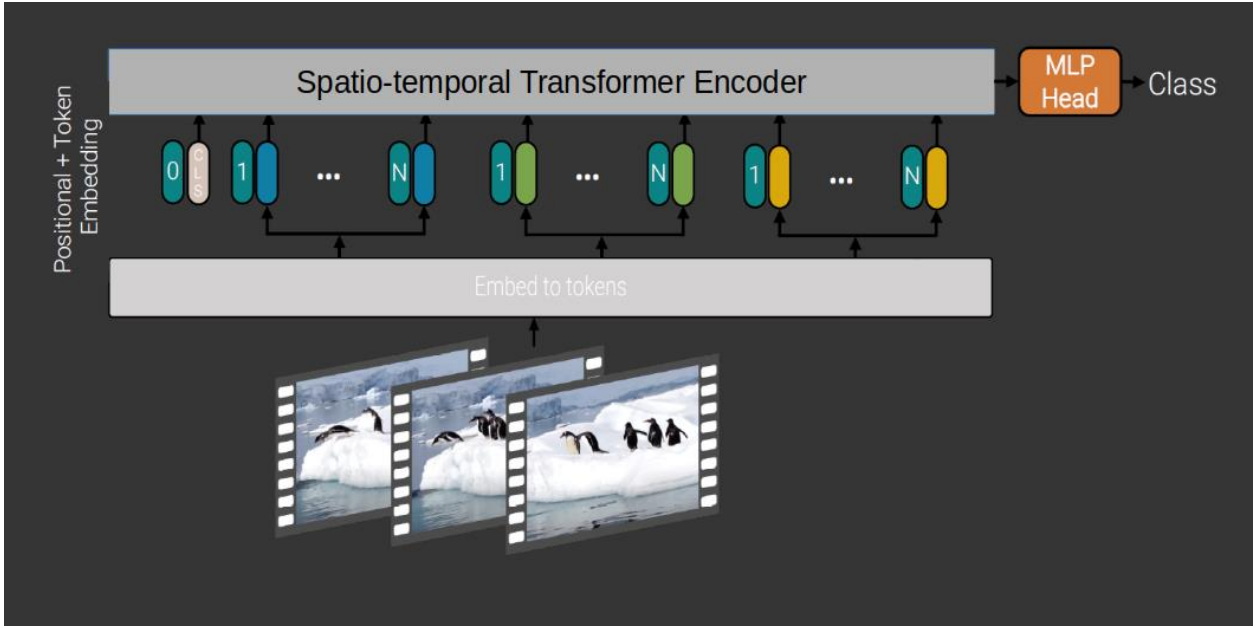


Model 1: Spatio-temporal attention

- simply forwards all spatio-temporal tokens extracted from the video, through the transformer encoder

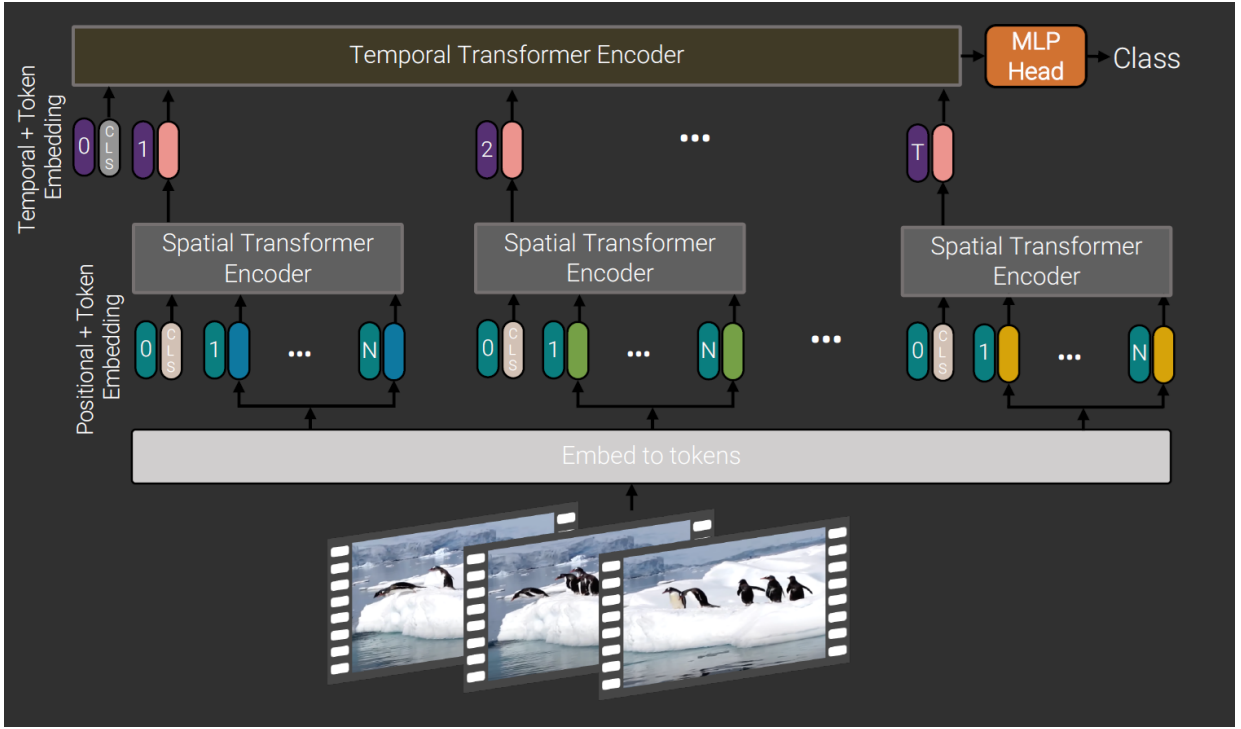
- models long-range interactions across the video from the first layer

- MSA has Quadratic complexity $O((n_t \cdot n_h \cdot n_w)^2)$



Model 2: Factorised encoder

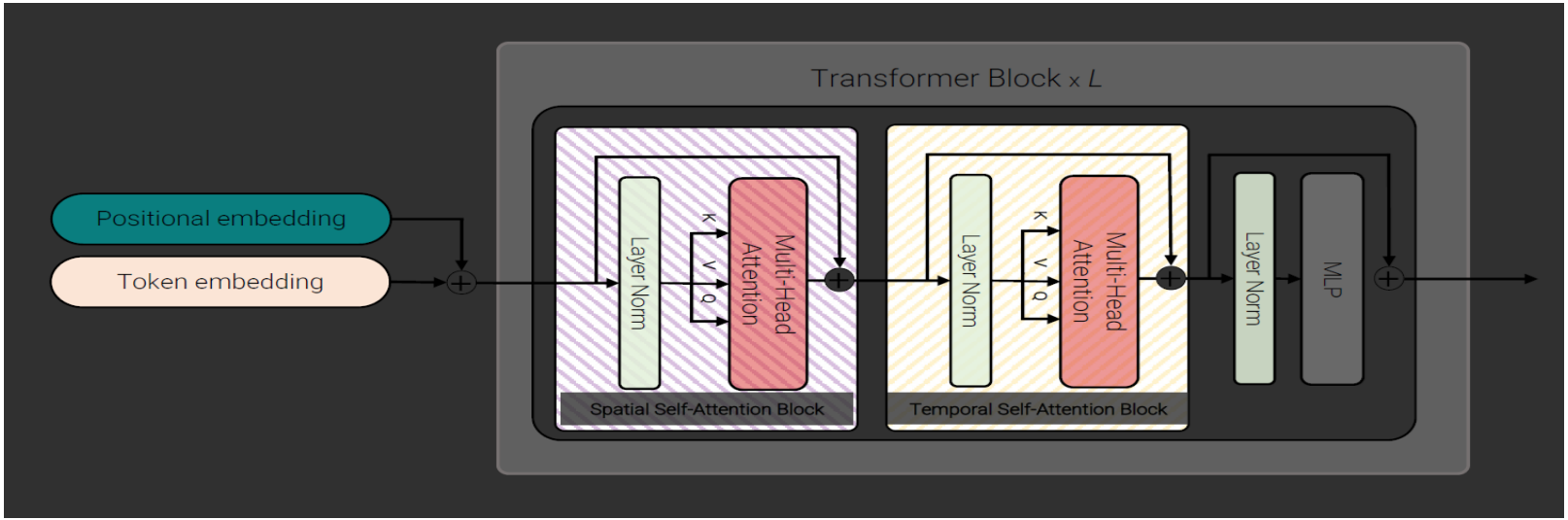
- two separate transformer encoders
- “late fusion” of temporal information
- more parameters, fewer FLOPs
- complexity $O((n_h \cdot n_w)^2 + n_t^2)$



Model 3: Factorised self-attention

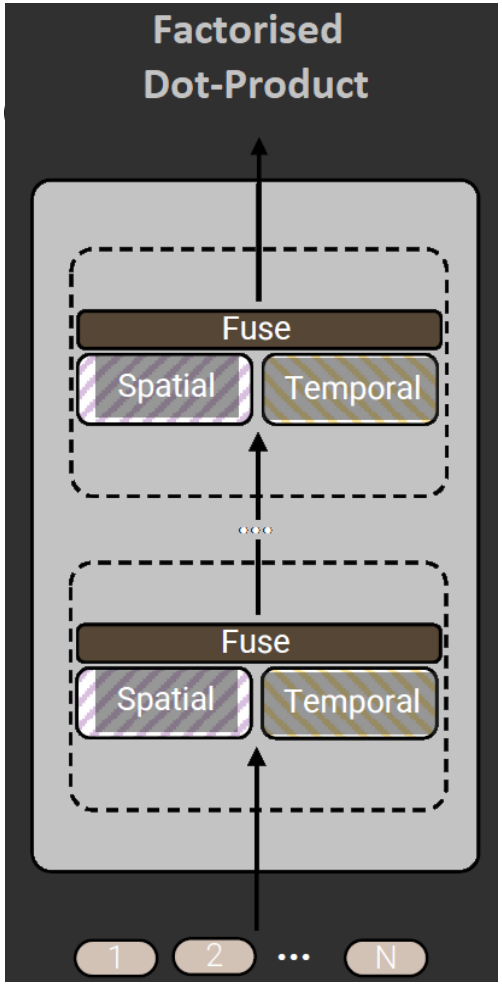
- first only compute self-attention spatially then temporally
- Same number of transformer layers as model 1, same complexity as model 2

$$\mathbf{y}_s^\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_s^\ell)) + \mathbf{z}_s^\ell$$
$$\mathbf{y}_t^\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{y}_s^\ell)) + \mathbf{y}_s^\ell$$
$$\mathbf{z}^{\ell+1} = \mathrm{MLP}(\mathrm{LN}(\mathbf{y}_t^\ell)) + \mathbf{y}_t^\ell$$



Model 4: Factorised dot-product attenti

- same complexity as Models 2 and 3, same number of parameters as unfactorised Model 1
- for half of the attention heads Y_s = Attention(Q,K_s,V_s)
- for the rest Y_t = Attention(Q,K_t,V_t)
- combine the outputs Y = Concat(Y_s,Y_t)W_O



Initialisation by leveraging pretrained models

Positional embeddings :

video model ⇒ nt times more tokens

Solution :

- initialise the positional embeddings by “repeating” them temporally
- all tokens with the same spatial index have the same embedding

Transformer weights for Model 3 :

model 3 ⇒ two MSA modules

Solution :

- initialise the spatial MSA module from the pretrained module
- initialise all weights of the temporal MSA with zeroes

Embedding weights, E :

tubelet embedding ⇒ 3D embedding filter E

but pretrained model has 2D filters

Solutions :

1. Inflating the filters $$\mathbf{E} = \frac{1}{t}[\mathbf{E}_{\text{image}}, \ldots, \mathbf{E}_{\text{image}}, \ldots, \mathbf{E}_{\text{image}}]$$

1. Central Frame Initialisation $$\mathbf{E} = [\mathbf{0}, \ldots, \mathbf{E}_{\text{image}}, \ldots, \mathbf{0}]$$

Emperical Evaluation

Experimental Setup

ViViT-Base/16x2

- Tubelet size 16x16x2

Synchronous SGD

- Multiple GPU Training

Cosine Learning Rate

Pre-train Datasets

- ImageNet-21K (Default)
- JFT

Datasets

Kinetics-400/600

Classes
400/600

Length
10 Sec, 25 FPS

Size
267K/446K



(i) playing violin



(g) riding a bike

https://arxiv.org/pdf/1705.06950.pdf

Epic Kitchens-100

Classes
- A verb + A Noun

Length
- 100 hours

Size
- 90K Clips



Moments in Time

Classes

339 Verbs, YT Clips

Length

3 seconds

Size

Over a Million



https://arxiv.org/pdf/1801.03150.pdf

SSv2

Classes

174*

Length

2-6 seconds

Size

100K*



https://arxiv.org/pdf/1706.04261.pdf

Inference

Input

- 32 Frames of Stride 2
- 4 Views per video

Output

- Average of Per View Logits

Ablation Study

Input Encoding Methods (3D tensor **E)**

Uniform Vs Tubelet Embedding

Model Variants (1 to 4)

Comparison of Unfactorized and Factorized Models

Model Regularization

Varying Tokens

Varying Frames

Which Input Encoding to Best?

ViViT-B / Spatio-Temporal Attention on Kinetics

	Top-1 accuracy
Uniform frame sampling	78.5
Tubelet embedding	
Random initialisation [24]	73.2 +0.7
Filter inflation [8]	77.6
Central frame	79.2

$$\mathbf{E} = [\mathbf{0}, \ldots, \mathbf{E}_{\text{image}}, \ldots, \mathbf{0}].$$

Which Variant is Best?

ViViT Base Backbone/ Tubelet 16 x 2

	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	115.1	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

Which Variant is Best?

- Unfactorized Model performs best on K400
- Factorized Encoder performs best on Epic Kitchens

	K400	EK
Model 1: Spatio-temporal	80.0	43.1
Model 2: Fact. encoder	78.8	43.7
Model 3: Fact. self-attention	77.4	39.1
Model 4: Fact. dot product	76.3	39.5
Model 2: Ave. pool baseline	75.8	38.8

Which Variant is Best?

- Factorized Models use fewer FLOPS than Unfactorized Models
- M2 FE has higher accuracy than M2 Avg Pool with same runtime

	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	115.1	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

Model Regularization

Transformers overfit on smaller data
- Epic Kitchens
- SSv2

Improvements
- 5.3% on Epic Kitchens
- 5.0% on SSv2

Progressive Regularization with ViViT-B/16x2 FE on EC

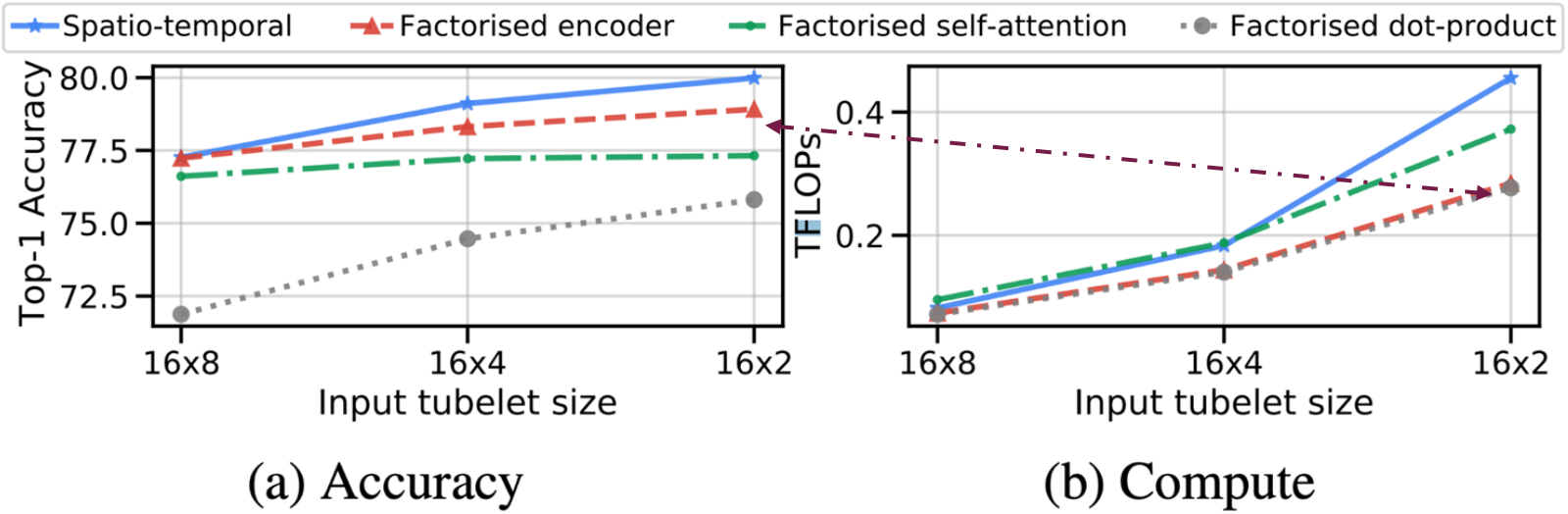
	Top-1 accuracy
Random crop, flip, colour jitter	38.4
+ Kinetics 400 initialisation	39.6
+ Stochastic depth [30]	40.2
+ Random augment [12]	41.1
+ Label smoothing [60]	43.1
+ Mixup [81]	43.7

How does crop size affect performance?

Crop size	224	288	320
Accuracy	80.3	80.7	81.0
GFLOPs	1446	2919 x2	3992 x2.7
Runtime	58.9	147.6 x2.5	238.8 x4

Varying the number of Tokens

As Tubelet Size↓, Tokens↑; Thus, Accuracy↑ and FLOPS↑



(a) Accuracy (b) Compute

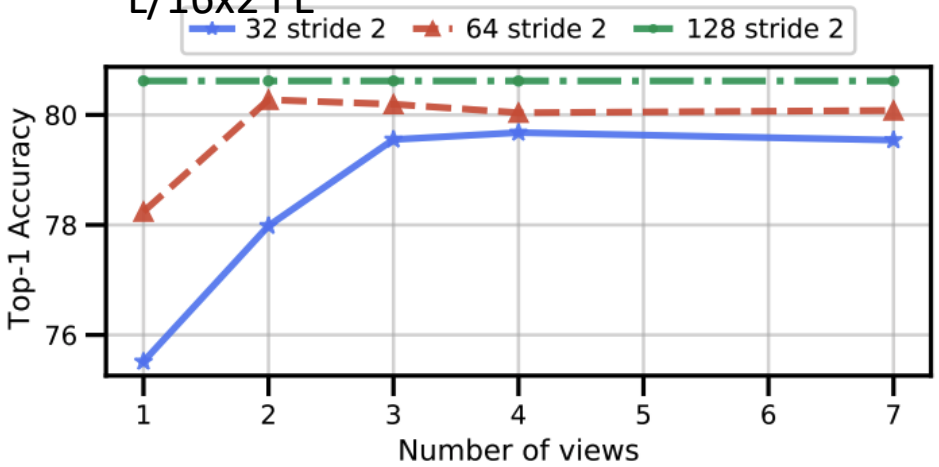
How does number of views affect accuracy?

Multi-View is Better

Saturating Performance with 4 views

128 Frames require single view

Kinetics Dataset - 250 Frames | ViViT-L/16x2 FE



Takeaways: FE(Model-2), large-scale pretraining, strong regularization, multiple views, 16x16x2 tubelet size, one can achieve good performance with not so high computation requirements.

Comparison to the state-of-the-art

While comparing with the current SOTA models, we consider two variants of the model.

- ViViT-L/16x2 FE - Factorised Encoder
- ViViT-H/16x2

Comparison to state-of-the-art : Kinetics-400 Dataset

(a) Kinetics 400

Method	Top 1	Top 5	Views	TFLOPs
X3D-XXL [19]	80.4	94.6	10 × 3	5.82
ViViT-L/16x2 FE	80.6	92.7	1 × 1	3.98
TimeSformer-L [4]	80.7	**94.7**	1 × 3	7.14
ViViT-L/16x2 FE	**81.7**	93.8	1 × 3	11.94
Methods with large-scale pretraining				
ViViT-L/16x2 FE (JFT)	83.5	94.3	1 × 3	11.94
ViViT-H/16x2 (JFT)	**84.9**	**95.8**	4 × 3	47.77

Comparison to state-of-the-art : Kinetics-600 Dataset

(b) Kinetics 600

Method	Top 1	Top 5
X3D-XL [19]	81.9	95.5
TimeSformer-L [4]	82.2	**95.6**
ViViT-L/16x2 FE	**82.9**	94.6
ViViT-L/16x2 FE (JFT)	84.3	94.9
ViViT-H/16x2 (JFT)	**85.8**	**96.5**

Comparison to state-of-the-art : Moments in Time Dataset

(c) Moments in Time

	Top 1	Top 5
AssembleNet-101 [53]	34.3	62.7
ViViT-L/16x2 FE	**38.5**	**64.1**

Comparison to state-of-the-art : Epic Kitchen Dataset

(d) Epic Kitchens 100 Top 1 accuracy

Method	Action	Verb	Noun
TSM [42]	38.3	**67.9**	49.0
SlowFast [20]	38.5	65.6	50.0
ViViT-L/16x2 FE	**44.0**	66.4	**56.8**

Comparison to state-of-the-art : Something-Something V2 Dataset

(e) Something-Something v2

Method	Top 1	Top 5
TSM [42]	63.4	88.5
blVNet [18]	65.2	**90.3**
TimeSformer-HR [4]	62.5	–
ViVIT-L/16x2 FE	**65.9**	89.9

Innovations

- This model presents a unique way of factorising spatial and temporal dimensions which increased efficiency and scalability

- Through thorough ablation analysis, model architecture and regularization methods, authors were able to achieve state-of-the-art results.

Criticism

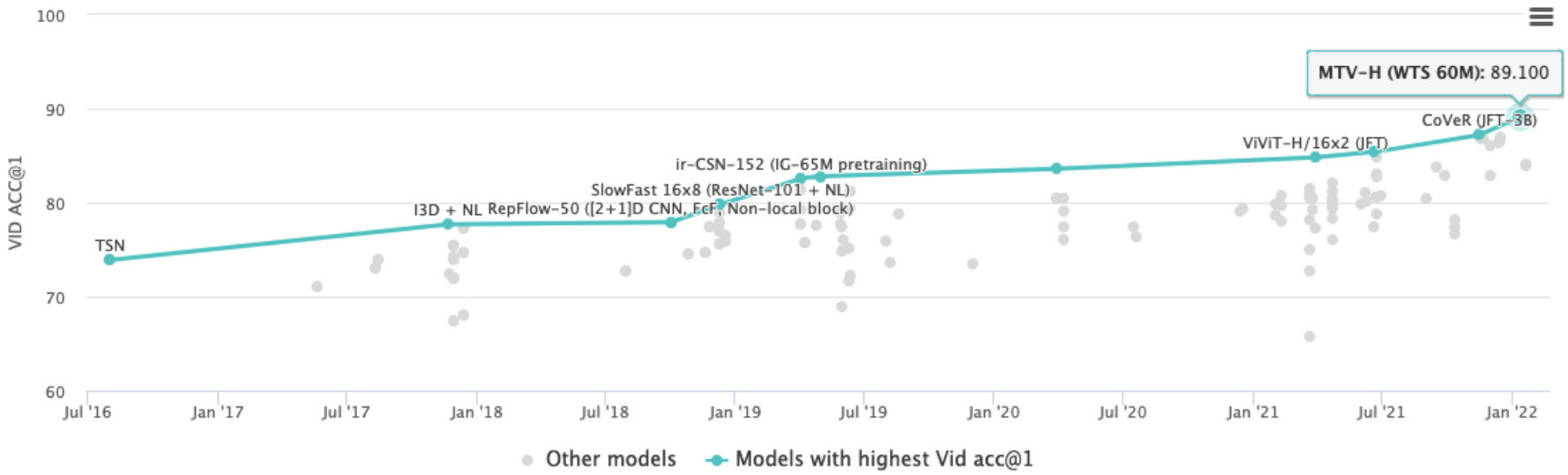
- Dependent on pre-trained Image weight from large private datasets such as JFT to achieve the SOTA results.

Conclusion and Future Work

- **ViViT is successful in implementing regularization and factorization at different levels to produce state-of-the-art models for Video tasks**

- **Dependence on Image-pretrained weights is an area for future improvements**

Beyond ViViT



Beyond ViViT : Kinetics-400 Dataset

Model Name	Top-1(%)	Top-5(%)	Paper
ViViT-H/16X2(JFT)	84.8	96.1	Multiview Transformers for Video Recognition
CoVeR	87.2	97.5	Co-Training Transformers with Videos and Images Improves Action Recognition
MTV-H	89.1	98.2	Multiview Transformers for Video Recognition

References

- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR, 2021.

- Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding?In arXiv preprint arXiv:2102.05095, 2021.

- Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In ICCV, 2019.

- Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition.In NeurIPS, 2016.

- Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. In arXiv