Supplementary Material
ViViT: A Video Vision Transformer

Anurag Arnab* Mostafa Dehghani* Georg Heigold Chen Sun Mario Lučić† Cordelia Schmid†
Google Research
{aarnab, dehghani, heigold, chensun, lucic, cordelias}@google.com

A1. Additional experimental details

In this supplementary, we provide additional experimental details. Section A1.1 ablates the number of temporal transformers in our Factorised encoder model, Section A1.2 provides additional details about the regularisers we used and Sec. A1.3 details the training hyperparamters used for our experiments. Finally, Sec. A1.4 provides further details about the Kinetics dataset as it is a dynamic dataset consisting of YouTube videos which may be removed.

A1.1. Temporal transformers in Factorised encoder model

Table A1 ablates the effect of the number of temporal transformers, L_t, in our Factorised encoder model on Kinetics 400 using ViViT-B as the backbone. We observe that the Top-1 accuracy on Kinetics is not sensitive to the choice of L_t. We also include an "average pooling baseline" corresponding to $L_t = 0$. As described in Sec. 4.2 of the main paper, instead of using temporal transformers to fuse temporal information across the video, we simply average pool the representations from each spatial transformer. This baseline performs substantially worse compared to $L_t \geq 1$.

A1.2. Further details about regularisers

In this section, we provide additional details and list the hyperparameters of the additional regularisers that we employed (Tab. 3 of the main paper). Hyperparameter values for all our experiments are listed in Tab. A2.

Stochastic depth Stochastic depth regularisation was originally proposed for training very deep residual networks [3]. Intuitively, the outputs of a layer, ℓ, are "dropped out" with probability, $p_{\text{drop}}(\ell)$ during training, by setting the output of the layer to be equal to its input.

Following [3], we linearly increase the probability of dropping a layer according to its depth within the network,

$$p_{\text{drop}}(\ell) = \frac{\ell}{L} p_{\text{drop}}, \tag{A1}$$

where ℓ is the index of the layer in the network, and L is the total number of layers.

Table A1. The effect of varying the number of temporal transformers, L_t, in the Factorised encoder model (Model 2). We report the Top-1 accuracy on Kinetics 400. Note that $L_t = 0$ corresponds to the "average pooling baseline" in Sec. 4.2 of the main paper.

L_t	0	1	4	8	12
Top-1	75.8	78.6	78.8	78.8	78.9

Random augment Random augment [2] randomly applies data augmentation transformations sequentially to an input example. We follow the public implementation[1], but modify the data augmentation operations to be temporally consistent throughout the video (in other words, the same transformation is applied on each frame of the video).

The authors define two hyperparameters for Random augment, "number of layers" , the number of augmentation transformations to apply sequentially to a video and "magnitude", the strength of the transformation that is shared across all augmentation operations. Our values for these parameters are shown in Tab. A2.

Label smoothing Label smoothing was proposed by [5] originally to regularise training Inception-v3. Concretely, the label distribution used during training, $\tilde{y}$, is a mixture of the one-hot ground-truth label, y, and a uniform distribution, u, to encourage the network to produce less confident predictions during training:

$$\tilde{y} = (1 - \lambda)y + \lambda u. \tag{A2}$$

There is therefore one scalar hyperparamter, $\lambda \in [0, 1]$.

*Equal contribution
†Equal advising

[1] https://github.com/tensorflow/models/blob/master/official/vision/beta/ops/augment.py

Table A2. Training hyperparamters for experiments in the main paper. "–" indicates that the regularisation method was not used at all. Values which are constant across all columns are listed once. Datasets are denoted as follows: K400: Kinetics 400. K600: Kinetics 600. MiT: Moments in Time. EK: Epic Kitchens. SSv2: Something-Something v2.

	K400	K600	MiT	EK	SSv2
Optimisation					
Optimiser			Synchronous SGD		
Momentum			0.9		
Batch size			64		
Learning rate schedule			cosine with linear warmup		
Linear warmup epochs			2.5		
Base learning rate	0.1	0.1	0.25	0.5	0.5
Epochs	30	30	10	50	35
Data augmentation					
Random crop probability			1.0		
Random flip probability			0.5		
Scale jitter probability			1.0		
Maximum scale			1.33		
Minimum scale			0.9		
Colour jitter probability	0.8	0.8	0.8	–	–
Rand augment number of layers [2]	–	–	–	2	2
Rand augment magnitude [2]	–	–	–	15	20
Other regularisation					
Stochastic droplayer rate, p_{drop} [3]	–	–	–	0.2	0.3
Label smoothing λ [5]	–	–	–	0.2	0.3
Mixup α [6]	–	–	–	0.1	0.3

Table A3. Additional details about the Kinetics datasets. As Kinetics consists of YouTube videos which may be removed by their original creators, we note the exact sizes of our dataset. Furthermore, we include results on the validation and test sets for our ViViT-L/16x2 model.

	Number of examples			Validation		Test		
	Train	Validation	Test	Top-1	Top-5	Top-1	Top-5	Views
Kinetics 400	214 834	17 637	34 579	81.7	93.8	80.8	93.2	1×3
Kinetics 600	363 213	27 676	55 377	82.9	94.6	82.5	94.3	1×3

Mixup Mixup [6] constructs virtual training examples which are a convex combination of pairs of training examples and their labels. Concretely, given (x_i, y_i) and (x_j, y_j) where x_i denotes an input vector and y_i a one-hot input label, mixup constructs the virtual training example,

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda) x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda) y_j. \end{aligned} \tag{A3}$$

$\lambda \in [0, 1]$, and is sampled from a Beta distribution, Beta(α, α). Our choice of the hyperparameter α is detailed in Tab. A2.

A1.3. Training hyperparameters

Table A2 details the hyperparamters for all of our experiments. We use synchronous SGD with momentum, a cosine learning rate schedule with linear warmup, and a batch size of 64 for all experiments. As mentioned in the main paper, we employed additional regularisation only when training on the smaller Epic Kitchens and Something-Something v2 datasets.

A1.4. Kinetics dataset details

Kinetics [1, 4] is a dynamic dataset – it consists of YouTube videos which are specified by their URLs, and it is possible that these videos are removed by their original creators. As a result, we report the exact number of videos in our version of Kinetics in Tab. A3. Furthermore, for completeness, we also report our results on the Kinetics test set. We note, however, that the prior work that we compared to in Tab. 6a and 6b of the main paper did not report results on the test set.

References

[1] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. In *arXiv preprint arXiv:1808.01340*, 2018. 2

[2] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 1, 2

[3] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 1, 2

[4] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. In *arXiv preprint arXiv:1705.06950*, 2017. 2

[5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 1, 2

[6] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2