Appendix

In this appendix, §A contains further *ablations* for Kinetics (§A.2) & ImageNet (§A.3), §C contains an *analysis* on computational complexity of MHPA, and §B qualitative *observations* in MViT and ViT models. §D contains additional *implementation details* for: Kinetics (§D.1), AVA (§D.2), Charades (§D.3), SSv2 (§D.4), and ImageNet (§D.5).

A. Additional Results

A.1. Results: Kinetics-700 Classification

model	pretrain	top-1	top-5	GFLOPs×views	Param
SlowFast 16×8 +NL [35]	K600	71.0	89.6	234×3×10	59.9
MViT-B, 16×4	-	71.2	90.0	70.5×1×5	36.8
MViT-B-24, 32×3	-	**74.0**	**91.7**	236×1×5	52.9

Table A.1. **Comparison with previous work on Kinetics-700**.

Kinetics-700 [13] is the largest version of Kinetics with 522k training videos. Results are in Table A.1. We train MViT from-scratch, without any pre-training. **MViT**-B, 16×4 achieves 71.2% top-1 accuracy already outperforming the best previous SlowFast [35] model. We further train a deeper 24-layer model with longer sampling, **MViT**-B-24, 32×3, which achieves 74.0% top-1 accuracy.

A.2. Ablations: Kinetics-400 Classification

This indicates that a naïve application of ViT to video does not model temporal information, and the temporal positional embedding in ViT-B seems to be fully ignored. We also verified this with the 79.3% ImageNet-21K pre-trained ViT-B of Table 4, which has ***the same accuracy*** of 79.3% for shuffling test frames, suggesting that it implicitly performs bag-of-frames video classification in Kinetics.

variant	$[N_1, N_2]$	FLOPs (G)	Mem (G)	Acc
ViT-B	[12, 0]	179.6	16.8	68.5
2-scale ViT-B, Q pool	[6, 6]	**111.1** (−68.5)	9.8 (−7.0)	**71.0** (+1.5)
ViT-B, K, V pool	[12, 0]	148.4 (−31.2)	**8.9** (−7.9)	69.1 (+0.6)

Table A.2. **Query (scale stage) and Key-Value pooling on ViT-B.** Introducing a *single* extra resolution stage into ViT-B boosts accuracy by +1.5%. Pooling K, V provides +0.6% accuracy. Both techniques allow dramatic FLOPs/memory savings.

Two scales in ViT. We provide a simple experiment that ablates the effectiveness of scale-stage design on ViT-B. For this we add a *single scale stage* to the ViT-B model. To isolate the effect of having different scales in ViT, we do not alter the channel dimensionality for this experiment. We do so by performing Q-Pooling with $\mathbf{s}^Q \equiv (1, 2, 2)$ after 6 Transformer blocks (*cf.* Table 3). Table A.2 shows the results. Adding a single scale stage to the ViT-B baseline boosts accuracy by +1.5% while deceasing FLOPs and memory cost by 38% and 41%. Pooling Key-Value tensors reduces compute and memory cost while slightly increasing accuracy.

	positional embedding	Param (M)	Acc
(i)	none	36.2	75.8
(ii)	space-only	36.5	76.7
(iii)	joint space-time	38.6	76.5
(iv)	**separate** in space & time	36.5	**77.2**

Table A.3. **Effect of separate space-time positional embedding.** Backbone: **MViT**-B, 16×4. FLOPs are 70.5G for all variants.

Separate space & time embeddings in MViT. In Table A.3, we ablate using (i) none, (ii) space-only, (iii) joint space-time, and (iv) a separate space and time (our default), positional embeddings. We observe that no embedding (i) decays accuracy by -0.9% over using just a spatial one (ii) which is roughly equivalent to a joint spatiotemporal one (iii). Our separate space-time embedding (iv) is best, and also has 2.1M fewer parameters than a joint spacetime embedding.

$T \times \tau$	$c_T \times c_H \times c_W$	$s_T \times s_H \times s_W$	FLOPs	Param	Acc
8×8	1×4×4	1×4×4	69.4	36.5	74.5
8×8	1×7×7	1×4×4	69.6	36.5	75.6
8×8	3×7×7	1×4×4	70.5	36.5	75.9
16×4	3×7×7	2×4×4	70.5	36.5	77.2
32×2	3×7×7	4×4×4	70.5	36.5	77.2
32×2	7×7×7	4×4×4	70.5	36.5	77.3

Table A.4. **Input sampling**: We vary sampling rate $T \times \tau$, the size $\mathbf{c}$=$c_T \times c_H \times c_W$ and stride of $\mathbf{s}$=$s_T \times s_H \times s_W$ the cube$_1$ layer that projects space-time cubes. Cubes with temporal extent $c_T > 1$ are beneficial. Our default setting is underlined.

Input Sampling Rate. Table A.4 shows results for different cubification kernel size $\mathbf{c}$ and sampling stride $\mathbf{s}$ (*cf.* Table 2). We observe that sampling *patches*, $\mathbf{c}_T = 1$, performs worse than sampling *cubes* with $\mathbf{c}_T > 1$. Further, sampling twice as many frames, T= 16, with twice the cube stride, s_T= 2, keeps the cost constant but boosts performance by +1.3% (75.9% → 77.2%). Also, sampling *overlapping* input cubes $\mathbf{s} < \mathbf{c}$ allows better information flow and benefits performance. While $\mathbf{c}_T > 1$ helps, very large temporal kernel size ($\mathbf{c}_T = 7$) does not further improve performance.

variant	$[N_2, N_3, N_4, N_5]$	FLOPs	Param	Mem	Acc
V1	[2, 6, 6, 2]	90.2	**29.5**	11.0	76.3
V2	[2, 4, 6, 4]	86.9	42.8	10.3	75.9
V3	[2, 4, 8, 2]	88.3	32.2	10.5	76.6
V4	[2, 2, 8, 4]	85.0	45.5	9.7	76.7
V5	[1, 2, 11, 2]	**83.6**	36.5	**9.1**	**77.1**
V6	[2, 2, 10, 2]	86.4	34.9	11.3	76.9

Table A.5. **Scale blocks**: We ablate the stage configuration as the number of blocks N in stages of **MViT**-B (*i.e.* where to pool Q). The overall number of transformer blocks is constant with N=16.

Stage distribution. The ablation in Table A.5 shows the results for distributing the number of transformer blocks in each individual scale stage. The overall number of transformer blocks, N=16 is consistent. We observe that having more blocks in early stages increases memory and having more blocks later stages the parameters of the architecture. Shifting the majority of blocks to the scale$_4$ stage (Variant V5 and V6 in Table A.5) achieves the best trade-off.

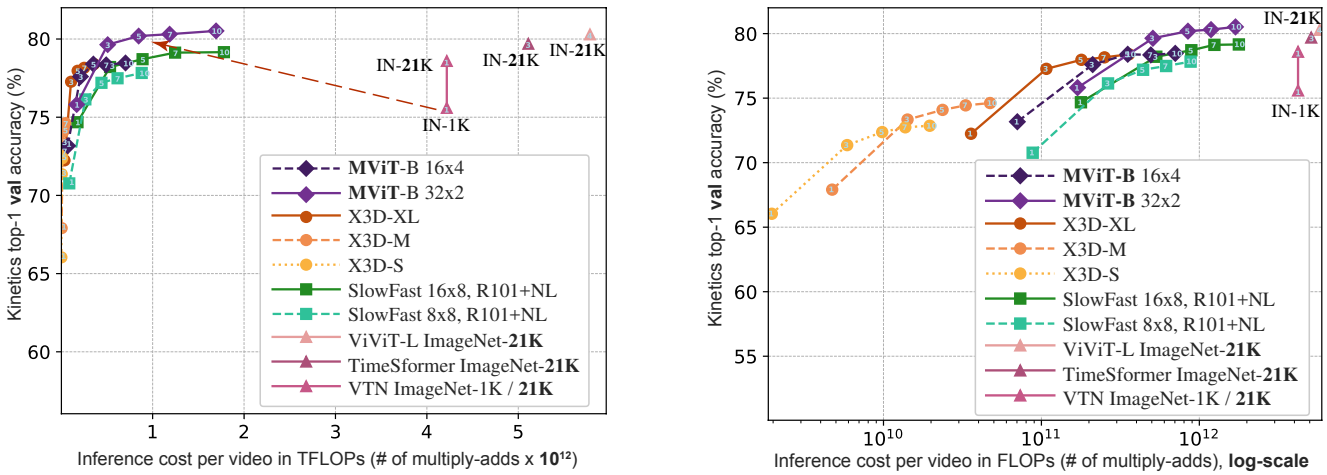


Figure A.4. **Accuracy/complexity trade-off** on K400-**val** for varying # of inference clips per video. The top-1 accuracy (vertical axis) is obtained by K-Center clip testing where the number of temporal clips $K \in \{1, 3, 5, 7, 10\}$ is shown in each curve. The horizontal axis measures the full inference cost per video. The left-sided plots show a linear and the right plots a logarithmic (**log**) scale.

stride s	adaptive	FLOPs	Mem	Acc
none	n/a	130.8	16.3	**77.6**
1×4×4		71.4	8.2	75.9
2×4×4		64.3	6.6	74.8
2×4×4	✓	83.6	9.1	77.1
1×8×8	✓	70.5	6.8	77.2
2×8×8	✓	**63.7**	**6.3**	75.8

Table A.6. **Key-Value pooling**: Vary stride $\mathbf{s} = s_T \times s_H \times s_W$, for pooling K and V. "adaptive" reduces stride w.r.t. stage resolution.

Key-Value pooling. The ablation in Table A.6 analyzes the pooling stride $\mathbf{s} = s_T \times s_H \times s_W$, for pooling K and V tensors. Here, we compare an "adaptive" pooling that uses a stride w.r.t. stage resolution, and keeps the K, V resolution *fixed* across all stages, against a non-adaptive version that uses the same stride at every block. First, we compare the baseline which uses no K, V pooling with non-adaptive pooling with a fixed stride of 2×4×4 across all stages: this drops accuracy from 77.6% to 74.8 (and reduces FLOPs and memory by over 50%). Using an adaptive stride that is 1×8×8 in the scale_1 stage, 1×4×4 in scale_2, and 1×2×2 in scale_3 gives the best accuracy of 77.2% while still preserving most of the efficiency gains in FLOPs and memory.

Inference cost. In the spirit of [33] we aim to provide further ablations for the effect of using *fewer* testing clips for efficient video-level inference. In Fig. A.4 we analyze the trade-off for the full inference of a video, when varying the number of temporal clips used. The vertical axis shows the top-1 accuracy on K400-val and the horizontal axis the overall inference cost in FLOPs for different model families: **MViT**, X3D [33], SlowFast [34], and concurrent ViT models, VTN [84] ViT-B-TimeSformer [8] ViT-L-ViViT [1], pre-trained on ImageNet-**21K**.

We first compare MViT with concurrent Transformer-based methods in the left plot in Fig. A.4. All these methods, VTN [84], TimeSformer [8] and ViViT [1], pre-train on ImageNet-**21K** and use the ViT [28] model with modifications on top of it. The inference FLOPs of these methods are around 5-10×higher than MViT models with equivalent performance; for example, ViT-L-ViViT [1] uses 4 clips of 1446G FLOPs (*i.e.* 5.78 TFLOPs) each to produce 80.3% accuracy while MViT-B, 32×3 uses 5 clips of 170G FLOPs (*i.e.* 0.85 TFLOPs) to produce 80.2% accuracy. Therefore, MViT-L can provide similar accuracy at 6.8× lower FLOPs (and 8.5× lower parameters), than concurrent ViViT-L [1]. More importantly, the MViT result is achieved *without external data*. All concurrent Transformer based works [84, 8, 1] require the huge scale ImageNet-21K to be competitive, and the performance degrades significantly (-3% accuracy, see IN-1K in Fig. A.4 for VTN [84]). These works further report failure of training without ImageNet initialization.

The plot in Fig. A.4 right shows this same plot with a logarithmic scale applied to the FLOPs axis. Using this scaling it is clearer to observe that smaller models convolutional models (X3D-S and X3D-M) can still provide more efficient inference in terms of multiply-add operations and MViT-B compute/accuracy trade-off is similar to X3D-XL.

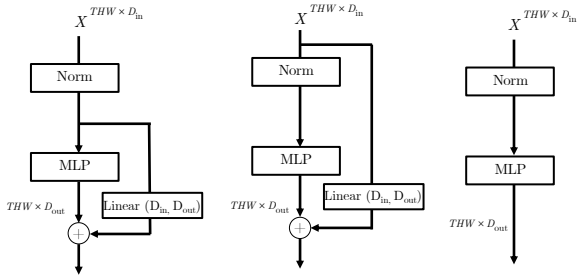
Ablations on skip-connections. Recall that, at each scale-stage transition in MViT, we expand the channel dimension by increasing the output dimension of the previous stages' MLP layer; therefore, it is not possible to directly apply the original skip-connection design [28], because the input channel dimension (D_{in}) differs from the output channel dimension (D_{out}). We ablate three strategies for this:

(a) First normalize the input with layer normalization and then expand its channel dimension to match the output dimension with a linear layer (Fig. A.5a); this is our default.

(b) Directly expand the channel dimension of the input

by using a linear layer to match the dimension (Fig. A.5b).

(c) No skip-connection for stage-transitions (Fig. A.5c).



(a) **normalized** skip-connection (b) unnormalized skip-connection (c) no skip-connection

Figure A.5. **Skip-connections at stage-transitions.** Three skip-connection variants for expanding channel dimensions: (a) first normalize the input with layer normalization (Norm) and then expand its channel dimension; (b) directly expand the channel dimension of the input; (c) no skip-connection at stage-transitions.

	method	top-1	top-5
(a)	**normalized** skip-connection	**77.2**	**93.1**
(b)	unnormalized skip-connection	74.6	91.3
(c)	no skip-connection	74.7	91.8

Table A.7. **Skip-connections at stage-transitions on K400.** We use our base model, MViT-B 16×4. Normalizing the skip-connection at channel expansion is essential for good performance.

Table A.7 shows the Kinetics-400 ablations for all 3 variants. Our default of using a normalized skip-connection (a) obtains the best results with 77.2% top-1 accuracy, while using an un-normalized skip-connection after channel expansion (b) decays significantly to 74.6% and using no skip-connection for all stage-transitions (c) has a similar result. We hypothesize that for expanding the channel dimension, normalizing the signal is essential to foster optimization, and use this design as our default in all other experiments.

backbone	recipe	Acc
SlowFast R50, 8×8	[34]	77.0
SlowFast R50, 8×8	MViT	67.4
SlowFast R101, 8×8	[34]	78.0
SlowFast R101, 8×8	MViT	61.6

Table A.8. **SlowFast models with MViT recipe on Kinetics-400.** The default recipe is using the recipe from the original paper. Accuracy is evaluated on 10×3 views.

SlowFast with MViT recipe. To investigate if our training recipe can benefit ConvNet models, we apply the same augmentations and training recipe as for MViT to SlowFast in Table A.8. The results suggest that SlowFast models do not benefit from the MViT recipe directly and more studies are required to understand the effect of applying our training-from-scratch recipe to ConvNets, as it seems higher capacity ConvNets (R101) perform worse when using our recipe.

A.3. Ablations: ImageNet Image Classification

We carry out ablations on ImageNet with the **MViT**-B-16 model with 16 layers, and show top-1 accuracy (Acc) as well as computational complexity measured in GFLOPs (floating-point operations). We also report Parameters in M(10^6) and training GPU memory in G(10^9) for a batch size of 512.

stride **s**	FLOPs	Mem	Acc
8×8	7.2	9.0	81.6
4×4	7.8	11.9	82.5
2×2	9.0	13.2	81.8
none	10.4	17.3	82.3

Table A.9. **ImageNet: Key-Value pooling**: We vary stride $s_H \times s_W$, for pooling K and V. We use "adaptive" pooling that reduces stride w.r.t. stage resolution.

Key-Value pooling for image classification. The ablation in Table A.9 analyzes the pooling stride $\mathbf{s} = s_H \times s_W$, for pooling K and V tensors. Here, we use our default 'adaptive' pooling that uses a stride w.r.t. stage resolution, and keeps the K, V resolution *fixed* across all stages.

First, we compare the baseline which uses pooling with a fixed stride of 4×4 with a model has a stride of 8×8: this drops accuracy from 82.5% to 81.6%, and reduces FLOPs and memory by 0.6G and 2.9G.

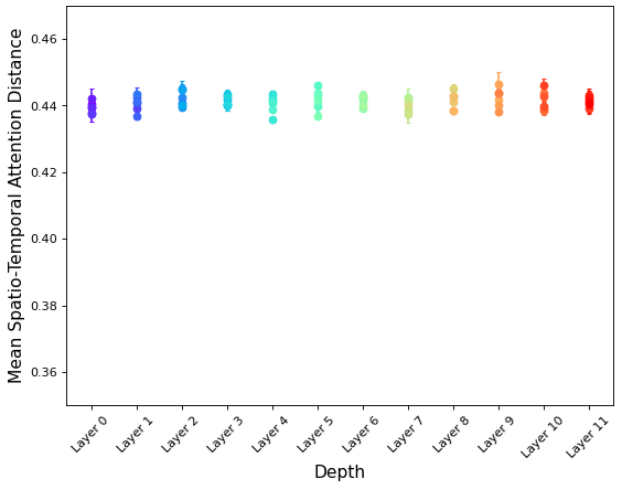
Second, we reduce the stride to 2×2, which increases FLOPs and memory significantly but performs 0.7% *worse* than our default stride of 4×4.

Third, we remove the K, V pooling completely which increases FLOPs by 33% and memory consumption by 45%, while providing lower accuracy than our default.

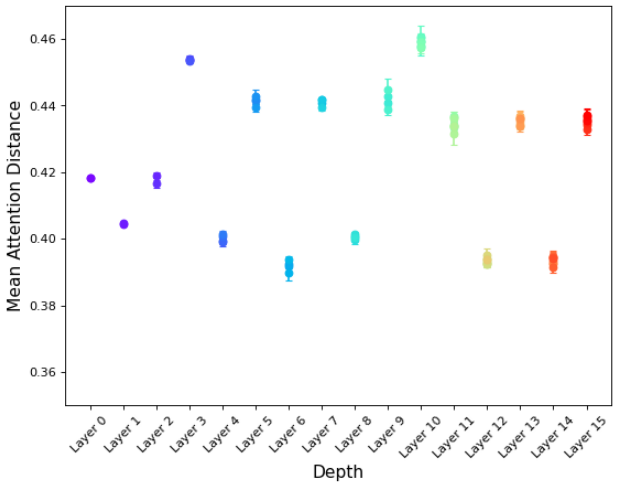
Overall, the results show that our K, V pooling is an effective technique to *increase* accuracy *and decrease* cost (FLOPs/memory) for image classification.

B. Qualitative Experiments: Kinetics

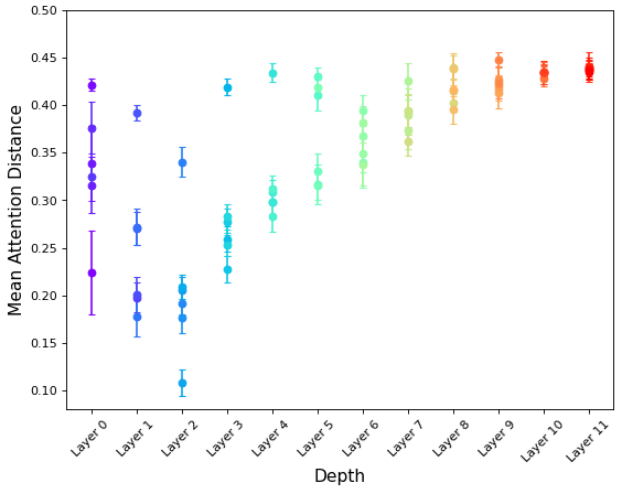
In Figure A.6, we plot the mean attention distance for all heads across all the layers of our Multiscale Transformer model and its Vision Transformer counterpart, at initialization with random weights, and at convergence after training. Each head represents a point in the plots (ViT-B has more heads). Both the models use the exact same weight initialization scheme and the difference in the attention signature stems purely from the multiscale skeleton in MViT. We observe that the dynamic range of attention distance is about **4**× larger in the MViT model than ViT *at initialization* itself (A.6a *vs.* A.6b). This signals the strong inductive bias stemming from the multiscale design of MViT. Also note that while at initialization, every layer in ViT has roughly the



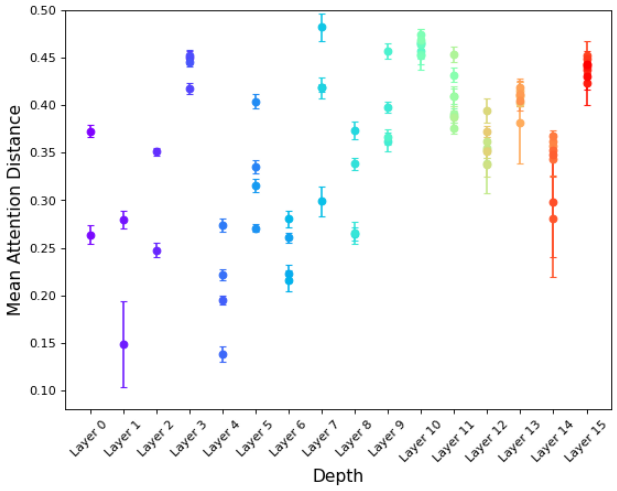
(a) ViT-B at **initialization**



(b) MViT-B at **initialization**



(c) ViT-B at **convergence**



(d) MViT-B at **convergence**

Figure A.6. **Mean attention distance** across layers ***at initialization/convergence*** for Vision Transformer (a)/(c) & Multiscale Vision Transformers (b)/(d). Each point shows the normalized average attention distance (weighted by the attention scores, with 1.0 being maximum possible distance) for each head in a layer. MViT attends close and distant features throughout the network hierarchy.

same mean attention distance, the MViT layers have strikingly different mean attention signatures indicating distinct predilections towards global and local features.

The bottom row of Fig. A.6 shows the same plot for a converged Vision Transformer (A.6c) and Multiscale Vision Transformer (A.6d) model.

We notice very different trends between the two models *after training*. While the ViT model (A.6c) has a consistent increase in attention distance across layers, the MViT model (A.6d) is not monotonic at all. Further, the intra-head variation in the ViT model decreases as the depth saturates, while, for MViT, different heads are still focusing on different features even in the higher layers. This suggests that some of the capacity in the ViT model might indeed be wasted with redundant computation while the lean MViT heads are more judiciously utilizing their compute. Noticeable is further a larger delta (between initialization in Fig. A.6a and convergence in A.6c) in the overall attention distance signature in the ViT model, compared to MViT's location distribution.

C. Computational Analysis

Since attention is quadratic in compute and memory complexity, pooling the key, query and value vectors have direct benefits on the fundamental compute and memory requirements of the pooling operator and by extension, on the complete Multiscale Transformer model. Consider an input tensor of dimensions $T \times H \times W$ and corresponding sequence length $L = T \cdot H \cdot W$. Further, assume the key, query and value strides to be $\mathbf{s}^K$, $\mathbf{s}^Q$ and $\mathbf{s}^V$. As described

in Sec. 3.1 in main paper, each of the vectors would experience a spatiotemporal resolution downsampling by a factor of their corresponding strides. Equivalently, the sequence length of query, key and value vectors would be reduced by a factor of f^Q, f^K and f^V respectively, where,

$$f^j = s_T^j \cdot s_H^j \cdot s_W^j, \; \forall \, j \in \{Q, K, V\}.$$

Computational complexity. Using these shorter sequences yields a corresponding reduction in space and runtime complexities for the pooling attention operator. Considering key, query and value vectors to have sequence lengths L/f_k, L/f_q and L/f_v after pooling, the overall runtime complexity of computing the key, query and value embeddings is $O(THWD^2/h)$ per head, where h is the number of heads in MHPA. Further, the runtime complexity for calculating the full attention matrix and the weighed sum of value vectors with reduced sequence lengths is $O(T^2H^2W^2D/f_qf_hh)$ per head. Computational complexity for pooling is

$$T(\mathcal{P}(\cdot;\mathbf{\Theta})) = O\left(THW \cdot D \cdot \frac{k_T k_W k_H}{s_T s_W s_H}\right),$$

which is negligible compared to the quadratic complexity of the attention computation and hence can be ignored in asymptotic notation. Thus, the final runtime complexity of MHPA is $O(THWD(D + THW/f_qf_k))$.

Memory complexity. The space complexity for storing the sequence itself and other tensors of similar sizes is $O(THWD)$. Complexity for storing the full attention matrix is $O(T^2H^2W^2h/f_qf_k)$. Thus the total space complexity of MHPA is $O(THWh(D/h + THW/f_qf_k))$.

Design choice. Note the trade-off between the number of channels D and the sequence length term THW/f_qf_k in both space and runtime complexity. This tradeoff in multi head pooling attention informs two critical design choices of Multiscale Transformer architecture.

First, as the effective spatiotemporal resolution decreases with layers because of diminishing THW/f_qf_k, the channel capacity is increased to keep the computational time spent (FLOPs) roughly the same for each stage.

Second, for a fixed channel dimension, D, higher number of heads h cause a prohibitively larger memory requirement because of the $(D + h * THW/f_qf_k)$ term. Hence, Multiscale Transformer starts with a small number of heads which is increased as the resolution factor THW/f_qf_k decreases, to hold the effect of $(D + h * THW/f_qf_k)$ roughly constant.

D. Additional Implementation Details

We implement our model with PySlowFast [31]. Code and models are available at: https://github.com/facebookresearch/SlowFast.

D.1. Details: Kinetics Action Classification

Architecture details. As in original ViT [28], we use residual connections [50] and Layer Normalization (LN) [2] in the pre-normalization configuration that applies LN at the beginning of the residual function, and our MLPs consist of two linear layers with GELU activation [53], where the first layer expands the dimension from D to $4D$, and the second restores the input dimension D, except at the end of a scale-stage, where we increase this channel dimensions to match the input of the next scale-stage. At such stage-transitions, our skip connections receive an extra linear layer that takes as input the layer-normalized signal which is also fed into the MLP. In case of Q-pooling at scale-stage transitions, we correspondingly pool the skip-connection signal.

Optimization details. We use the truncated normal distribution initialization in [47] and adopt synchronized AdamW [82] training on 128 GPUs following the recipe in [101, 34]. For Kinetics, we train for 200 epochs with 2 repeated augmentation [55] repetitions. The mini-batch size is 4 clips per GPU (so the overall batchsize is 512).

We adopt a half-period cosine schedule [81] of learning rate decaying: the learning rate at the n-th iteration is $\eta \cdot 0.5[\cos(\frac{n}{n_{\max}}\pi) + 1]$, where $n_{\max}$ is the maximum training iterations and the base learning rate η is set as $1.6 \cdot 10^{-3}$. We linearly scale the base learning rate w.r.t. the overall batch-size, $\eta = 1.6{\cdot}10^{-3}\frac{\text{batchsize}}{512}$, and use a linear warm-up strategy in the first 30 epochs [42]. The cosine schedule is completed when reaching a final learning rate of $1.6 \cdot 10^{-5}$. We extract the class token after the last stage and use it as the input to the final linear layer to predict the output classes. For **Kinetics-600** all hyper-parameters are identical to K400.

Regularization details. We use weight decay of $5{\cdot}10^{-2}$, a dropout [54] of 0.5 before the final classifier, label-smoothing [98] of 0.1 and use stochastic depth [59] (*i.e.* drop-connect) with rate 0.2.

Our data augmentation is performed on input clips by applying the same transformation across all frames. To each clip, we apply a random horizontal flip, Mixup [119] with $\alpha = 0.8$ to half of the clips in a batch and CutMix [118] to the other half, Random Erasing [122] with probability 0.25, and Rand Augment [22] with probability of 0.5 for 4 layers of maximum magnitude 7.

For the temporal domain, we randomly sample a clip from the full-length video, and the input to the network are T frames with a temporal stride of τ; denoted as $T \times \tau$ [34]. For the spatial domain, we use Inception-style [97] cropping that randomly resizes the input *area* between a [min, max], scale of [0.08, 1.00], and jitters aspect ratio between 3/4 to 4/3, before taking an $H \times W$ = 224×224 crop.

Fine-tuning from ImageNet. To fine-tune our ViT-B baseline, we extend it to take a video clip of $T = 8$ frames

as input and initialize the model weights from the ViT-B model [28] pre-trained on ImageNet-21K dataset. The positional embedding is duplicated for each frame. We fine-tune the model for 30 epochs with SGD using the recipe in [34]. The mini-batch size is 2 clips per GPU and a half-period cosine learning rate decay is used. We linearly scale the base learning rate w.r.t. the overall batch-size, $\eta = 10^{-3}\frac{\text{batchsize}}{16}$. Weight decay is set to 10^{-4}.

D.2. Details: AVA Action Detection

Dataset. The AVA dataset [44] has bounding box annotations for spatiotemporal localization of (possibly multiple) human actions. It has 211k training and 57k validation video segments. We follow the standard protocol reporting mean Average Precision (mAP) on 60 classes [44] on AVA v2.2.

Detection architecture. We follow the detection architecture in [34] to allow direct comparison of MViT against SlowFast networks as a backbone.

First, we reinterpret our transformer spacetime cube outputs from MViT as a spatial-temporal feature map by concatenating them according to the corresponding temporal and spatial location.

Second, we employ a the detector similar to Faster R-CNN [90] with minimal modifications adapted for video. Region-of-interest (RoI) features [41] are extracted at the generated feature map from MViT by extending a 2D proposal at a frame into a 3D RoI by replicating it along the temporal axis, similar as done in previous work [44, 95, 63], followed by application of frame-wise RoIAlign [48] and temporal global average pooling. The RoI features are then max-pooled and fed to a per-class, sigmoid classifier for prediction.

Training. We initialize the network weights from the Kinetics models and adopt synchronized SGD training on 64 GPUs. We use 8 clips per GPU as the mini-batch size and a half-period cosine schedule of learning rate decaying. The base learning rate is set as 0.6. We train for 30 epochs with linear warm-up [42] for the first 5 epochs and use a weight decay of 10^{-8} and stochastic depth [59] with rate 0.4. Ground-truth boxes, and proposals overlapping with ground-truth boxes by IoU > 0.9, are used as the samples for training. The region proposals are identical to the ones used in [34].

Inference. We perform inference on a single clip with T frames sampled with stride τ centered at the frame that is to be evaluated.

D.3. Details: Charades Action Classification

Dataset. Charades [92] has ~9.8k training videos and 1.8k validation videos in 157 classes in a multi-label classification setting of longer activities spanning ~30 seconds on average. Performance is measured in mean Average Precision (mAP).

Training. We fine-tune our MViT models from the Kinetics models. A per-class sigmoid output is used to account for the multi-class nature. We train with SGD on 32 GPUs for 200 epochs using 8 clips per GPU. The base learning rate is set as 0.6 with half-period cosine decay. We use weight decay of 10^{-7} and stochastic depth [59] with rate 0.45. We perform the same data augmentation schemes as for Kinetics in §D.1, except of using Mixup.

Inference. To infer the actions over a single video, we spatiotemporally max-pool prediction scores from multiple clips in testing [34].

D.4. Details: Something-Something V2 (SSv2)

Dataset. The Something-Something V2 dataset [43] contains 169k training, and 25k validation videos. The videos show human-object interactions to be classified into 174 classes. We report accuracy on the validation set.

Training. We fine-tune the pre-trained Kinetics models. We train for 100 epochs using 64 GPUs with 8 clips per GPU and a base learning rate of 0.02 with half-period cosine decay [81]. Weight decay is set to 10^{-4} and stochastic depth rate [59] is 0.4. Our training augmentation is the same as in §D.1, but as SSv2 requires distinguishing between directions, we disable random flipping in training. We use segment-based input frame sampling [75] that splits each video into segments, and from each of them, we sample one frame to form a clip.

Inference. We take single clip with 3 spatial crops to form predictions over a single video in testing.

D.5. Details: ImageNet

Datasets. For image classification experiments, we perform our experiments on ImageNet-1K [25] dataset that has ~1.28M images in 1000 classes. We train models on the train set and report top-1 and top-5 classification accuracy (%) on the val set. Inference cost (in FLOPs) is measured from a single center-crop with resolution of 224^2 if the input resolution was not specifically mentioned.

Training. We use the training recipe of DeiT [101] and summarize it here for completeness. We train for 100 epochs with 3 repeated augmentation [55] repetitions (overall computation equals 300 epochs), using a batch size of 4096 in 64 GPUs. We use truncated normal distribution initialization [47] and adopt synchronized AdamW [82] optimization with a base learning rate of 0.0005 per 512 batch-size that is warmed up and decayed as half-period cosine, as in [101]. We use a weight decay of 0.05, label-smoothing [98] of 0.1. Stochastic depth [59] (*i.e.* drop-connect) is also used with rate 0.1 for model with depth of 16 (MViT-B-16), and rate 0.3 for deeper models (MViT-B-24). Mixup [119] with $\alpha = 0.8$ to half of the clips in a batch and CutMix [118] to

the other half, Random Erasing [122] with probability 0.25, and Rand Augment [22] with maximum magnitude 9 and probability of 0.5 for 4 layers (for max-pooling) or 6 layers (for conv-pooling).

Acknowledgements

We are grateful for discussions with Chao-Yuan Wu, Ross Girshick, and Kaiming He.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 2, 3, 6, 14

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1, 17

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv:1607.06450*, 2016. 4

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014. 1

[5] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020. 2

[6] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. *International Conference on Learning Representations*, 2021. 3

[7] Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V Le. Attention augmented convolutional networks. *International Conference on Computer Vision*, 2019. 2

[8] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 2, 3, 6, 7, 14

[9] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[10] Peter J Burt and Edward H Adelson. The laplacian pyramid as a compact image code. In *Readings in computer vision*, pages 671–679. Elsevier, 1987. 1

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. ECCV*, pages 213–229. Springer, 2020. 2

[12] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about Kinetics-600. *arXiv:1808.01340*, 2018. 2, 6

[13] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 13

[14] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 2, 6

[15] Chun-Fu Chen, Quanfu Fan, Neil Mallinar, Tom Sercu, and Rogerio Feris. Big-little net: An efficient multi-scale feature representation for visual and speech recognition. *arXiv preprint arXiv:1807.03848*, 2018. 2

[16] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 2

[17] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *Proc. ICML*, pages 1691–1703. PMLR, 2020. 2

[18] Yunpeng Chen, Haoqi Fang, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. *arXiv preprint arXiv:1904.05049*, 2019. 2

[19] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019. 3

[20] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020. 3

[21] Xiangxiang Chu, Bo Zhang, Zhi Tian, Xiaolin Wei, and Huaxia Xia. Do we really need explicit position encodings for vision transformers? *arXiv preprint arXiv:2102.10882*, 2021. 2

[22] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proc. CVPR*, 2020. 17, 19

[23] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc V Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *arXiv preprint arXiv:2006.03236*, 2020. 3

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255. Ieee, 2009. 2, 3

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 6, 8, 18

[26] Karan Desai and Justin Johnson. Virtex: Learning visual representations from textual annotations. *arXiv preprint arXiv:2006.06666*, 2020. 2

[27] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *Proc. CVPR*, 2021. 8

[28] 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. *arXiv preprint arXiv:2010.11929*, 2020. 1, 2, 3, 4, 5, 6, 8, 14, 17, 18

[29] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018. 1

[30] Alaaeldin El-Nouby, Natalia Neverova, Ivan Laptev, and Hervé Jégou. Training vision transformers for image retrieval. *arXiv preprint arXiv:2102.05644*, 2021. 2

[31] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast. https://github.com/facebookresearch/slowfast, 2020. 2, 17

[32] Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, and Christoph Feichtenhofer. PyTorchVideo: A deep learning library for video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. https://pytorchvideo.org/. 2

[33] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proc. CVPR*, pages 203–213, 2020. 2, 6, 7, 8, 14

[34] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proc. ICCV*, 2019. 2, 6, 7, 8, 14, 15, 17, 18

[35] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition in ActivityNet challenge 2019. http://static.googleusercontent.com/media/research.google.com/en//ava/2019/fair_slowfast.pdf, 2019. 13

[36] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. CVPR*, 2016. 2

[37] Kunihiko Fukushima and Sei Miyake. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, pages 267–285. Springer, 1982. 1

[38] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *Proc. ECCV*, volume 5. Springer, 2020. 2

[39] Shanghua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip HS Torr. Res2net: A new multi-scale backbone architecture. *IEEE PAMI*, 2019. 2

[40] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proc. CVPR*, 2019. 2

[41] Ross Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 18

[42] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv:1706.02677*, 2017. 17, 18

[43] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 2, 6, 7, 18

[44] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Proc. CVPR*, 2018. 2, 6, 7, 18

[45] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *arXiv preprint arXiv:2012.09688*, 2020. 2

[46] Zhang Hang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, and Yue Sun. Resnest: Split-attention networks. 2020. 2

[47] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *arXiv preprint arXiv:1803.01719*, 2018. 17, 18

[48] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 18

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. CVPR*, 2015. 1

[50] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 5, 17

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. ECCV*, 2016. 2

[52] Shuting He, Hao Luo, Pichao Wang, Fan Wang, Hao Li, and Wei Jiang. Transreid: Transformer-based object re-identification. *arXiv preprint arXiv:2102.04378*, 2021. 2

[53] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016. 17

[54] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012. 17

[55] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proc. CVPR*, pages 8129–8138, 2020. 6, 17, 18

[56] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proc. CVPR*, pages 3588–3597, 2018. 2

[57] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proc. ICCV*, pages 3464–3473, 2019. 2

[58] Ronghang Hu and Amanpreet Singh. Transformer is all you need: Multimodal multitask learning with a unified transformer. *arXiv preprint arXiv:2102.10772*, 2021. 2

[59] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Proc. ECCV*, 2016. 17, 18

[60] DH Hubel and TN Wiesel. Receptive fields of optic nerve fibres in the spider monkey. *The Journal of physiology*, 154(3):572–580, 1960. 1

[61] Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *Proc. CVPR*, 2019. 7

[62] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proc. CVPR*, pages 2000–2009, 2019. 2

[63] Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proc. CVPR*, 2018. 18

[64] 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. *arXiv:1705.06950*, 2017. 2, 6

[65] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*, 2020. 3

[66] Jan J Koenderink. The structure of images. *Biological cybernetics*, 50(5):363–370, 1984. 1

[67] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[68] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. In *Proc. ECCV*, pages 345–362. Springer, 2020. 7

[69] Hung Le, Doyen Sahoo, Nancy F Chen, and Steven CH Hoi. Multimodal transformer networks for end-to-end video-grounded dialogue systems. *arXiv preprint arXiv:1907.01166*, 2019. 2

[70] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 1, 2

[71] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019. 2

[72] Rui Li, Chenxi Duan, and Shunyi Zheng. Linear attention mechanism: An efficient attention for semantic segmentation. *arXiv preprint arXiv:2007.14902*, 2020. 3

[73] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proc. CVPR*, pages 909–918, 2020. 7

[74] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. VideoLSTM convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 2

[75] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. In *Proc. ICCV*, 2019. 18

[76] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. CVPR*, pages 7083–7093, 2019. 7

[77] Kevin Lin, Lijuan Wang, and Zicheng Liu. End-to-end human pose and mesh reconstruction with transformers. *arXiv preprint arXiv:2012.09760*, 2020. 2

[78] Wei Liu, Sihan Chen, Longteng Guo, Xinxin Zhu, and Jing Liu. Cptr: Full transformer network for image captioning. *arXiv preprint arXiv:2101.10804*, 2021. 2

[79] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 8

[80] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020. 2

[81] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 17, 18

[82] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 17, 18

[83] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *arXiv preprint arXiv:1908.02265*, 2019. 2

[84] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 2, 3, 6, 14

[85] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proc. ICCV*, 2017. 2

[86] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021. 2

[87] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proc. CVPR*, June 2020. 2, 8

[88] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 2

[89] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021. 2

[90] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 18

[91] Azriel Rosenfeld and Mark Thurston. Edge and curve detection for visual scene analysis. *IEEE Transactions on computers*, 100(5):562–569, 1971. 1

[92] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2, 6, 7, 18

[93] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2, 5

[94] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 2

[95] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *ECCV*, 2018. 18

[96] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 2

[97] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 17

[98] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv:1512.00567*, 2015. 17, 18

[99] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 2, 8

[100] Yi Tay, Dara Bahri, Liu Yang, Donald Metzler, and Da-Cheng Juan. Sparse sinkhorn attention. In *Proc. ICML*, pages 9438–9447. PMLR, 2020. 3

[101] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 1, 2, 6, 8, 17, 18

[102] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proc. ICCV*, 2019. 2, 6

[103] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. Medical transformer: Gated axial-attention for medical image segmentation. *arXiv preprint arXiv:2102.10662*, 2021. 2

[104] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1, 2, 3, 4

[105] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. *arXiv preprint arXiv:2012.00759*, 2020. 2

[106] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 3

[107] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. CVPR*, 2018. 2, 7

[108] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proc. ECCV*, 2018. 7

[109] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. *arXiv preprint arXiv:2011.14503*, 2020. 2

[110] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 2

[111] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019. 2, 7

[112] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv:1712.04851*, 2017. 2

[113] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. 2

[114] Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. Multimodal transformer with multi-view visual representation for image captioning. *IEEE transactions on circuits and systems for video technology*, 30(12):4467–4480, 2019. 2

[115] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 2

[116] Zhenxun Yuan, Xiao Song, Lei Bai, Wengang Zhou, Zhe Wang, and Wanli Ouyang. Temporal-channel transformer for 3d lidar-based video object detection in autonomous driving. *arXiv preprint arXiv:2011.13628*, 2020. 2

[117] Boxiang Yun, Yan Wang, Jieneng Chen, Huiyu Wang, Wei Shen, and Qingli Li. Spectr: Spectral transformer for hyperspectral pathology image segmentation. *arXiv preprint arXiv:2103.03604*, 2021. 2

[118] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proc. ICCV*, 2019. 17, 18

[119] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Proc. ICLR*, 2018. 17, 18

[120] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proc. CVPR*, pages 10076–10085, 2020. 2

[121] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020. 2

[122] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 17, 19

[123] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 2