

VidTr: Video Transformer Without Convolutions

Yanyi Zhang^{*1}, Xinyu Li^{*2}, Chunhui Liu², Bing Shuai², Yi Zhu²,
Biagio Brattoli², Hao Chen², Ivan Marsic¹, Joseph Tighe²
¹ Rutgers University; ² Amazon Web Services

Abstract

We introduce Video Transformer (VidTr) with separable-attention for video classification. Comparing with commonly used 3D networks, VidTr is able to aggregate spatio-temporal information via stacked attentions and provide better performance with higher efficiency. We first introduce the vanilla video transformer and show that transformer module is able to perform spatio-temporal modeling from raw pixels, but with heavy memory usage. We then present VidTr which reduces the memory cost by $3.3\times$ while keeping the same performance. To further compact the model, we propose the standard deviation based topK pooling attention, which reduces the computation by dropping non-informative features. VidTr achieves state-of-the-art performance on five commonly used dataset with lower computational requirement, showing both the efficiency and effectiveness of our design. Finally, error analysis and visualization show that VidTr is especially good at predicting actions that require long-term temporal reasoning. The code and pre-trained weights will be released.

1. Introduction

We introduce Video Transformer (VidTr) with separable-attention, one of the first transformer-based video action classification architecture that performs global spatio-temporal feature aggregation. Convolution-based architectures have dominated the video classification literature in recent years [17, 30, 51], and although successful, the convolution-based approaches have two drawbacks: 1. they have limited receptive field on each layer and 2. information is slowly aggregated through stacked convolution layers, which is inefficient and might be ineffective [29, 51]. Attention is a potential candidate to overcome these limitations as it has a large receptive field which can be leveraged for spatio-temporal modeling. Previous works use attention to modeling long-range spatio-temporal features in videos but still rely on convolutional backbones [29, 51].

^{*}Equally contributed



Figure 1: The proposed VidTr first locates the spatial important regions on each temporal instances via spatial attention (third row) and further select informative temporal instance via temporal attention (bottom row). I3D usually only focus on features in a local region even they are not informative to decision making (top row).

Inspired by recent successful applications of transformers on NLP [11, 48] and computer vision [13, 43], we propose a transformer-based video network that directly applies attentions on raw video pixels for video classification, aiming at higher efficiency and better performance (Figure 1).

We first introduce a vanilla video transformer that directly learns spatio-temporal features from raw-pixel inputs via vision transformer [13], showing that it is possible to perform pixel-level spatio-temporal modeling. However, as discussed in [53], the transformer has $\mathcal{O}(n^2)$ complexity with respect to the sequence length. The vanilla video transformer is memory consuming, as training on a 16-frame clip (224×224) with only batch size of 1 requires more than 16GB GPU memory, which makes it infeasible on most commercial devices. Inspired by the R(2+1)D convolution [46], we further introduce our separable-attention, which performs spatial and temporal attention separately. This reduces the memory consumption by $3.3\times$ with no drop in accuracy. We can further reduce the memory and computa-

tional requirements of our system by exploiting the fact that a large portion of many videos have redundant information as they contain many near duplicate frames. This notion has been explored in the context of convolutional networks to reduce computation previously [30]. We build on this intuition and propose a standard deviation based topK pooling operation (*topK_std* pooling), which reduces the sequence length and encourages the transformer network to focus on representative frames.

We evaluated our VidTr on 6 most commonly used datasets, including Kinetics 400/700, Charades, Something-something V2, UCF-101 and HMDB-51. Our model achieved state-of-the-art (SOTA) or comparable performance on five datasets with lower computational requirements and latency compared to previous SOTA approaches. Our error analysis and ablation experiments show that the VidTr works significantly better than I3D on activities that requires longer temporal reasoning (e.g. making a cake vs. eating a cake), which aligns well with our intuition. This also inspires us to ensemble the VidTr with the I3D convolutional network as features from global and local modeling methods should be complementary. We show that simply combining the VidTr with a light weight I3D50 model (8 frames input) via ensemble can lead to roughly a 2% performance improvement on Kinetics 400 (see Appendix C for details). We further illustrate how and why the VidTr works by visualizing the separable-attention using attention rollout [1], and show that the spatial-attention is able to focus on informative patches while temporal attention is able to reduce the duplicated/non-informative temporal instances. Our contributions are:

1. **Video transformer:** We propose to efficiently and effectively aggregate spatio-temporal information with stacked attentions as opposed to convolution based approaches. We introduce vanilla video transformer as proof of concept with SOTA comparable performance on video classification.
2. **VidTr:** We introduce VidTr and its permutations, including the VidTr with SOTA performance and the compact-VidTr with significantly reduced computational costs using the proposed standard deviation based pooling method, that fit in different application scenarios.
3. **Results and model weights:** We provide detailed results and analysis on 6 commonly used datasets which can be used as reference for future research. Our pre-trained model can be used for many down-streaming tasks.

2. Related Work

2.1. Action Classification

The early research on video based action recognition relies on 2D convolutions [26]. The LSTM [23] was later proposed to model the image feature based on ConvNet features [28, 47, 60]. However, the combination of ConvNet and LSTM did not lead to significantly better perfor-

mance. Instead of relying on RNNs, the segment based method TSN [50] and its permutations [20, 33, 61] were proposed with good performance.

Although 2D network was proved successful, the spatio-temporal modeling was still separated. Using 3D convolution for spatio-temporal modeling was initially proposed in [24] and further extended to the C3D network [44]. However, training 3D convnet from scratch was hard, initializing the 3D convnet weights by inflate from 2D networks was initially proposed in I3D [6] and soon proved applicable with different type of 2D network [9, 22, 55]. The I3D was used as backbone for many following work including two-stream network [17, 51], the networks with focus on temporal modeling [29, 30, 56], and the 3D networks with refined 3D convolution kernels [25, 31, 37, 40].

The 3D networks are proved effective but often not efficient, the 3D networks with better performance often requires larger kernels or deeper structures. The recent research demonstrates that depth convolution significantly reduce the computation [45], but depth convolution also increase the network inference latency. TSM [35] and TAM [15] proposed a more efficient backbone for temporal modeling, however, such design couldn't achieve SOTA performance on Kinetics dataset. The neural architecture search was proposed for action recognition [16, 39] recently with competitive performance, however, the high latency and limited generalizability remain to be improved.

The previous methods heavily rely on convolution to aggregate features spatio-temporally, which is not efficient. A few previous work tried to perform global spatio-temporal modeling [29, 51] but still limited by the convolution backbone. The proposed VidTr is fundamentally different from previous works based on convolutions, the VidTr doesn't require heavily stacked convolutions [56] for feature aggregation but efficiently learn feature globally via attention from first layer. Besides, the VidTr don't rely on sliding convolutions and depth convolutions, which runs at less FLOPs and lower latency compared with 3D convolutions [16, 56].

2.2. Vision Transformer

The transformers [48] was previously proposed for NLP tasks [12] and recently adopted for computer vision tasks. The transformers were roughly used in three different ways in previous works: 1. To bridge the gap between different modalities, e.g. video captioning [62], video retrieval [18] and dialog system [34]. 2. To aggregate convolutional features for down-streaming tasks, e.g. object detection [4, 10], pose estimation [58], semantic segmentation [14] and action recognition [19]. 3. To perform feature learning on raw pixels, e.g. most recently image classification [13, 43].

Action recognition with self-attention on convolution features [19] is proved successful, however, convolution also generates local feature and gives redundant computa-

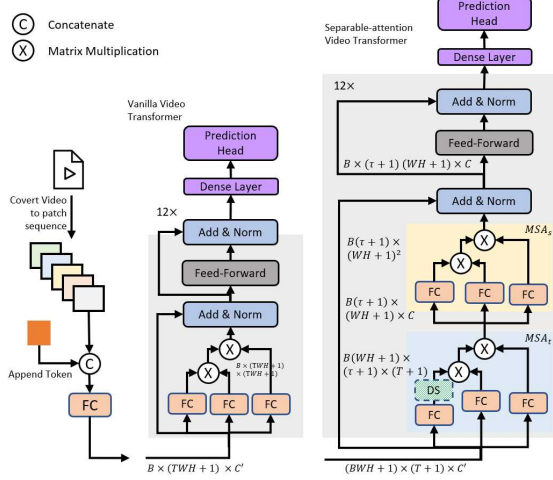


Figure 2: Spatio-temporal separable-attention video transformer (VidTr). The model takes pixels patches as input and learn the spatial temporal feature via proposed separable-attention. The green shaded block denotes the down-sample module which can be inserted into VidTr for higher efficiency. τ denotes the temporal dimension after downsampling.

tions. Different from [19] and inspired by very recent work on applying transformer on raw pixels [13, 43], we pioneer the work on aggregating spatio-temporal feature from raw videos without relying on convolution features. Different from very recent work [38] that extract spatial feature with vision transformer on every video frames and then aggregate feature with attention, our proposed method jointly learns spatio-temporal feature with lower computational cost and higher performance. Our work also differs from very recent work [3], we present a split attention with better performance without requiring larger video resolution nor extra long clip length. Our intention is not to challenge the convolution based approach but rather explore an alternative to existing methods which can be complementary to convolution based approaches.

3. Video Transformer

We introduce the Video Transformer starting with the vanilla video transformer (section 3.1) which illustrates our idea of video action recognition without convolutions. We then present VidTr by first introducing separable-attention (section 3.2), and then the attention pooling to drop non-representative information temporally (section 3.2).

3.1. Vanilla Video Transformer

Following previous efforts in NLP [12] and image classification [13], we adopted the transformer [48] encoder structure for action recognition that operates on raw pixels. Given a video clip $V \in \mathbb{R}^{C \times T \times W \times H}$, where T denotes

the clip length, W and H denote the video frame width and height, and C denotes the number of channel, we first convert V to a sequence of $s \times s$ spatial patches, and apply a linear embedding to each patch, namely $S \in \mathbb{R}^{T \frac{H}{s} \frac{W}{s} \times C'}$, where C' is the channel dimension after the linear embedding. We add a 1D learnable positional embedding [12, 13] to S and following previous work [12, 13], append a class token as well, whose purpose is to aggregate features from the whole sequence for classification. This results in $S' \in \mathbb{R}^{(\frac{TW}{s^2}+1) \times C'}$, where $S'_0 \in \mathbb{R}^{1 \times C'}$ is the attached class token. S' is feed into our transformer encoder structure detailed next.

As Figure 2 middle shows, we expand the previous successful ViT transformer architecture for 3D feature learning. Specifically, we stack 12 encoder layers, with each encoder layer consisting of an 8-head self-attention layer and two dense layers with 768 and 3072 hidden units. Different from transformers for 2D images, each attention layer learns a spatio-temporal affinity map $Attn \in \mathbb{R}^{(\frac{TW}{s^2}+1) \times (\frac{TW}{s^2}+1)}$.

3.2. VidTr

In Table 2 we show that this simple formulation is capable of learning 3D motion features on a sequence of local patches. However, as explained in [2], the affinity attention matrix $Attn \in \mathbb{R}^{(\frac{TW}{s^2}+1) \times (\frac{TW}{s^2}+1)}$ needs to be stored in memory for back propagating, and thus the memory consumption is quadratically related to the sequence length. We can see that the vanilla video transformer increases memory usage for the affinity map from $\mathcal{O}(W^2H^2)$ to $\mathcal{O}(T^2W^2H^2)$, leading to $T^2 \times$ memory usage for training, which makes it impractical on most available GPU devices. We now address this inefficiency with a separable attention architecture.

3.2.1 Separable-Attention

To address such memory constraint, we introduce a multi-head separable-attention (MSA) by decoupling the 3D self-attention to a spatial attention MSA_s and a temporal attention MSA_t (Figure 2):

$$MSA(S) = MSA_s(MSA_t(S)) \quad (1)$$

Different from the vanilla video transformer that applies 1D sequential modeling on S , we decouple S to a 2D sequence $\hat{S} \in \mathbb{R}^{(T+1) \times (\frac{WH}{s^2}+1) \times C'}$ with positional embedding and two types of class tokens that append additional tokens along the spatial and temporal dimensions. Here, the spatial class tokens gather information from spatial patches in a single frame using spatial attention, and the temporal class tokens gather information from patches across frames (at same location) using temporal attention. Then the intersection of the spatial and temporal class tokens $\hat{S}^{(0,0,:)}$

is used for the final classification. To decouple 1D self-attention functions on 2D sequential features \hat{S} , we first operate on each spatial location (i) independently, applying temporal attention as:

$$\hat{S}_t^{(:,i,:)} = \text{MSA}_t(k = q = v = \hat{S}^{(:,i,:)}) \quad (2)$$

$$= \text{pool}(\text{Attn}_t) \cdot v_t \quad (3)$$

$$= \text{pool}(\text{Softmax}(q_t \cdot k_t^T)) \cdot v_t \quad (4)$$

where $\hat{S}_t \in \mathbb{R}^{(\tau+1) \times (\frac{WH}{s^2}+1) \times C}$ is the output of MSA_t , pool denotes the down-sampling method to reduce the redundant information (from T to τ , $\tau = T$ when no down-sampling is performed) that will be described later, q_t , k_t , and v_t denote key, query, and value features after applying independent linear functions (LN) on \hat{S} :

$$q_t = \text{LN}_q(\hat{S}^{(:,i,:)}); k_t = \text{LN}_k(\hat{S}^{(:,i,:)}); v_t = \text{LN}_v(\hat{S}^{(:,i,:)}) \quad (5)$$

Moreover, $\text{Attn}_t \in \mathbb{R}^{(\tau+1) \times (T+1)}$ represent a temporal attention obtained from matrix multiplication between q_t and k_t . Following MSA_s , we apply a similar 1D sequential self-attention MSA_s on spatial dimension:

$$\hat{S}_{st}^{(i,:,:)} = \text{MSA}_s(k = q = v = \hat{S}_t^{(i,:,:)}) \quad (6)$$

$$= \text{Attn}_s \cdot v_s \quad (7)$$

$$= \text{Softmax}(q_s \cdot k_s^T) \cdot v_s \quad (8)$$

where $\hat{S}_{st} \in \mathbb{R}^{(\tau+1) \times (\frac{WH}{s^2}+1) \times C}$ is the output of MSA_s , q_s , k_s , and v_s denotes key, query, and value features after applying independent linear functions on \hat{S}_t . $\text{Attn}_s \in \mathbb{R}^{(\frac{WH}{s^2}+1) \times (\frac{WH}{s^2}+1)}$ represent a spatial-wise affinity map. We did not apply a down-sampling method on the spatial attention because there was significant performance drop in our preliminary experiments.

Our spatio-temporal split attention decreased the memory usage of the transformer layer by reducing the affinity matrix from $\mathcal{O}(T^2 W^2 H^2)$ to $\mathcal{O}(\tau^2 + W^2 H^2)$. This allows us to explore longer temporal sequence lengths that were infeasible on modern hardware with the vanilla transformer.

3.2.2 Temporal Down-sampling method

The temporal dimension in video clips usually contains redundant information [29]. We introduce compact VidTr (C-VidTr) by applying temporal down-sampling within our transformer architecture. We study different temporal down-sampling methods (pool in Eq. 3) including temporal average pooling and 1D convolutions with stride 2, which reduce the temporal dimension by half (details in Table 4d).

A limitation of these pooling the methods is that they uniformly aggregate information across time but often in video clips the informative frames are not uniformly distributed. We adopted the idea of non-uniform temporal feature aggregation from previous work [29]. Different from

Model	<i>clip_len</i>	<i>sr</i>	Down-sample Layer	τ
VidTr-S	8	8	-	-
VidTr-M	16	4	-	-
VidTr-L	32	2	-	-
C-VidTr-S	8	8	[1,2,4]	[6,4,2]
C-VidTr-M	16	4	[1,2,4]	[8,4,2]
C-VidTr-L	32	2	[1,2,4,6]	[16,8,4,2]

Table 1: Detailed configuration of different VidTr permutations. *clip_len* denotes the sampled clip length and *sr* stands for the sample rate. We uniformly sample *clip_len* frames out of *clip_len* \times *sr* consecutive frames. The configurations are empirically selected, details in [Ablations](#).

previous work [29] that directly down-sample the query using average pooling, we noticed that when a temporal instance is informative, the temporal attention highly activated on a small number of temporal instances, while if a temporal instance is non-informative, the attention is more likely to equally distributed over the length of the clip. Building on this intuition, we propose a topK based pooling (*topK_std* pooling) that orders instances by the standard deviation of each row in the attention matrix. This pooling selects the rows with topK highest standard deviation in the affinity matrix:

$$\text{pool}_{\text{topK_std}}(\text{Attn}_t^{(1,:,:)}) = \text{Attn}_t^{(\text{topK}(\sigma(\text{Attn}_t^{(1,:,:)})\text{}),:)} \quad (9)$$

where $\sigma \in \mathbb{R}^T$ is row-wise standard deviation of $\text{Attn}_t^{(1,:,:)}$. Note that the *topK_std* pooling was applied to the affinity matrix excludes the token ($\text{Attn}_t^{(0,:,:)}$) as we always preserve token for information aggregation (more details in Appendix A). Our experiments show that *topK_std* pooling gives better performance than average pooling or convolution. The *topK_std* pooling can be intuitively understood as selecting the frames with strong localized attention and removing frames with uniform attention.

3.3. Implementation Details

Model Instantiating: Based on the input clip length and sample rate, we introduce 3 base VidTr models (VidTr-S, VidTr-M and VidTr-L). By applying the different pooling strategies we introduce three compact VidTr permutations (C-VidTr-S, C-VidTr-M and C-VidTr-L). To normalize the feature space, we apply layer normalization before and after the residual connection of each transformer layer and adopt the GELU activation as suggested in [13]. Detailed configurations can be found in Table 1. We empirically determined the configuration for different clip length to produce a set of models from low FLOPs and low latency to high accuracy (details in [Ablations](#)).

During **training** we initialize our model weights from ViT-B [13]. To avoid over fitting, we adopted the commonly used augmentation strategies including random crop, random horizontal flip. We trained the model using 64 Tesla

V100 GPUs, with batch size of 6 per-GPU (for VidTr-S) and weight decay of $1e-5$. We adopted SGD as the optimizer but found the Adam optimizer also gives us the same performance. We trained our network for 50 epochs in total with initial learning rate of 0.01, and reduced it by 10 times after epochs 25 and 40. It takes about 12 hours for VidTr-S model to converge, the training process also scales well with fewer GPUs (e.g. 8 GPUs for 4 days). During **inference** we adopted the commonly used 30-crop evaluation for VidTr and compact VidTr, with 10 uniformly sampled temporal segments and 3 uniformly sampled spatial crop on each temporal segment [51]. It is worth mentioning that we can further boost the inference speed of compact VidTr by adopting a single pass inference mechanism, this is because the attention mechanism captures global information more effectively than 3D convolution. We do this by training a model with frames sampled in TSN [50] style, and uniformly sampling N frames in inference (details in Appendix B).

4. Experimental Results

4.1. Datasets

We evaluate our method on six of the most widely used datasets. **Kinetics 400** [7] and **Kinetics 700** [5]. Kinetics 400/700 consists of approximately 240K/650K training videos and 20K/35K validation videos trimmed to 10 seconds from 400/700 human action categories. We report top-1 and top-5 classification accuracy on the validation sets. **Something-Something V2** [21] dataset consists of 174 actions and contains 168.9K training videos and 24.7K evaluation videos. We report top-1 accuracy following previous works [35] evaluation setup. **Charades** [41] has 9.8k training videos and 1.8k validation videos spanning about 30 seconds on average. Charades contains 157 multi-label classes with longer activities, performance is measured in mean Average Precision (mAP). **UCF-101** [42] and **HMDB-51** [27] are two smaller datasets. UCF-101 contains 13320 videos with an average length of 180 frames per video and 101 action categories. The HMDB-51 has 6,766 videos and 51 action categories. We report the top-1 classification on the validation videos based on split 1 for both dataset.

4.2. Kinetics 400 Results

4.2.1 Comparison To SOTA

We report results on the validation set of Kinetics 400 in Table 2, including the top-1 and top-5 accuracy, GFLOPs (Giga Floating-Point Operations) and latency (ms) required to compute results on one view.

As shown in Table 2, the VidTr achieved the SOTA performance comparing with previous I3D based SOTA architectures at lower FLOPs and latency. The VidTr is able to

Model	Input	GFLOPs	Lat.	top-1	top-5
I3D50 [57]	32×2	167	74.4	75.0	92.2
I3D101 [57]	32×2	342	118.3	77.4	92.7
NL50 [51]	32×2	282	53.3	76.5	92.6
NL101 [51]	32×2	544	134.1	77.7	93.3
TEA50 [32]	16×2	70	-	76.1	92.5
TEINet [37]	16×2	66	49.5	76.2	92.5
CIDC [30]	32×2	101	82.3	75.5	92.1
SF50 8×8 [17]	$(32+8) \times 2$	66	49.3	77.0	92.6
SF101 8×8 [17]	$(32+8) \times 2$	106	71.9	77.5	92.3
SF101 16×8 [17]	$(64+16) \times 2$	213	124.3	78.9	93.5
TPN50 [57]	32×2	199	89.3	77.7	93.3
TPN101 [57]	32×2	374	133.4	78.9	93.9
CorrNet50 [49]	32×2	115	-	77.2	N/A
CorrNet101 [49]	32×2	187	-	78.5	N/A
X3D-XXL [16]	16×5	196	-	80.4	94.6
Vanilla-Tr	8×8	89	59.1	77.5	93.2
VidTr-S	8×8	89	65.3	77.7	93.3
VidTr-M	16×4	179	114.9	78.6	93.5
VidTr-L	32×2	351	138.8	79.1	93.9
En-VidTr-S	8×8	130	102.3	79.4	94.0
En-VidTr-M	16×4	220	151.9	79.7	94.2
En-VidTr-L	32×2	392	175.8	80.5	94.6

Table 2: Results on Kinetics-400 dataset. We report top-1 accuracy(%) on the validation set. The ‘Input’ column indicates what frames of the 64 frame clip are actually sent to the network. $n \times s$ input indicates we feed n frames to the network sampled every s frames. Lat. stands for the latency on single crop.

significantly outperform previous SOTA at roughly same computational budget, e.g. at around 200 GFLOPs, the VidTr-M outperform I3D50 by 3.6%, NL50 by 2.1%, and TPN50 by 0.9%. Given the similar performance, the VidTr is also significantly computationally efficient comparing with previous SOTA, e.g. at around 78% top-1 accuracy, the VidTr-S has $6 \times$ less FLOPs than NL-101, $2 \times$ less FLOPs than TPN and 12% less FLOPs than Slowfast-101. We also see that our VidTr outperform I3D based networks at higher sample rate (e.g. $s = 8$, TPN achieved 76.1% top-1 accuracy), this denotes, the global attention learns temporal information more effectively than 3D convolutions. X3D-XXL from architecture search is the only network outperforms us, using architecture search technique for attention based architecture design will be our future work.

4.2.2 Compact VidTr

We evaluate the effectiveness of our compact VidTr with the proposed temporal down-sampling method (Table 1). The results (Table 3) show that the proposed down-sampling strategy reduced about 56% of the computation required by VidTr with only 2% performance drop in accuracy. The compact VidTr complete the VidTr family from small models (only 39GFLOPs) to high performance models (up to 79.1% accuracy). Comparing with previous SOTA compact models [32, 37], our compact VidTr achieves better or sim-

Model	Input	Res.	GFLOPs	Latency(ms)	top-1
TSM [35]	8fTSN	256	10×69	290	74.7
TEA [32]	16×4	256	30×70	-	76.1
3DEff-B4 [16]	16×5	224	10×7	-	72.4
TEINet [37]	16×4	256	30×33	1080	74.9
X3D-M [16]	16×5	224	10×5	1100	74.6
X3D-L [16]	16×5	312	10×19	1900	76.8
C-VidTr-S	8×8	224	30×39	1110	75.7
C-VidTr-M	16×4	224	30×59	1590	76.7
C-VidTr-L	32×2	224	30×79	1740	77.3

Table 3: Comparison of VidTr to other fast networks. We present the number of views used for evaluation and FLOPs required for each view. The latency denotes the total time required to get the reported top-1 score.*

ilar performance with lower FLOPs and latency, including: TEA (+0.6% with 16% less FLOPs) and TEINet (+0.5% with 11% less FLOPs).

4.2.3 Error Analysis

We compare the errors made by VidTr-S and the I3D50 network to better understand the local networks (I3D) and global networks (VidTr) behavior. We provide the top-5 activities that our VidTr-S gain most significant improvement over the I3D50 (details in Appendix D). We find that our VidTr-S outperformed the I3D on the activities that requires long-term video contexts to be recognized. For example, our VidTr-S outperformed the I3D50 on “making a cake” by 26% in accuracy. The I3D50 overfits to “cakes” and often recognize making a cake as eating a cake. We also analyze the top-5 activities where I3D does better than our VidTr-S. Our VidTr-S performs poorly on the activities that need to capture fast and local motions. For example, our VidTr-S performs 21% worse in accuracy on “shaking head” (detailed results in Appendix D).

4.2.4 Ensemble Analysis

Inspired by the findings in our error analysis that the VidTr and I3D seem to have different strengths. We ensemble our VidTr with a light weight I3D50 network by averaging the output values between the two networks. Based on the results in Table 2, the VidTr ensemble with I3D50 achieves a roughly 2% performance improvement on Kinetics 400 with limited additional FLOPs (37G). The performance gain by ensembling the VidTr with I3D is more significant than the improvement by ensembling other networks with I3D (see Appendix C for details).

4.2.5 Ablations

We perform all ablation experiments with our VidTr-S model on Kinetics 400. We used $8 \times 224 \times 224$ input with

*we measure latency of X3D using the authors’ code <https://github.com/facebookresearch/SlowFast/blob/master/projects/x3d/README.md>, which only has models for X3D-M and X3D-L and not the XL and XXL variants

Model	FP.	top-1
Cubic (4×16^2)	23G	73.1
Cubic (2×16^2)	45G	75.5
Square (1×16^2)	89G	77.7
Square (1×32^2)	21G	71.2

(a) Comparison between different patching strategies.

Temp. module	Mem.	top-1
Spatial only	2.1GB	74.7
Jointly	7.6GB	77.5
Temp.-spat.	2.3GB	77.7
Spat.-temp.	2.3GB	77.7

(b) Comparison between temporal modeling strategies.

Init. from	FP.	top-1
T2T [59]	34G	76.3
ViT-B [13]	89G	77.7
ViT-L [13]	358	77.5

(c) Comparison between different backbones.

Configurations	top-1	top-5
Temp. Avg. Pool.	74.9	91.6
1D Conv. [59]	75.4	92.3
STD Pool.	75.7	92.2

(d) Comparison between different down-sample methods.

Layer	τ	FP.	top-1
[0, 2]	[4, 2]	26G	72.9
[1, 3]	[4, 2]	32G	74.9
[2, 4]	[4, 2]	47G	74.9
[6, 8]	[4, 2]	60G	75.3

(e) Compact VidTr down-sampling twice at layer k and $k + 2$.

Layer	τ	FP.	top-1
[1, 2]	[4, 2]	30G	73.9
[1, 3]	[4, 2]	32G	74.9
[1, 4]	[4, 2]	33G	75.0
[1, 5]	[4, 2]	34G	75.2

(f) Compact VidTr down-sampling twice starting from layer 1 and skipping different number of layers.

Table 4: Ablation studies on Kinetics 400 dataset. We use an VidTr-S backbone with 8 frames input for (a,b) and C-VidTr-S for (c,d). The evaluation is performed on 30 views with 8 frame input unless specified. FP. stands for FLOPs.

a frame sample rate of 8, and 30-view evaluation.

Patching strategies: We first compare the cubic patch (4×16^2), where the video is represented as a sequence of spatio-temporal patches, with the square patch (1×16^2), where the video is represented as a sequence of spatial patches. Our results (Table 4a) show that the model using cubic patches with longer temporal size has fewer FLOPs but results to significant performance drop (73.1 vs. 75.5). The model using square patch significantly outperform all the cubic patch based models, because the linear embedding is not enough to represent the shot-term temporal association in the cubic. We further compared the performance of using different patch sizes (1×16^2 vs. 1×32^2), using 32^2 patches lead to $4 \times$ decreasing of the sequence length, which decreases memory consumption of the affinity matrices by $16 \times$, however, using 16^2 patches significantly outperform the model using 32^2 patches (77.7 vs. 71.2). We did not evaluate the model using smaller patching sizes (e.g., 8×8) because of the high memory consumption.

Attention module design: We compare different attention module designs, including spatial modeling only, jointly spatio-temporal modeling module (vanilla-Tr), and our proposed separable-attention (VidTr). We first evaluate an spatio-only transformer. We average the class token for each input frame for our final output. Our results (Table 4b) show that the spatio-only transformer requires the least memory but also has worst performance among different

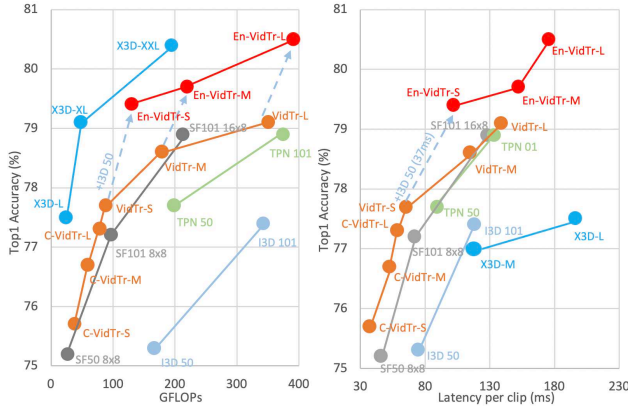


Figure 3: The comparison between different models on accuracy, FLOPs and latency.

attention modules. This shows that temporal modeling is critical for attention based architecture as well. The joint spatio-temporal transformer significantly outperforms the spatio-only transformer but is also heavily memory consuming (T^2 times for the affinity matrices). Our VidTr using separable-attention requires $3.3\times$ less memory at no performance drop. We further test if the order of spatial attention and temporal attention matters, the results (Table 4b) show that the order of temporal attention does not affect the model performance.

Sequence down-sampling comparison: We compare different down-sampling strategy including temporal average pooling, 1D temporal convolution and the proposed STD-based topK pooling method. The results (Table 4d) show that our proposed STD-based down-sampling method outperformed the temporal average pooling and the convolution-based down-sampling strategies that uniformly aggregate information over time.

Backbone generalization: We evaluate our VidTr initialized with different models, including T2T [59], ViT-B, and ViT-L. The results on Table 4c show that our VidTr achieves reasonable performance across all backbones. The VidTr using T2T as the backbone has the lowest FLOPs but also the lowest accuracy. The ViT-L-based VidTr achieve similar performance with the ViT-B-based VidTr even with $3\times$ FLOPs. As showed in previous work [13], transformer-based network are more likely to over-fit and Kinetics-400 is relatively small for ViT-L-based VidTr.

Where to down-sample: Finally we study where to perform temporal down-sampling. We perform temporal down-sampling at different layers (Table 4e). Our results (Table 4e) show that starting to perform down-sampling after the first encoder layer has the best trade-off between the performance and FLOPs. Starting to perform down-sampling at very beginning leads to the fewest FLOPs but

Model	pre-train	frames	top-1	top-5
I3D50 [5]	N/A	32×2	58.7	NA
CSN152 [45]	IG65M	32×2	70.1	89.5
SF101 8×8	K4&K6	$(32 + 8) \times 2$	70.2	89.8
SF101-NL 8×8 [17]	K4&K6	$(32 + 8) \times 2$	70.6	89.7
NUA50 [29]	K400	32×2	68.9	88.9
VidTr-S	K400	8×8	67.3	87.7
VidTr-M	K400	16×4	69.5	88.3
VidTr-L	K400	32×2	70.2	89.0
En-VidTr-L	K400	32×2	70.8	89.4

Table 5: Results on Kinetics-700 dataset. We report top-1 and top-5 accuracy (%) on validation set. K4&K6 stand for Kinetics 400 and Kinetics 600. SF-NL 101 stands for Slowfast network with non-local block [17].

Model	Input	Chad	SSv2	UCF	HMDB
I3D [6]	64×1	32.9	50.0	95.1	74.3
TSM [35]	8(TSN)	-	59.3	94.5	70.7
I3D101 [56]	32×4	40.3	-	-	-
STRG [52]	32×4	39.7	-	-	-
LFB [53]	32×4	42.5	-	-	-
NL-101 [51]	32×4	37.5	-	-	-
TEINet[37]	16 (TSN)	-	62.1	96.7	73.3
SF101 [17]	$(64 + 8) \times 2$	-	60.9	-	-
SF101-NL [17]	$(64 + 8) \times 2$	45.2	-	-	-
X3D-XL [16]	16×5	47.1	-	-	-
VidTr-M	16×4	-	60.1	96.6	74.4
VidTr-L	32×4	43.5	60.2	96.7	74.4
En-VidTr-L	32×4	47.3	-	-	-

Table 6: Results on Charades dataset and something-something-V2 dataset. The evaluation metrics are mean average precision (mAP) in percentage for charades, top-1 accuracy for something-something-V2 (TSN styled dataloader is used), UCF and HMDB.

has a significant performance drop (72.9 vs. 74.9). Performing down-sampling later only has slight performance improvement but requires higher FLOPs. We then analyze how many layers should we skip between two down-sample layers. Based on the results in Table 4f, skip one layer between the two down-samples has the best trade-off. Performing down-samples on consecutive layers (0 skip layers) has lowest FLOPs but the performance decreases (73.9 vs. 74.9). Skipping more layers did not show significant performance improvement but have higher FLOPs.

4.2.6 Run-time Analysis

We further analyzed the trade-off between latency, FLOPs and accuracy. We note that the VidTr achieved the best balance between these factors (Figure 3). The VidTr-S achieve similar performance but significantly less FLOPs compare with I3D101-NL ($5\times$ less FLOPs), Slowfast101 8×8 (12% less FLOPs), TPN101 ($2\times$ less FLOPs), and CorrNet50 ($20\times$ less FLOPs). Note that the X3D has very low FLOPs but high latency due to the use of depth convolution. Our

experiments show that the X3D-L has about $3.6\times$ higher latency comparing with VidTr-S (Figure 3).

4.3. More Results

Kinetics-700 Results: Our experiments show a consistent performance trend on Kinetics 700 (Table 5). The VidTr-S significantly outperformed the baseline I3D model (+9%), the VidTr-M achieved the performance comparable to NUTA-50, Slowfast101 8×8 and the VidTr-L is comparable to previous SOTA slowfast101-nonlocal and NUTA101. There is a small performance gap between our model and Slowfast-NL [17], because Slowfast is pre-trained on both Kinetics 400 and 600 while we only pre-trained on Kinetics 400. Previous finding on VidTr and I3D as being complementary is consistent on Kinetics 700, ensemble VidTr-L with I3D leads to +0.6% performance boost.

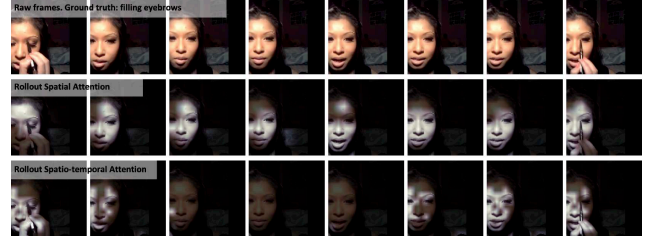
Charades Results: We compare our VidTr with previous SOTA models on Charades. Our VidTr-L outperformed previous SOTA methods LFB and NUTA101, and achieved the performance comparable to Slowfast101-NL (Table 6). The results on Charades demonstrates that our VidTr generalizes well to multi-label activity datasets. Our VidTr performs worse than the current SOTA networks (X3D-XL) on Charades likely due to overfitting. As discussed in previous work [13], the transformer-based networks overfit easier than convolution-based models, and Charades is relatively small. We observed a similar finding with our ensemble, ensembling our VidTr with a I3D network (40.3 mAP) achieved SOTA performance (additional ensemble results including ensembling with CSN-152 to achieve 51.2% mAP are in Appendix C).

Something-something V2 Results: We noticed that the VidTr doesn't work well on the something-something dataset (Table 6), probably because purely transformer based approaches do not model local motion as well as convolutions. This aligns with our observation in our error analysis. Further improving the local motion modeling ability is a good area for future work.

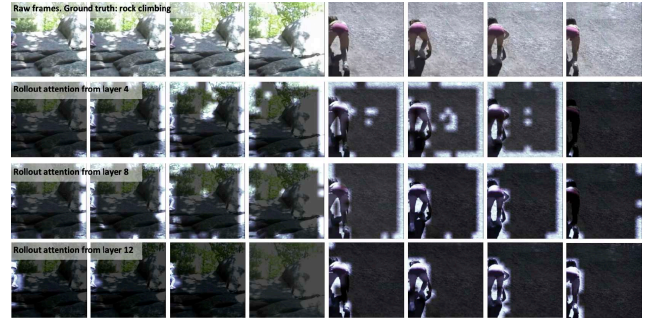
UCF and HMDB Results: Finally we train our VidTr on two small dataset UCF-101 and HMDB-51 to test if VidTr generalizes to smaller datasets. The VidTr achieved SOTA comparable performance with 6 epochs of training (96.6% on UCF and 74.4% on HMDB), showing that the model generalize well on small dataset (Table 6).

5. Visualization and Understanding VidTr

We first visualized the VidTr's separable-attention with attention roll-out method [1] (Figure 4a, implementation details and more samples in Appendix E). We find that the spatial attention is able to focus on informative regions and temporal attention is able to skip the duplicated/non-representative information temporally. We then visualized the attention at 4th, 8th and 12th layer of VidTr (Figure 4b), we found the spatial attention is getting to concentrate bet-



(a) The spatial and temporal attention in VidTr. The attention is able to focus on the informative frames and regions.



(b) The rollout attentions from different layers of VidTr.



(c) Comparison of I3D activations and VidTr attentions.

Figure 4: Visualization of spatial and temporal attention of VidTr and comparison with I3D activation.

ter when it goes to the deeper layer. The attention did not capture meaningful temporal instances at early stages because the temporal feature relies on the spatial information to determine informative temporal instances. Finally we compared the I3D activation map and rollout attention from VidTr (Figure 4c). The I3D mis-classified the catching fish as sailing, as the I3D attention focused on the people sitting behind and water. The VidTr is able to make the correct prediction and the attention showed that the VidTr is able to focus on the action related regions across time.

6. Conclusion

In this paper, we present video transformer with separable-attention, an novel stacked attention based architecture for video action recognition. Our experimental results show that the proposed VidTr achieves state-of-the-art or comparable performance on five public action recognition datasets. The experiments and error analysis show that

the VidTr is especially good at modeling the actions that requires long-term reasoning. Further combining the advantage of VidTr and convolution for better local-global action modeling [36, 54] and adopt self-supervised training [8] on large-scaled data will be our future work.

A. Compact-VidTr implementation details

We provide some details for $topK_{std}$ pooling. We calculate the row-wise standard deviation as:

$$\sigma^{(i)} = \frac{1}{T} \sqrt{\sum_{t=1}^T (Attn_t^{(i,:)} - \mu)^2} \quad (10)$$

$$\mu^{(i)} = \frac{1}{T} \sum_{t=1}^T Attn_t^{(i,:)} \quad (11)$$

where $\sigma \in \mathbb{R}^T$ and $\mu \in \mathbb{R}^T$ are row-wise standard deviation, and mean of $Attn_t^{(1::)}$. Note that the $topK_{std}$ pooling was applied to the affinity map excluded the token $Attn_t^{(1::)}$ as we will always preserve token for information aggregation.

B. Fast VidTr

As a common practice, 3D ConvNets are usually tested on 30 crops per video clip (3 spatial and 10 temporal) that show performance boost while greatly increase the computation cost. The VidTr has been proved that learn long-term global spatio-temporal features better in a video clip, thus we propose to sample the data in TSN style (segment video into N chunks and randomly pick one frame from each chunk). During testing, we uniformly sample N frames from the video regardless the length of the video, and perform single-pass inference (center crop). Such design significantly reduce the inference computation and latency caused by the dense sampling with a very small performance drop (about 2%, see Table A.1). Note that the R2D and I3D based methods do not work well with sparsely sampled frames, mainly because the convolution kernel has limited receptive field and can only aggregate features slowly. If adjacent frames are too far away from each other, the temporal convolution will not be able to establish the temporal relations well. We compare our fast VidTr model with pre-

Model	Input	Res.	GFLOPs	Latency(ms)	Top1
TSM [35]	8fTSN	256	330	170	74.1
3DEff-B4 [16]	16 × 5	224	69	NA	72.4
TEINet [37]	16 × 4	256	990	1080	74.9
X3D-M [16]	16 × 5	224	47	1100	74.6
F-VidTr-S	8 × 8	224	1 × 39	37	72.9
F-VidTr-M	16 × 4	224	1 × 59	53	74.7

Table A.1: Comparison of VidTr to other fast networks. All results from previous methods except for TEINet (30-crops) are based on 10 temporal crop and center spatial crop. The VidTr was achieved by uniformly sample 8/16/32 frames temporally and center-crop spatially.

vious SOTA light-weight models including TSM, TEINet

and models from architecture search such as X3D on Kinetics 400 dataset and report the FLOPs, the latency and top1 accuracy with 10 center crops (Table A.1) The results show that our proposed one-pass inference significantly outperforms the competitors with less FLOPs, lower latency and higher accuracy. The Fast VidTr (16 frames) is able to outperform TSM (+0.6% accuracy, 70% less FLOPs, 68% less latency); TEINet (-0.2% accuracy, 94% less FLOPs, 95% less latency), also note that the reported TEINet score is based on 30 crop evaluation; and X3D-M (+0.1% accuracy, 24% more FLOPs, 96% less latency). The results proves that the VidTr is able to aggregate long-term spatio-temporal features more effectively comparing the 3D ConvNets.

It is worth mentioning that: 1. Even without considering the 10-crop evaluation required for ConvNets to achieve reported scores, the VidTr is still able to inference roughly at same speed comparing with TEINet and significantly faster than X3D. 2. X3D has low FLOPs but high latency mainly due to the heavily use of depth convolution.

C. More Ensemble Results

We provide additional ensemble results on Kinetics 400 (Table A.2) and charades (Table A.3), showing that the VidTr and 3D convolution based models can be complementary to each other, ensemble VidTr and 3D convolution based network significantly outperform the ensemble of any two 3D convolution based models. Our results show that the result level ensemble of I3D-101 and SOTA 3D model TPN-101 lead to about 1% accuracy boost and result level ensemble of VidTr-S with TPN-101 lead to about 3% performance boost. The similar conclusion can be draw from Charades on multi-label activities, where the ensemble of I3D-101 and CSN-152 only gives 2.8% mAP boost, while ensemble of VidTr-L with CSN-152 lead to SOTA (4.8% mAP boost over CSN-152) performance on Charades datasets.

Model	input	Ensemble	input	Top1	Top5
I3D50 [56]	16 × 4	-	-	75.0	92.2
I3D101 [56]	16 × 4	-	-	77.4	92.7
TPN101 [56]	16 × 4	-	-	78.2	93.4
I3D50 [56]	16 × 4	I3D101	16 × 4	77.7	93.2
TPN101[56]	16 × 4	I3D50	16 × 4	78.5	93.3
TPN101 [56]	16 × 4	I3D101	16 × 4	79.3	93.8
VidTr-S	8 × 8	I3D50	16 × 4	79.4	94.0
VidTr-S	8 × 8	I3D101	16 × 4	80.3	94.6
VidTr-S	8 × 8	TPN101	16 × 4	80.5	94.8

Table A.2: More ensemble results on Kinetics-400 dataset. We report top 1 and top5 accuracy (%) on validation set.

Model	Input	Res.	Ensemble	Chad
I3D-Inception [6]	64×1	256	-	32.9
SlowFast-101-NL*	32×4	256	-	44.7
CSN-152*	32×4	256	-	46.4
En-I3D-101	32×4	256	I3D-50	42.1
En-I3D-101	32×4	256	SF-101	47.9
En-I3D-101	32×4	256	CSN-152	49.2
En-VidTr-L	32×4	224	I3D-101	47.3
En-VidTr-L	32×4	224	SF-101	48.9
En-VidTr-L	32×4	224	CSN-152	51.2

Table A.3: Results on Charades dataset. The evaluation metrics are mean average precision (mAP) in percentage. * denotes the result that we re-produced.

D. Error Analysis Details

We show the top 5 classes that gains performance boost from VidTr and top 5 classes that got reduced performance from VidTr. The results (Table A.4) show that the I3D generally performance well on local and fast action while the VidTr works well on actions require long-term temporal information. For example, the VidTr achieved 21.2 % accuracy improvement over I3D on “catching fish” that requires long-term information from the status when the fish is in water to the final status after the fish is caught (Figure A.1a). The VidTr performs worse than I3D on the activities that rely on slight motions (e.g., playing guitar, and shaking head, Figure A.1b)

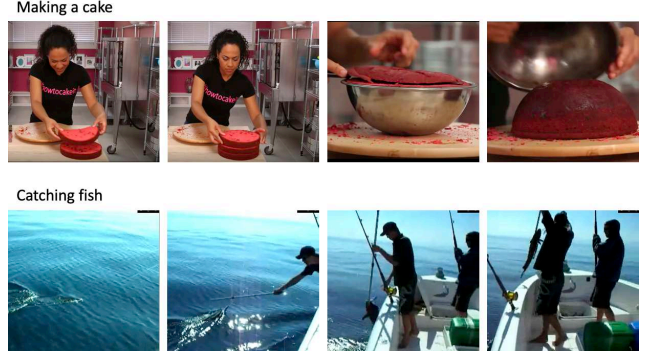
Top 5 (+)	Accuracy gain
making a cake	+26.0%
catching fish	+21.2%
catching or throwing baseball	+20.8%
stretching arm	+19.1%
spraying	+ 18.0 %

(a) Top 5 classes that VidTr works better than I3D.

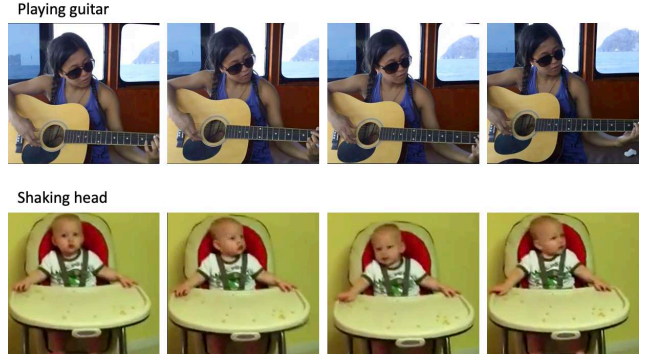
Top 5 (-)	Accuracy gain
shaking head	-21.7%
dunking basketball	-20.8%
lunge	-19.9%
playing guitar	-19.9%
tap dancing	-16.3%

(b) Top 5 classes that I3D works better than VidTr.

Table A.4: Quantitative analysis on Kinetics-400 dataset. The performance gain is defined as the disparity of the top-1 accuracy between VidTr network and that of I3D.



(a) video examples that VidTr performs better than I3D.



(b) video examples that VidTr performs worse than I3D.

Figure A.1: Visualizations of video samples that VidTr works better and I3D works better.

E. Visualization Details

We visualized the VidTr’s separable-attention with attention roll-out method [1]. We multiplied all the affinity matrices between every two encoder layers and get $mask_t \in \mathbb{R}^{(WH+1) \times (T+1) \times (T+1)}$ for the temporal roll-out attention and $mask_s \in \mathbb{R}^{(T+1) \times (WH+1) \times (WH+1)}$ for the spatial roll-out attention. We selected the rows of class token from the roll-out attention for visualization as:

$$mask'_t = mask_t^{(1:,0,1:)} \in \mathbb{R}^{WH \times T} \quad (12)$$

$$mask'_s = mask_s^{(1:,0,1:)} \in \mathbb{R}^{T \times WH} \quad (13)$$

We multiplied $mask'_t$ and $mask'_s$ to represent the spatio-temporal attention for visualize as:

$$mask'_{st} = Re(mask'_t) \times mask'_s \quad (14)$$

where $mask'_{st}$ is the spatio-temporal attention for visualize, and Re denotes a reshape function. We threshold $mask'_s$ and $mask'_{st}$ by only highlighting the top 30% of values of them, and attached them onto the original frames for visualizing the spatio-only and spatio-temporal attentions.

E.1. More Visualizations

We first show more results of the VidTr’s separable-attention with attention roll-out method [1] (Figure A.2). We find that the spatial attention is able to focus on informative regions and temporal attention is able to skip the duplicated/non-representative information temporally.

We then show more results of the attention at 4th, 8th and 12th layer of VidTr (Figure A.3), we found the spatial attention is getting to concentrate better when it goes to the deeper layer. The attention did not capture meaningful temporal instances at early stages because the temporal feature relies on the spatial information to determine informative temporal instances.

Finally we compared the I3D activation map and rollout attention from VidTr (Figure A.4). The I3D mis-classified the catching fish as sailing, as the I3D attention focused on the people sitting behind and water. The VidTr is able to make the correct prediction and the attention showed that the VidTr is able to focus on the action related regions across time.

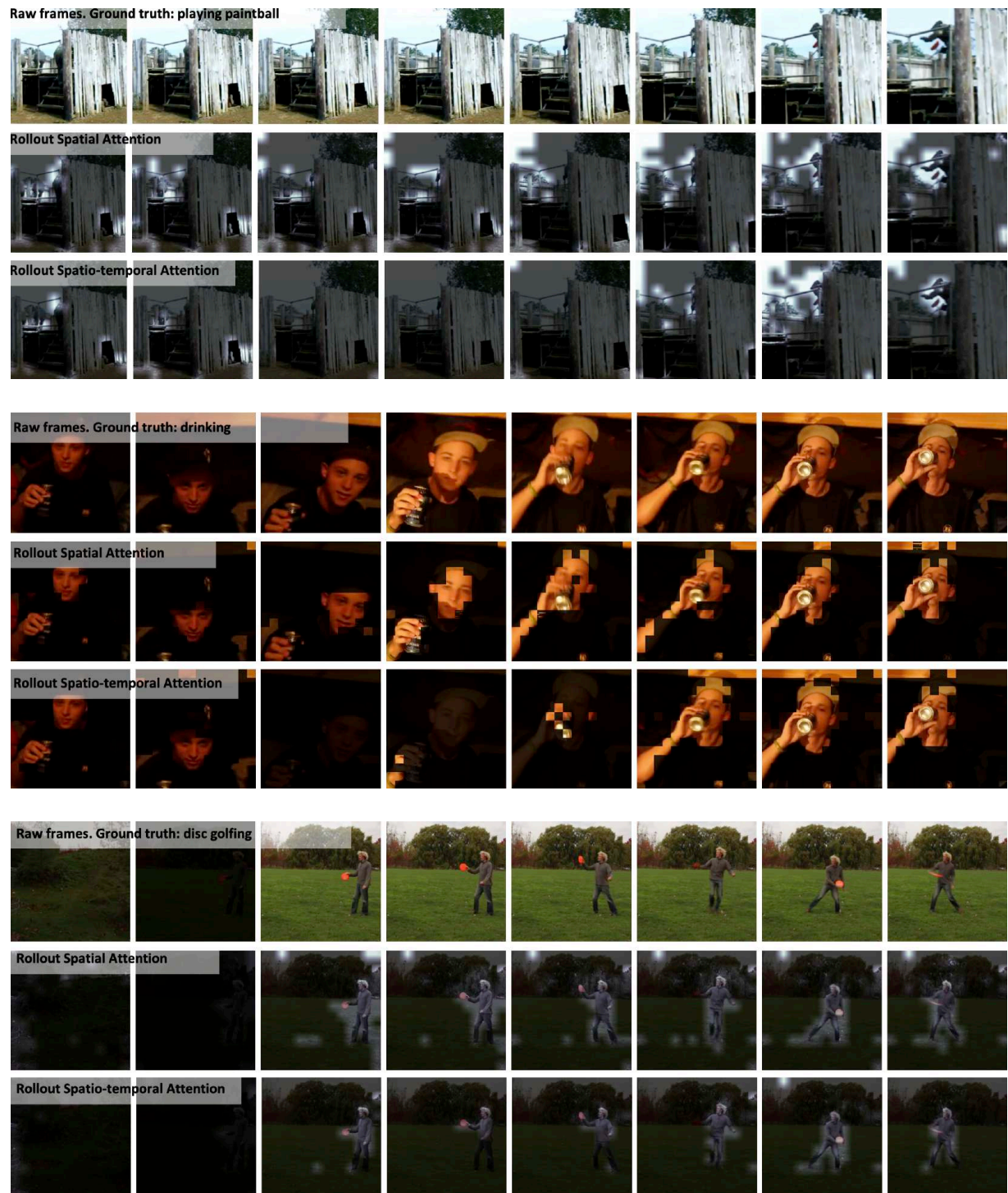


Figure A.2: The spatial and temporal attention in Vidtr. The attention is able to focus on the informative frames and regions.

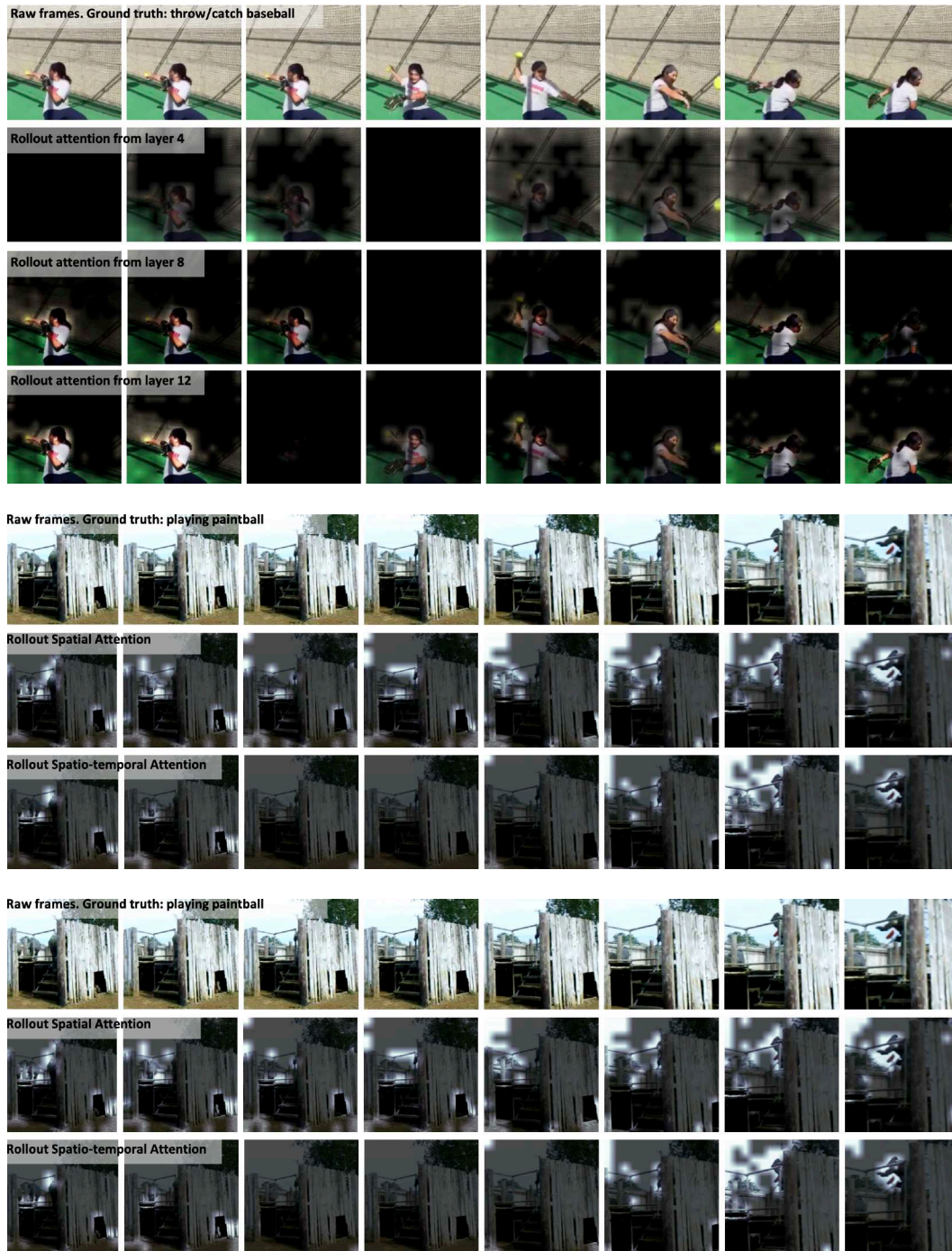


Figure A.3: The rollout attentions from different layers of VidTr.

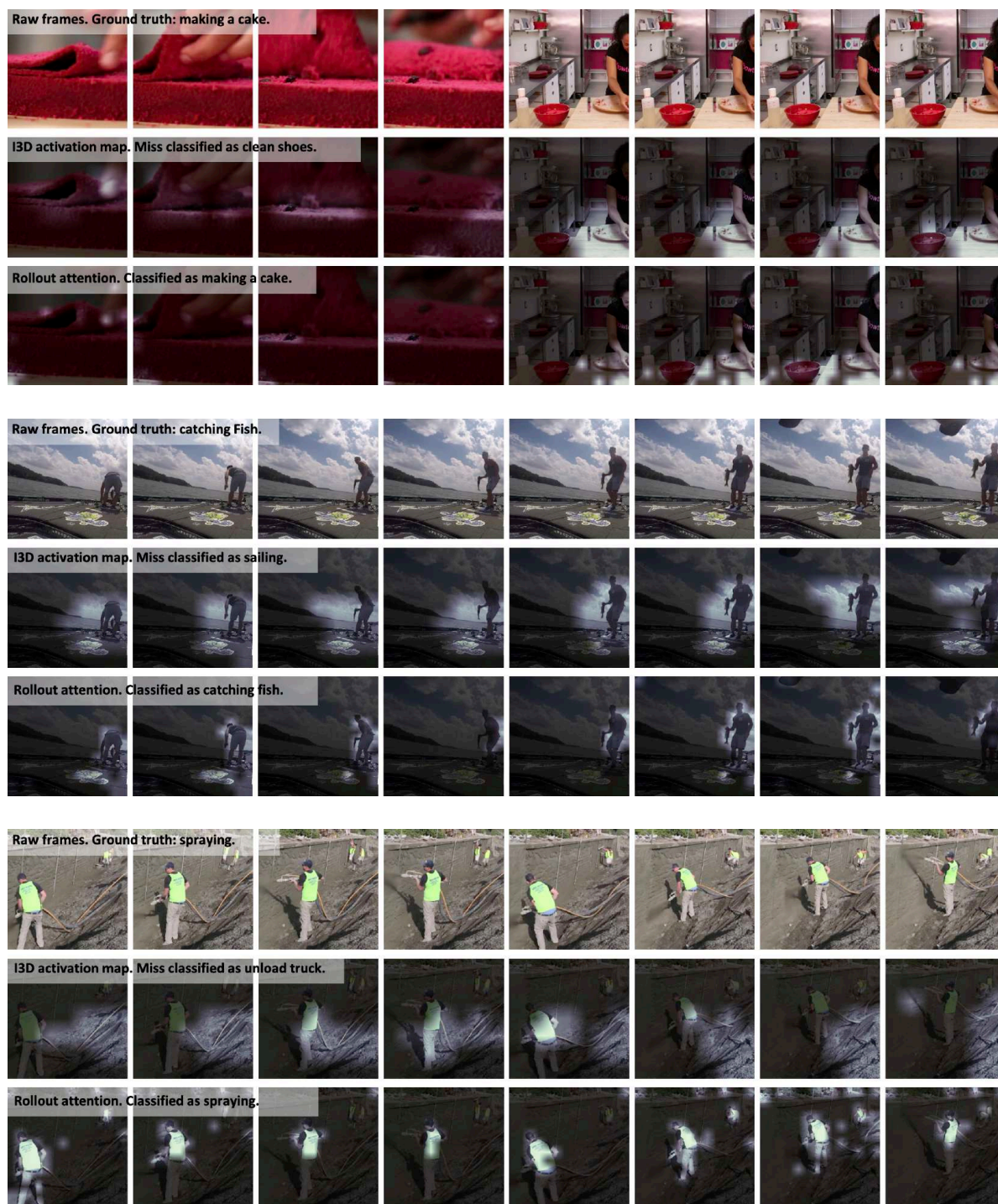


Figure A.4: Comparison of I3D activations and VidTr attentions.

References

- [1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4190–4197, 2020. [2](#), [8](#), [11](#), [12](#)
- [2] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020. [3](#)
- [3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. [3](#)
- [4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. [2](#)
- [5] Joao Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. [5](#), [7](#)
- [6] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. [2](#), [7](#), [11](#)
- [7] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, July 2017. [5](#)
- [8] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised visual transformers. *arXiv preprint arXiv:2104.02057*, 2021. [9](#)
- [9] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Multi-fiber networks for video recognition. In *Proceedings of the european conference on computer vision (ECCV)*, pages 352–367, 2018. [2](#)
- [10] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020. [2](#)
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. [1](#)
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019. [2](#), [3](#)
- [13] 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](#), [6](#), [7](#), [8](#)
- [14] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. *arXiv preprint arXiv:2101.08833*, 2021. [2](#)
- [15] Quanfu Fan, Chun-Fu Chen, Hilde Kuehne, Marco Pistoia, and David Cox. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. *arXiv preprint arXiv:1912.00869*, 2019. [2](#)
- [16] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 203–213, 2020. [2](#), [5](#), [6](#), [7](#), [10](#)
- [17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. [1](#), [2](#), [5](#), [7](#), [8](#)
- [18] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision (ECCV)*, volume 5. Springer, 2020. [2](#)
- [19] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–253, 2019. [2](#), [3](#)
- [20] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. ActionVLAD: Learning Spatio-Temporal Aggregation for Action Classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [2](#)
- [21] 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*, volume 1, page 3, 2017. [5](#)
- [22] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018. [2](#)
- [23] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. [2](#)
- [24] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2012. [2](#)
- [25] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. STM: SpatioTemporal and Motion Encoding for Action Recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. [2](#)
- [26] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. [2](#)
- [27] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pages 2556–2563. IEEE, 2011. [5](#)
- [28] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 159–166, 2016. [2](#)

- [29] Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Hao Chen, and Joseph Tighe. Nuta: Non-uniform temporal aggregation for action recognition. *arXiv preprint arXiv:2012.08041*, 2020. 1, 2, 4, 7
- [30] Xinyu Li, Bing Shuai, and Joseph Tighe. Directional temporal modeling for action recognition. In *European Conference on Computer Vision*, pages 275–291. Springer, 2020. 1, 2, 5
- [31] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: Temporal Excitation and Aggregation for Action Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2
- [32] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020. 5, 6
- [33] Yingwei Li, Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. VLAD3: Encoding Dynamics of Deep Features for Action Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2
- [34] Zekang Li, Zongjia Li, Jinchao Zhang, Yang Feng, Cheng Niu, and Jie Zhou. Bridging text and video: A universal multimodal transformer for video-audio scene-aware dialog. *arXiv preprint arXiv:2002.00163*, 2020. 2
- [35] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 2, 5, 6, 7, 10
- [36] 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. 9
- [37] Zhaoyang Liu, Donghao Luo, Yabiao Wang, Limin Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Tong Lu. TEINet: Towards an Efficient Architecture for Video Recognition. In *The Conference on Artificial Intelligence (AAAI)*, 2020. 2, 5, 6, 7, 10
- [38] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Aselsmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 3
- [39] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Tiny video networks. *arXiv preprint arXiv:1910.06961*, 2019. 2
- [40] Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11966–11973, 2020. 2
- [41] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ivan Laptev, Ali Farhadi, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv e-prints*, 2016. 5
- [42] Khurram Soomro, Amir Roshan Zamir, and M Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2012. 5
- [43] 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, 3
- [44] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2
- [45] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5552–5561, 2019. 2, 7
- [46] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1
- [47] Amin Ullah, Jamil Ahmad, Khan Muhammad, Muhammad Sajjad, and Sung Wook Baik. Action recognition in video sequences using deep bi-directional lstm with cnn features. *IEEE access*, 6:1155–1166, 2017. 2
- [48] 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
- [49] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 352–361, 2020. 5
- [50] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *The European Conference on Computer Vision (ECCV)*, 2016. 2, 5
- [51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 1, 2, 5, 7
- [52] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018. 7
- [53] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019. 1, 7
- [54] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 9
- [55] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 2
- [56] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal Pyramid Network for Action Recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 7, 10
- [57] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 591–600, 2020. 5

- [58] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. [2](#)
- [59] 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. [6](#), [7](#)
- [60] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4694–4702, 2015. [2](#)
- [61] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal Relational Reasoning in Videos. In *The European Conference on Computer Vision (ECCV)*, 2018. [2](#)
- [62] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8739–8748, 2018. [2](#)