Are Attention Maps Richer than we Imagined for Action Recognition?

Tanay Agrawal* Abid Ali* Antitza Dantcheva Francois Bremond
INRIA Sophia Antipolis – Méditerranée, France
{firstname.lastname}@inria.fr

Abstract

Deep learning models are becoming more general and robust by the day. Specifically, image foundation models have recently shown exponential growth. In this work, we introduce a way to exploit this growth in the field of video classification. The basic idea here is that if we have a good understanding of space, we should not require complicated spatio-temporal processing. We introduce Attention Map (AM) flow, a way to identify the location of local changes between two frames in a video, without adding additional parameters specifically for it. We utilise adapters, which have been growing in popularity in the field of parameter-efficient transfer learning. These help us incorporate AM flow in a pretrained image model without the need of fine-tuning it. With just these changes and minimal temporal processing, an image model is able to achieve state-of-the-art results on popular action recognition datasets with low training time and requiring minimal pretraining. This work explores the theory behind this idea and the intricacies involved. Through relevant experiments, we show the efficacy of this method and discuss various ideas to take this work forward. We use kinetics-400, something-something v2 and Toyota smarthome datasets and achieve state-of-the-art or comparable results. We also show that video models suffer from extensive pretraining on multiple datasets and a large training time, but our work answers these problems. action-recognition transformers image-to-video-models

1. Introduction

Foundation models excel beyond their initial training tasks. We demonstrate that image foundation models contain sufficient spatial information and don't require conversion to video models for effective video understanding. With minimal pretraining, they provide a resource-efficient alternative to video processing backbones while requiring significantly less training time compared to SOTA approaches.

Video foundation models, despite their capabilities, demand substantial pretraining data and computational resources, limiting accessibility and widespread adoption. Our approach addresses these limitations while preserving spatio-temporal learning benefits through adapters [50, 19] - parameter-efficient modules added to frozen pre-trained backbones that adapt the network distribution to new tasks or modalities without fully inflating 2D layers to 3D.

We introduce Attention Map (AM) flow, calculated as the absolute difference between attention maps from two video frames, which provides encoded motion information with minimal computational overhead. This approach leverages recent advances in image foundation models like DINOv2 [46] and Hiera [53] with their strong spatial attention capabilities. By combining adapters and AM flow, we enable efficient spatio-temporal reasoning for video classification.

AM flow can be concatenated to the adapter inputs across attention modules. The resulting downsampled embeddings provide rich spatial information about motion, improving backbone performance and enabling direct classification using temporal processing models like LSTMs, transformers, or TCNs.

We achieve SOTA results on three datasets: Kinetics-400 [5], Something-Something v2 [17], and Toyota Smarthome [8].

To summarise, the contributions of our work are:

- We introduce AM flow, a way to add temporal attention to image models with adapters, without additional trainable parameters.
- We explore various scenarios and offer different ways to define AM flow and also the model architecture depending on the use case.
- We demonstrate how to exploit adapters to incorporate AM flow into image models and also for global temporal processing.
- We achieve SOTA performance on three datasets - Kinetics-400, Something-something v2 and Toyota

*These authors contributed equally to this work.

Smarthome utilising AM Flow and basic temporal processing using just LSTMs.

2. Related Work

2.1. Image and video foundation models

Pretraining ConvNets [28] or Transformers [61] on large datasets like ImageNet-65M [16, 42, 57], Instagram [43], or JFT [73] has been a prevalent strategy for visual recognition tasks. Self-supervised approaches have introduced Image Foundation Models (IFMs) [3] that learn general-purpose representations from unlabeled data. For this work, we use DINOv2 [46] with its robust spatial features, though CLIP [52] and Hiera [53] could work similarly.

Video Foundation Models (VFMs) are typically built on IFMs [71, 68, 74, 29, 31, 62, 67, 1]. Despite impressive results from Florence [72], CoCa [71], UniFormerV2 [29], and MTV [68], these models struggle with high temporal variance actions [17, 55]. InterVideo [65], VIOLET [15], All-in-One [63], and LAVENDER [31] perform well on multimodal benchmarks but struggle with video-only tasks. Our method facilitates temporal processing module integration with demonstrated performance improvements.

2.2. Video Classification

Deep learning algorithms dominate action recognition in unconstrained videos, leveraging datasets like Kinetics [5] and Something-Something [17]. Model architectures have evolved from CNNs [26, 13, 14, 34, 40, 47, 60, 66, 64] to Transformers [1, 12, 61, 30, 33]. Since Vision Transformer (ViT)[11] emerged, researchers have extended pretrained image models for video understanding by initializing video transformer components [1, 2, 68, 11] or inflating video transformers [38]. While effective, these approaches require extensive finetuning. Our method achieves comparable performance with approximately 10 times fewer training epochs.

Previous work [18, 35, 51] explored intermediate features for classification but overlooked the rich relational information in attention maps. Our approach leverages this information to achieve superior performance.

2.3. Parameter Efficient Transfer Learning

Efficient tuning has gained interest in NLP with large pretrained language models across downstream tasks [20, 6, 22, 45, 58]. Parameter-efficient fine-tuning techniques like adapters are now common in computer vision [48, 49, 6, 23, 36, 44, 58], showing pretrained image models can effectively learn video tasks. Previous approaches added decoders with 3D convolution and cross-frame attention [33], placed 3D convolution between adapter layers [38] (causing inefficiencies), or required text-encoder branches [21, 37]. Recent work includes ST-adapter [48] with depthwise convolution for temporal pooling and dual pathway adapters [49]. We argue pretrained image models already provide rich spatial features needing only minimal temporal attention modifications for video classification, and present such a method.

3. Methodology

In this section, we walk the reader through the details of our architecture, illustrated in Figure 1. We take ViT trained using Dinov2 [46] as our image model and freeze its weights. Then, scaled parallel adapters are added to the model, and AM flow is concatenated with adapters across multi-head self attention (MHSA). Various temporal processing modules along with the image model branch give output logits which are averaged, giving the final output. Details are given below.

3.1. Adding AM Flow and Modifying Attention

Building upon the previous step, we have a frozen image model with adapters added to it for finetuning, but, we are missing the benefit of spatio-temporal attention in video models. To answer this, we introduce the AM flow. It examines two temporal frames to detect local movement and informs the adapters so that they can account for it when adjusting the spatial attention in the image model. Thus, we achieve spatio-temporal attention with only a minuscule number of parameters added to the original model.

To compute AM flow, we use the attention maps from a transformer block. An attention map corresponding to the frame t ($X_{A_t} \in \mathbb{R}^{N \times N}$) is a matrix of size $N \times N$, where N is the number of embeddings in the input to the transformer block. For us (and also commonly in most transformer-based architectures), N is equal to the number of patches of the input. So, the attention map is a matrix in which each entry signifies the relationship of two particular patches of the input. Taking the absolute difference of this matrix for two consecutive inputs results in a matrix where each entry signifies the amount of change in the relationship of the two particular patches. Taking a row-wise sum of the transpose of this matrix, we get the amount of change in each input patch. The softmax and addition are performed along different dimensions of the attention map. Addition aggregates the importance of a particular pixel with respect to all the others and softmax normalises it. So, the difference of normalised, aggregated significance of pixels for two frames gives AM Flow. Mathematically, this can be expressed as:

$$X_{A_t} = \frac{Q_t K_t^T}{\sqrt{d_K}} \tag{1}$$

As shown in Figure 1 (Yellow), X_{A_t} is an intermediate term used to compute AM flow. The rest of the symbols have the

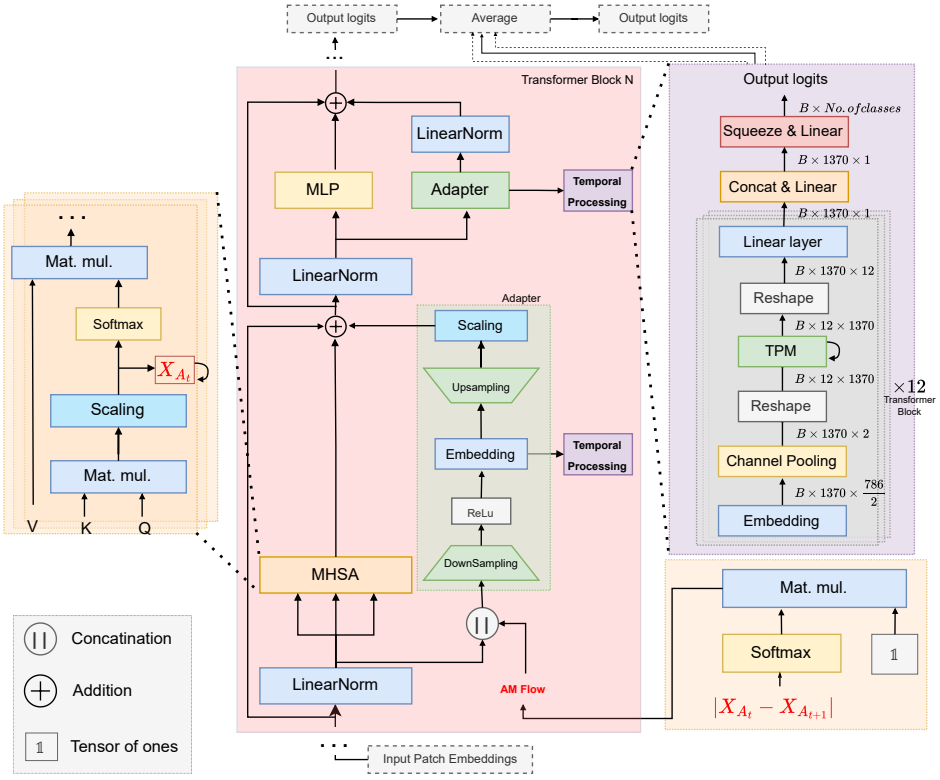

Figure 1: (Red) The middle part of the figure shows the frozen image model (ViT) with trainable additions: adapters in green and temporal processing module in violet. It takes input (Yellow) On the left, X_{A_t} is shown as computed inside $MHSA$. On the right, X_{A_t} and $X_{A_{t+1}}$ are used to compute $AMflow$. t and $t+1$ signify different time-steps for the input frames. (Violet) shows the global temporal processing module and the classification head added to it. All the logits received from the temporal processing modules and the frozen model branch are averaged to get the final classification logits.

common meaning of self-attention.

$$AMflow = softmax(|X_{A_t} - X_{A_{t+1}}|)^T \cdot \mathbb{1} \quad (2)$$

$\mathbb{1} \in \mathbb{R}^{N \times 1}$ is a matrix of the same shape as Q, K, V, but filled with ones. The multiplication operation is equivalent to the row-wise sum of a matrix. We use $\mathbb{1}$ in place of V (which is commonly used in attention) as AM flow needs to highlight the position of the relevant patches and contextual information provided by the input is not required by it.

3.1.1 Handling irrelevant motion - Aligning Encoder

AM flow as described above is based on the assumption: the camera is stable. In the case of moving cameras, the computed flow would not be very useful unless the temporal difference between two consecutive frames is reduced. This can be solved in preprocessing, for example, by taking a crop of the subject performing the action using foreground segmentation. Otherwise, this would have to be corrected by the model. Therefore, before computing the absolute difference, we take the row-wise sum of the attention maps and pass it through a transformer encoder (called Aligning Encoder). [1] The transformer encoder learns to align the relevant information in attention maps and, in turn, disregards the rest. It still has information about global motion owing to backpropogation from the temporal processing module. The weights of the aligning encoder are shared between the two frames used to compute AM flow. Figure 2 illustrates the aligning encoder added to self-attention in a transformer block of the frozen image model. Mathematically expressed as:

$$X_{A_t} = AligningEncoder(softmax(\frac{Q_t K_t^T}{\sqrt{d_K}})^T \cdot \mathbb{1}) \quad (3)$$

[1] We use fast attention[27] for the aligning encoder.

Method	Pretrain	#F	Model # Params (M)	Trainable # Params (M)	FLOPs (T)	SSV2 Top-1	SSV2 Top-5
Specialised backbone with supervised pretraining							
VideoSwin-B [39]	IN-21K+K400	32	89	89	1.0	69.6	92.7
MViTv2-L [33]	IN-21K+K400	40	213	213	8.5	73.3	92.7
Vanilla ViT with self-supervised pretraining for 1600 epochs.							
VideoMAE-L [59]	-	16	305	305	3.6	74.3	94.6
Well-prepared ViT with plug-and-play modules.							
TimeSformer-L [2]	IN-21K	96	121	121	7.1	62.3	81.0
MTV-B [68]	IN-21K+K400	32	310	310	11.2	68.5	90.4
CoVeR [75]	JFT-3B+KMI	16	431	431	17.6	70.8	-
Full tuning							
UniFormerV2-B [29]	CLIP-400M	16	163	163	0.6	69.5	92.3
UniFormerV2-L [29]	CLIP-400M	16	574	574	8.0	72.1	93.6
Parameter Efficient Tuning							
VPT-B [21]	CLIP-400M	8	92	6,0	0.5	36,2	61,1
AdaptFormer-B [6]	CLIP-400M	8	94	8	0.5	51.3	70.6
ST-Adapter [48]	CLIP-400M	32	97	11	2	69.5	92.6
DUALPATH-B [49]	CLIP-400M	16	99	13	0.7	70.3	92.9
DUALPATH-L [49]	CLIP-400M	48	336	33	1.9	72.2	93.7
Ours - AM/12, TCN (Dinov2)	**IN-21K**	**24**	**86+28+45+54**	**28+45+54**	**5.1**	**74.8**	**95.0**
Ours - AM/12, TCN (CLIP)	IN-21K	24	86+28+45+54	28+45+54	5.1	73.5	94.7
Ours(ViT-B)-AM/12, LSTM (Dinov2)	IN-21K	8	86+30+45+360	30+45+360	4.6	58.3	82.8

Table 1: Comparison with the SOTA on SSv2. The colours represent # of parameters in the modules. AM flow and linear layers, Aligning encoder, and Temporal Processing Module. AM/12 signifies that AM flow is added to each transformer block.

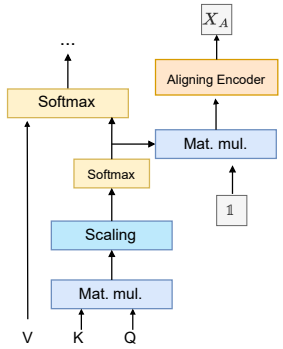

Figure 2: This figure shows how AM flow (X_A) is computed in case there is a camera movement or motion in the background

The variables have the same references as above. The aligning encoder is a simple transformer encoder with 12 heads.

$$AMFlow = |X_{A_t} - X_{A_{t+1}}| \quad (4)$$

3.2. Classification Head and Temporal Processing

With the above step, we have successfully merged local temporal attention with spatial attention in a frame. To extend this to video classification, we need global temporal processing. It is very interesting to note here that with the rich downsampled embedding from the adapter, very simple temporal processing allows us to achieve SOTA results. Figure 1 (violet) demonstrates the temporal and classification module and is self-explanatory. The formulation of the input embedding can be expressed as:

$$E = ReLU(W_{down}(LayerNorm(X)||AMFlow)) \quad (5)$$

E is the embedding and W_{down} is the downsampling layer of the adapter.

Thus, we are able to obtain SOTA results even with LSTMs for temporal processing. We would like the reader to note that LSTMs are just used as a proof of concept, and we do not claim that this is the best configuration for performance. For robustness, we show results with transformer encoder and TCN.

The final step of classification is managing the 25 outputs we get (for ViT-L). 2 from each transformer block as there are 2 adapters in each block and 1 from the frozen image model taking the final frame as input. There are 12 blocks in ViT-B. We initially added a linear voting layer, but it just makes training unstable. Average pooling works well here. An interesting thing to note is that even a randomly initialised frozen linear layer works well in place of aver-

Method	Backbone	Pretrain	Views	Model # Params (M)	Trainable # Params (M)	FLOPs (T)	K400 Top-1	K400 Top-5
Specialised backbone with supervised pretraining								
VideoSwin-L [39]	Swin-L	IN-21K	32 × 3 × 4	197	197	7.2	83.1	95.9
MViT-L [33]	MViTv2-L	IN-21K	40 × 3 × 5	218	218	42.4	86.1	97.0
Vanilla ViT with self-supervised pretraining for 1600 epochs.								
VideoMAE-L [59]	ViT-L		40 × 3 × 4	305	305	47.5	86.1	97.3
Well-prepared ViT with plug-and-play modules.								
TimeSformer-L [2]	ViT-B	IN-21K	96 × 3 × 1	121	121	7.1	80.7	94.7
X-CLIP-L [44]	ViT-L	CLIP-400M	16 × 3 × 4	430	430	37.0	87.7	97.4
MTV-H [68]	ViT-H+B+S+T	IN-21K+WTS-600M	32 × 3 × 4	1000+	1000+	44.5	**89.1**	98.2
Full tuning								
UniFormerV2-B [29]	ViT-B	CLIP-400M+K710-0.66M	8 × 3 × 4	115	115	1.6	85.6	97.0
UniFormerV2-L [29]	ViT-L	CLIP-400M+K710-0.66M	32 × 3 × 4	354	354	16.0	**89.7**	98.3
Parameter Efficient Tuning								
ST-Adapter [48]	ViT-B	CLIP-400M	32 × 3 × 1	93	7	1.8	82.7	96.2
DUALPATH-B [49]	ViT-B	CLIP-400M	32 × 3 × 1	96	10	0.7	85.4	97.1
DUALPATH-L [49]	ViT-L	CLIP-400M	32 × 3 × 1	330	27	1.9	87.7	97.8
Ours - AM/12, LSTM	**ViT-B(Dinov2)**	**IN-21K**	**8 × 3 × 1**	**86+32+45+360**	**32+45+360**	**5.3**	**88.8**	**98.2**
Ours - AM/12, Transformer	**ViT-B(Dinov2)**	**IN-21K**	**8 × 3 × 1**	**86+32+45+103**	**32+45+103**	**6.9**	**89.1**	**98.3**
Ours - AM/12, Transformer	**ViT-B(Dinov2)**	**IN-21K**	**32 × 3 × 1**	**86+32+45+103**	**32+45+103**	**13.8**	**89.6**	**98.4**
Ours - AM/12, LSTM	**-**	**IN-21K**	**8 × 3 × 1**	**86+32+45+360**	**32+45+360**	**5.3**	**78.3**	**91.5**

Table 2: Comparison for K400 with the SOTA methods. The colours represent # of parameters in the modules. AM flow and linear layers, Aligning encoder, and temporal module. AM/12 signifies that AM flow is added to all 12 transformer blocks.

age pooling. The scaling parameter in the adapters learns to work around the randomly assigned weights.

4. Experiments and Observations

4.1. Specific Details and Comparison to SOTA

In this section, we compare the performance of our model against the baselines and the respective SOTA [68, 29, 59, 9] for the datasets.

4.1.1 SSv2

We achieve the SOTA results for Ssv2 as in Table 1. Since this is a more challenging dataset and 8 frames are not enough to capture the essence of the actions, we use TCN in place of LSTM, with 24 frames as input. We present the results for both settings. We obtain state-of-the-art results and also show that the total number of parameters can be reduced using a different temporal processing module. We also show results using CLIP backbone in place of dinov2 and are discussed in 4.2. *Since there is a lot of motion in the frames and there are multiple objects of interest, the computation of AM flow is utilised along with the aligning encoder. Adapters and temporal processing modules are added to each transformer block.*

4.1.2 K400

We report the SOTA comparison for K400 in Table 2. We achieve the best results with 32 frames and using a transformer encoder for temporal processing (discussed in 4.2). But we also achieve high performance when using only 8 frames and pretraining just on ImageNet dataset and using LSTMs. We lag behind Uniformerv2-L, but compared to them, we have negligible pretraining data and require fewer frames (8 for us vs. 32), training time (480 GPU hours for us vs. 9600), flops (5.3T for us vs. 16T), and backbone (ViT-B for us vs. ViT-L) to achieve comparable results. Comparison against CoCa and MTV-H is not fair, as they have 1B+ parameters. Our number of parameters is high, as the aligning encoder and LSTMs (or transformer enocoder) are not optimised for their function and are only used as a proof of concept. Therefore, our total number of parameters is reducible but is left for future work (further discussed in 4.2). *The aligning encoder and AM flow computation are employed since the camera is moving. Each transformer block has temporal processing modules and adapters attached to it.*

Method	RGB	Skeleton	Pretrain	# F	*CS*
Separable STA [7] 64	✓	✓	K400	-	54.2
VPN [10]	✓	✓	K400	64 +pose	60.8
MMNet [4]	✓	✓	-	-	70.1
VPN++ [9]	✓	✓	-	64 +pose	69.0
ST-GCN [69]		✓	Scratch	-	53.8
2s-AGCN [56]		✓	Scratch	-	60.9
MS-G3D [41]		✓	Scratch	-	61.1
UNIK [70]		✓	Posetics	-	64.3
I3D [32]	✓		K400	64	53.4
AssembleNet++[54]	✓		K400	-	63.6
Ours - AM/2(1,12), LSTM	✓		**IN-21K**	**8**	**70.2**

Table 3: Results of Toyota Smarthome. AM/2 signifies that AM flow is added to two transformer blocks (1,12). Thus it has 1/6th the number of parameters for temporal and aligning encoders as compared to kinetics-400 and SSv2.

Model	Top-1
Ours	88.8
w/o AM Flow	74.3

(a) Impact of AM flow

Model	Top-1
Ours	88.8
w/o Aligning Encoder	72.7

(b) Impact of Aligning Encoder

Model	Top-1
Ours	88.8
Ours with hyperformer	84.2

(c) Type of Adapter

Table 4: Tables for Component Analysis (Experiments performed on K400)

4.1.3 Toyota Smarthome

We report the comparison for Smarthome in Table 3 achieving SOTA results. This shows that our method also adapts well to small datasets. *The camera is stable here and since these are daily-action video for old-people, the subjects do not move much in the frame. Therefore, the vanilla AM flow (without aligning encoder) is used.*

4.2. Additional Experiments and Discussion

This section covers ablation studies and other experiments to validate the efficacy of the contributions.

Number of parameters and changing the temporal processing module and the input frames. The choice of temporal processing module is not focused on in this work as even with LSTMs, we get enough performance to prove the efficacy of the additions. TCN and transformer encoder provide the same performance with fewer parameters. The additions also do not have to be done at each step and this further reduces the number of trainable parameters considerably. This is discussed in the following part of this subsection. But irrespective of the module used, as discussed above, we require very less training and have negligible pretraining requirements compared to the SOTA.

LSTMs do not have enough processing power to handle more than 8 frames as input. Since SSv2 has complex temporal relations, we use TCN with 24 frames to achieve better performance. Thus, even though this is not explored in detail as it is not a direct contribution, a better module and a higher number of frames would give better results, as shown by this experiment in Table 1. Using more number of frames for K400 also improves performance showing that the model is scalable.

Impact of AM Flow. Results in Table 4(a). We observe that the addition of AM flow greatly improves performance. Two factors affect this improvement: 1) Concatenation of AM flow to the adapter input pushes the downsampled embedding to learn semantic information about the input, thus parallel adapters overcome their shortcoming over serial adapters for temporal processing (sec 3.1.2). 2) As intended, it adds local temporal attention to the model.

Impact of aligning encoder. Results in Table 4(b). With irrelevant motion in the frames, AM flow is noisy, as the patches with change are not only because of the action but also due to motion. The aligning encoder resolves this issue (as demonstrated in the Supplementary Materials).

Changing type of adapter. Results in Table 4(c). There are various variations of adapters, such as hyperformer [25] and compacter [24]. We compare against hyperformer here.

Training from scratch. To make sure that a well-chosen pretrained image model is important for our work, we train from scratch with the same architecture and achieve 78.3% accuracy for K400 as compared to 88.8% in Table 2. This shows our method succeeds in utilising the spatial awareness learnt by pretrained image models and it is an important step.

Changing the backbone. For completeness, we use CLIP as a backbone in place of dino v2 in Table 1 and show that we still achieve SOTA results and that our method is not specific to the pretraining method. We show results with Hiera [53] in the supplementary materials. The important thing is to have good spatial features from the backbone, irrespective of their nature.

5. Conclusion and Future Work

In this paper, we introduce a novel image-to-video transfer learning model using two key innovations: (1) infusing local temporal attention into spatial attention via our new AM flow concept, and (2) employing adapters to incorporate AM flow into frozen pretrained image models while providing downsampled embeddings for global temporal processing.

Our approach performs well on both large datasets (K400, SSv2) with minimal training steps and only ImageNet pretraining, and adapts effectively to small datasets as demonstrated by SOTA results on Toyota Smarthome. Ablation studies validate our design choices, achieving SOTA or comparable performance across all three datasets with significantly reduced training time.

Future work could explore enhancing AM flow with additional memory for more nuanced temporal information, extending to video detection tasks, optimizing network design for lighter models, comparing against other PETL techniques, and developing more efficient alternatives to the resource-intensive aligning encoder.

References

[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.

[2] G. Bertasius, L. Torresani, M. Paluri, and D. Tran. Is spacetime attention all you need for video understanding? In *International Conference on Learning Representations*, 2021.

[3] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[4] X. Bruce, Y. Liu, X. Zhang, S.-h. Zhong, and K. C. Chan. Mmnet: A model-based multimodal network for human action recognition in rgb-d videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3522–3538, 2022.

[5] J. Carreira and A. 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.

[6] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *NeurIPS*, 2022.

[7] P. Climent-Perez and F. Florez-Revuelta. Improved action recognition with separable spatio-temporal attention using alternative skeletal and video pre-processing. *Sensors*, 21(3):1005, 2021.

[8] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca. Toyota smarthome: Real-world activities of daily living. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 833–842, 2019.

[9] S. Das, R. Dai, D. Yang, and F. Bremond. Vpn++: Rethinking video-pose embeddings for understanding activities of daily living. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):9703–9717, 2021.

[10] S. Das, S. Sharma, R. Dai, F. Bremond, and M. Thonnat. Vpn: Learning video-pose embedding for activities of daily living. In *Proc. Eur. Conf. Comput. Vis.*, pages 72–90. Springer, 2020.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[12] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[13] C. 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.

[14] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[15] T.-J. Fu, L. Li, Z. Gan, K. Lin, W. Y. Wang, L. Wang, and Z. Liu. VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling. In *arXiv:2111.1268*, 2021.

[16] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[17] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017.

[18] A. Habibian, H. Ben Yahia, D. Abati, E. Gavves, and F. Porikli. Delta distillation for efficient video processing. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, pages 213–229, Cham, 2022. Springer Nature Switzerland.

[19] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for NLP. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR, 09–15 Jun 2019.

[20] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, page 2790–2799. PMLR, 2019.

[21] M. Jia, L. Tang, B.-C. Chen, C. Cardie1, S. Belongie, B. Hariharan, and S.-N. Lim. Visual prompt tuning. In *European Conference on Computer Vision (ECCV)*, 2022.

[22] S. Jie and Z.-H. Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022.

[23] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision (ECCV)*. Springer, 2022.

[24] R. Karimi Mahabadi, J. Henderson, and S. Ruder. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34:1022–1035, 2021.

[25] R. Karimi Mahabadi, S. Ruder, M. Dehghani, and J. Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. *arXiv e-prints*, pages arXiv–2106, 2021.

[26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. 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.

[27] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[29] K. Li, Y. Wang, Y. He, Y. Li, Y. Wang, L. Wang, and Y. Qiao. Uniformerv2: Unlocking the potential of image vits for video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1632–1643, 2023.

[30] K. Li, Y. Wang, J. Zhang, P. Gao, G. Song, Y. Liu, H. Li, and Y. Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[31] L. Li, Z. Gan, K. Lin, C.-C. Lin, Z. Liu, C. Liu, and L. Wang. Lavender: Unifying video-language understanding as masked language modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23119–23129, 2023.

[32] O. I. Li, O. Camps, and M. Sznaier. Cross-view activity recognition using hankelets. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pages 1362–1369. IEEE, 2012.

[33] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.

[34] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019.

[35] J. Lin, C. Gan, and S. Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[36] Z. Lin, S. Geng, R. Zhang, P. Gao, G. de Melo, X. Wang, J. Dai, Y. Qiao, and H. Li. Frozen clip models are efficient video learners. In *European Conference on Computer Vision*, pages 388–404. Springer, 2022.

[37] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo. Swin transformer v2: Scaling up capacity and resolution. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[38] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *CVPR*, 2022.

[39] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu. Video swin transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[40] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu. Tam: Temporal adaptive module for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13708–13718, 2021.

[41] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 143–152, 2020.

[42] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[43] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018.

[44] B. Ni, H. Peng, M. Chen, S. Zhang, G. Meng, J. Fu, S. Xiang, and H. Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[45] X. Nie, B. Ni, J. Chang, G. Meng, C. Huo, Z. Zhang, S. Xiang, Q. Tian, and C. Pan. Pro-tuning: Unified prompt tuning for vision tasks, 2022.

[46] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[47] J. Pan, S. Chen, M. Z. Shou, Y. Liu, J. Shao, and H. Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021.

[48] J. Pan, Z. Lin, X. Zhu, J. Shao, and H. Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022.

[49] J. Park, J. Lee, and K. Sohn. Dual-path adaptation from image to video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2203–2213, 2023.

[50] J. Pfeiffer, A. Kamath, A. Rücklé, K. Cho, and I. Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. In *EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference*, EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, pages 487–503. Association for Computational Linguistics (ACL), 2021.

[51] A. Piergiovanni and M. S. Ryoo. Representation flow for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[52] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[53] C. Ryali, Y.-T. Hu, D. Bolya, C. Wei, H. Fan, P.-Y. Huang, V. Aggarwal, A. Chowdhury, O. Poursaeed, J. Hoffman, J. Malik, Y. Li, and C. Feichtenhofer. Hiera: A hierarchical vision transformer without the bells-and-whistles. *ICML*, 2023.

[54] M. S. Ryoo, A. Piergiovanni, J. Kangaspunta, and A. Angelova. Assemblenet++: Assembling modality representations via attention connections. In *Eur. Conf. Comput. Vis.*, pages 654–671, 2020.

[55] D. Shao, Y. Zhao, B. Dai, and D. Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2616–2625, 2020.

[56] L. Shi, Y. Zhang, J. Cheng, and H. Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12026–12035, 2019.

[57] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[58] Y.-L. Sung, J. Cho, and M. Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-language tasks. In *CVPR*, 2022.

[59] Z. Tong, Y. Song, J. Wang, and L. Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Neural Information Processing Systems*, 2022.

[60] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[62] J. Wang, D. Chen, Z. Wu, C. Luo, L. Zhou, Y. Zhao, Y. Xie, C. Liu, Y.-G. Jiang, and L. Yuan. Omnivl: One foundation model for image-language and video-language tasks. *Advances in neural information processing systems*, 35:5696–5710, 2022.

[63] J. Wang, Y. Ge, R. Yan, Y. Ge, K. Q. Lin, S. Tsutsui, X. Lin, G. Cai, J. Wu, Y. Shan, et al. All in one: Exploring unified video-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6598–6608, 2023.

[64] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2740–2755, 2018.

[65] Y. Wang, K. Li, Y. Li, Y. He, B. Huang, Z. Zhao, H. Zhang, J. Xu, Y. Liu, Z. Wang, S. Xing, G. Chen, J. Pan, J. Yu, Y. Wang, L. Wang, and Y. Qiao. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

[66] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Proceedings of the European conference on computer vision (ECCV)*, pages 305–321, 2018.

[67] H. Xue, Y. Sun, B. Liu, J. Fu, R. Song, H. Li, and J. Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv preprint arXiv:2209.06430*, 2022.

[68] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3333–3343, 2022.

[69] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[70] D. Yang, Y. Wang, A. Dantcheva, L. Garattoni, G. Francesca, and F. Brémond. Unik: A unified framework for real-world skeleton-based action recognition. *arXiv preprint arXiv:2107.08580*, 2021.

[71] J. Yu, Z. Wang, V. Vasudevan, L. Yeung, M. Seyedhosseini, and Y. Wu. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*, 2022.

[72] L. Yuan, D. Chen, Y.-L. Chen, N. Codella, X. Dai, J. Gao, H. Hu, X. Huang, B. Li, C. Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021.

[73] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

[74] B. Zhang, J. Yu, C. Fifty, W. Han, A. M. Dai, R. Pang, and F. Sha. Co-training transformer with videos and images improves action recognition. *arXiv preprint arXiv:2112.07175*, 2021.

[75] B. Zhang, J. Yu, C. Fifty, W. Han, A. M. Dai, R. Pang, and F. Sha. Co-training transformer with videos and images improves action recognition. *arXiv preprint arXiv:2112.07175*, 2021.