

# AR-VPT: Simple Auto-Regressive Prompts for Adapting Frozen ViTs to Videos

Muhammad Zain Yousuf<sup>1</sup><sup>a</sup>, Syed Talal Wasim<sup>2</sup><sup>b</sup>, Syed Nouman Hasany<sup>3</sup><sup>c</sup>  
and Muhammad Farhan<sup>1</sup><sup>d</sup>

<sup>1</sup>Habib University, Pakistan

<sup>2</sup>MBZUAI, U.A.E.

<sup>3</sup>Université de Rouen, France

**Keywords:** Computer Vision, Frozen ViT, Temporal Modelling, Video Recognition, Prompt Tuning.

**Abstract:** The rapid progress of deep learning in image recognition has driven increasing interest in video recognition. While image recognition has benefited from the abundance of pre-trained models, video recognition remains challenging due to the absence of strong pre-trained models and the computational cost of training from scratch. Transfer learning techniques have been used to leverage pre-trained networks for video recognition by extracting features from individual frames and combining them for decision-making. In this paper, we explore the use of Visual-Prompt Tuning (VPT) for video recognition, a computationally efficient technique previously proposed for image recognition. Our contributions are two-fold: we introduce Auto-Regressive Visual Prompt Tuning (AR-VPT) method to perform temporal modeling, addressing the weakness of VPT in this aspect. Finally, we achieve significantly improved performance compared to vanilla VPT on three benchmark datasets: UCF-101, Diving-48, and Something-Something-v2. Our proposed method achieves an optimal trade-off between performance and computation cost, making it a promising approach for video recognition tasks.

## 1 INTRODUCTION


Following deep learning's success in image recognition, interest in video recognition has been steadily growing in recent years. Whereas image recognition on modestly sized datasets is now considered close to being solved, one cannot say the same with regard to video recognition. This can partly be attributed to the plethora of pre-trained models available for image recognition. While it would be difficult to train such models from scratch on a custom dataset, the availability of transfer learning techniques imply that they can achieve a decent performance thanks to fine-tuning. Datasets such as ImageNet-1k and ImageNet-22k have played a key role in popularizing these techniques.


Video recognition, on the other hand, is far from


being considered solved. Training a model from scratch is a challenge both in terms of its computational cost as well as the amount of training data required by the model. Given the absence of strong pre-trained models for video recognition, many techniques try to leverage the already available pre-trained networks for classification and utilize them to extract features from individual video frames. These features are then combined in order to arrive at a final decision.


While these pre-trained models are often fine-tuned in order to update them given the video recognition task, it is often desirable to avoid the fine-tuning step owing to its computationally expensive nature. Examples of this include utilizing a frozen backbone (CNN-based or transformer based) in order to extract features from individual video frames and then learning a sequential model over such features. However, since our feature-extracting backbone is never updated, it is not always certain that the features obtained from it will be relevant to the task.

Given the aforementioned considerations, prompting techniques seem to be an ideal candidate

<sup>a</sup> <https://orcid.org/0009-0003-1313-9554>

<sup>b</sup> <https://orcid.org/0000-0003-0343-419X>

<sup>c</sup> <https://orcid.org/0000-0002-5915-4528>

<sup>d</sup> <https://orcid.org/0000-0002-8244-8313>

for video recognition. Visual-Prompt Tuning (VPT) (Jia et al., 2022a) was proposed as a computationally efficient technique that allowed the fine-tuning of vision transformers for image recognition without having to update the entire model. Prompts - which are simply extra input tokens - are introduced in the pipeline, and instead of modifying the network architecture, only the prompts are modified during transfer learning in order to obtain the task-relevant prompts. These prompts can either be added to the input tokens only (shallow approach), or they can be added at every transformer layer (deep approach). As VPT allows for fine-tuning in image recognition without having to modify the existing architecture, it's natural to consider it in a video recognition pipeline. This is because it will allow the extraction of useful features from individual frames while also learning computationally inexpensive task-specific prompts.

In this work, we consider the application of VPT to video recognition. In addition to applying the vanilla VPT, we propose a simple modification (AR-VPT) by introducing 'recurrent prompts' which allow the VPT to be more tailored towards video recognition problems. Instead of having a set of prompts for each input frame, prompts are only added to the first frame and the updated version of these prompts is then fed into the next frame. This allows the prompts to take the sequential nature of the problem into account.

Our major contributions are summarized as follows:

- Considering the weakness in VPT which is its incapability to perform temporal modeling, we propose our very own temporal attention methods based on prompting called Auto-Regressive VPT (AR-VPT) by extending the VPT model to perform temporal modeling.
- We achieve significantly improved performance compared to the vanilla VPT on three major benchmarks: UCF-101 (Soomro et al., 2012), Diving-48 (Li et al., 2018) and Something-Something-v2 (Goyal et al., 2017). We also achieve an optimal trade-off between performance and computational cost.

## 2 RELATED WORK

### 2.1 Video Recognition

Video recognition has seen significant progress in spatiotemporal learning techniques, evolving from

hand-crafted features to end-to-end deep learning methods. Initially, video recognition relied on feature-based approaches (Dollár et al., 2005; Klaser et al., 2008; Wang et al., 2013). However, the success of 2D CNNs in image recognition led to their adoption in video recognition tasks (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016; Tan and Le, 2019).

Subsequently, with the introduction of large-scale datasets like Kinetics (Kay et al., 2017), 3D CNN-based methods emerged as more effective in capturing spatio-temporal relations and outperformed their 2D counterparts (Carreira and Zisserman, 2017; Feichtenhofer et al., 2016; Tran et al., 2015). Despite their improved performance, 3D CNNs came with high computational costs, prompting the development of various optimized variants (Feichtenhofer, 2020; Sun et al., 2015; Szegedy et al., 2016; Tran et al., 2018; Xie et al., 2018; Li et al., 2020; Lin et al., 2019; Qiu et al., 2019; Feichtenhofer et al., 2019; Duan et al., 2020).

In parallel, self-attention based architectures gained prominence in image recognition with the introduction of the Vision Transformer (ViT) model (Dosovitskiy et al., 2021). These models were subsequently adapted for video recognition, initially combining Vision Transformers with CNNs to model long-range context (Wang et al., 2018; Wang et al., 2020; Kondratyuk et al., 2021). Later advancements introduced fully transformer-based architectures (Liu et al., 2022; Arnab et al., 2021; Bertasius et al., 2021; Yan et al., 2022; Zhang et al., 2021; Patrick et al., 2021; Fan et al., 2021; Li et al., 2022b), surpassing previous methods across multiple benchmarks.

Recently, hybrid methods were proposed, combining elements from both CNNs and Vision Transformers, achieving competitive performance compared to state-of-the-art fully transformer-based methods (Li et al., 2022a; Wasim et al., 2023a).

### 2.2 Visual Prompting

Prompting, a concept introduced in the field of Natural Language Processing (NLP) (Liu et al., 2021a; Jiang et al., 2020), involves generating task specific instructions to elicit desired behavior from language models. These instructions can either be manually crafted (Brown et al., 2020) or acquired through training with discrete (Gao et al., 2020; Jiang et al., 2020; Rohrbach et al., 2017; Schick and Schütze, 2020) or continuous vectors (Lester et al., 2021; Li and Liang, 2021). Recently, prompt learning has extended to vision-related problems to transfer knowledge from large-scale models to downstream tasks. Presently,

the techniques of prompting are applied in both uni-models, such as Vision Transformers (ViTs) trained on images (Dosovitskiy et al., 2021), and multimodal models like CLIP (Radford et al., 2021).

In the context of ViTs, researchers like (Jia et al., 2022b; Bahng et al., 2022) have employed learnable prompts to guide pretrained vision transformers (Dosovitskiy et al., 2021; Liu et al., 2021b). In a different approach, methods such as (Zhou et al., 2022b; Zhou et al., 2022a; Sun et al., 2022) have introduced learnable vectors into the text encoder of CLIP to enable transfer learning for image recognition tasks. Additional methods such as (Khattak et al., 2023) and (Wasim et al., 2023b) employed multi-modal prompts for image and video recognition tasks respectively.

### 3 METHODOLOGY

Our work, AR-VPT, adapts the pretrained image-based vision model for videos using a prompting scheme in the visual space while keeping the transformer backbone frozen. We want to introduce recurrence in the original VPT architecture to perform temporal learning only while keeping the spatial attention frozen. AR-VPT allows the utilization of the existing pretrained transformer model rather than training one from scratch for videos.

This section presents our approach. We begin with the architecture of the transformer model followed by the adaptation of ViT for visual prompting in 2D space. Basically, in this section, we explain the architecture of VPT and then gradually build upon it to provide a detailed explanation of our Auto-Regressive Visual Prompt Tuning (AR-VPT) scheme.

#### 3.1 Overview of Visual Prompt Tuning (VPT)

Consider a video  $V \in \mathbb{R}^{T \times H \times W \times 3}$  of spatial size  $H \times W$  with  $T$  frames. Each frame  $t \in \{1 \dots T\}$  is divided into  $N$  non-overlapping patches of size  $P \times P$ . Hence, the total number of patches would be  $N = H \times W / P^2$ . Each patch of shape  $P \times P \times 3$  in each frame  $t$  is flattened which is represented as  $\{x_{t,i} \in \mathbb{R}^{3P^2}\}_{i=1}^N$ , where  $t$  is the frame and  $i$  is the patch number. The vectors are then projected to form token embeddings using a linear projection layer  $P_{emb} \in \mathbb{R}^{3P^2 \times D}$  where  $D$  is the output dimension for each token. A CLS token  $x_{cls} \in \mathbb{R}^D$  is always prepended to the embedded token sequence for each frame. Hence, the final frame-level

token sequence is given as:

$$z_t^{(0)} = [x_{cls}, P_{emb}x_{t,1}, \dots, P_{emb}x_{t,N}] + e, \quad (1)$$

where  $e$  represents positional encodings.

To adapt a pretrained frozen Vision Transformer (ViT) (Dosovitskiy et al., 2021) to downstream tasks, VPT employs trainable prompt tokens that are prepended to the above token sequence  $z_t^{(0)}$ . More specifically, in the VPT Deep architecture, we prepend a few learnable prompts  $\{M_{t,i}^{(l)}\}_{i=1}^{num}$  in each layer  $l$  of the image encoder, where  $num$  is the number of prompts added for frame  $t$ . Hence, the frame-level token sequence with learnable prompts is now given as follows:

$$z_t^{(0)}, \{M_{t,i}^{(0)}\}_{i=1}^{num} = [x_{cls} + e, P_{emb}x_{t,1} + e, \dots, P_{emb}x_{t,N} + e, M_{t,0}^{(0)}, \dots, M_{t,num}^{(0)}], \quad (2)$$

Note that for the VPT architecture, the prompts for each frame are the same as the previous and subsequent frames. Hence,  $\{M_{0,i}^{(l)} = M_{1,i}^{(l)} = \dots = M_{T,i}^{(l)}\}_{i=1}^{num}$

From the  $L_v$  layered image encoder in the ViT model, the frame-level representation for each frame  $t$  in VPT deep architecture is obtained as follows:

$$z_t^{(l)} = f_{\theta_v}^{(l)}(z_t^{(l-1)}, \{M_{t,i}^{(l-1)}\}_{i=1}^{num}), \quad l \in 1, \dots, L_v, \quad (3)$$

where  $f_{\theta_v}^{(l)}$  is the  $l$ -th layer of the frozen ViT image encoder. The final video-level representation is then formed by averaging across all frames:

$$z^{(l)} = \text{AVGPOOL}(z_0^{(l)}, z_1^{(l)}, \dots, z_t^{(l)}), \quad (4)$$

where AVGPOOL is the average pooling operator.

#### 3.2 AR-VPT Architecture

We have seen the architecture of VPT deep for video recognition in the previous section. We will now explain the AR-VPT scheme which basically introduces a form of auto-regression in prompts and re-uses the prompts updated after layer  $l$  in frame  $t - 1$  for the layer  $l$  in the frame  $t$  of the image encoder. Hence, the frame-level token sequence representation with learnable prompts for the first layer  $l = 0$  of the first frame  $t = 0$  is the same as VPT Deep as follows:

$$z_0^{(0)}, \{M_{0,i}^{(0)}\}_{i=1}^{num} = [x_{cls} + e, P_{emb}x_{0,1} + e, \dots, P_{emb}x_{0,N} + e, M_{0,0}^{(0)}, \dots, M_{0,num}^{(0)}], \quad (5)$$

For layer  $l = 0$  of frame  $t = 1$ , it would be:

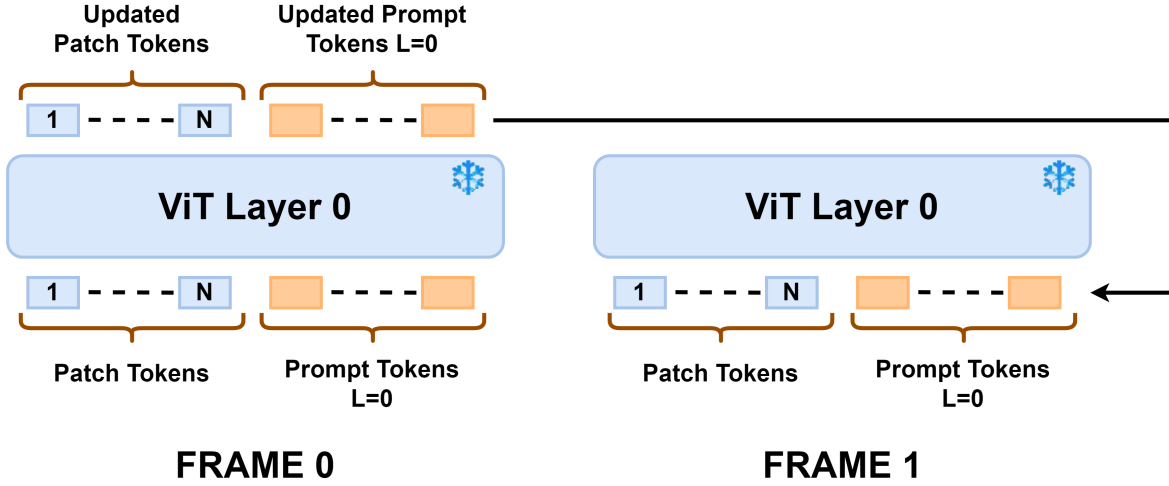


Figure 1: **The AR-VPT architecture:** We show for a single layer of the Frozen ViT - denoted by  $\text{ViT}$  - of the AR-VPT architecture. Prompts for the layer  $L = 0$ , are updated at the output of layer  $L = 0$  for frame 0 which are then used as input for layer  $L = 0$  for frame 1.

$$z_1^{(0)}, \{M_{1,i}^{(0)}\}_{i=1}^{num} = [x_{cls} + e, P_{emb}x_{1,1} + e, \dots, P_{emb}x_{1,N} + e, M_{1,0}^{(0)}, \dots, M_{1,num}^{(0)}], \quad (6)$$

where  $\{M_{1,i}^{(0)}\}_{i=1}^{num}$  are the output prompt tokens produced after the transformer layer  $l = 0$  for frame  $t = 0$ .

Therefore, for the  $L_v$  layered video encoder, the frame-level representation for each frame  $t$  in AR-VPT is given as follows:

$$z_t^{(l)}, \{M_{t,i}^{(l)}\}_{i=1}^{num} = f_{\theta_v}^{(l)}(z_t^{(l-1)}, \{M_{t,i}^{(l-1)}\}_{i=1}^{num}), \quad l \in 1, \dots, L_v, \quad (7)$$

where  $f_{\theta_v}^{(l)}$  is the  $l$ -th layer of the image encoder. This is also represented in Figure 1. Similar to VPT, the final video level representation is then formed by averaging across all frames:

$$z^{(l)} = \text{AVGPOOL}(z_0^{(l)}, z_1^{(l)}, \dots, z_t^{(l)}), \quad (8)$$

where AVGPOOL is the average pooling operator.

## 4 RESULTS AND ANALYSIS

### 4.1 Experimental Setup and Protocols

#### 4.1.1 Datasets

We present results for video recognition on three datasets: UCF-101 (Soomro et al., 2012), Diving-48 (Li et al., 2018) and Something-Something-v2

(SSv2) (Goyal et al., 2017). UCF-101, a coarse-grained video dataset, consists of a total of 13,320 videos with 9,573 training videos and 3,783 testing samples across 101 classes. Diving-48 consists of 18k total temporally fine-grained videos with 16k training and 2k testing videos across 48 classes. SSv2 is another fine-grained dataset of 220,847 labeled video clips with 169k training and 24.7k validation videos across 174 classes. For all three datasets, we report the Top-1 accuracy and compare it against the relevant baselines.

#### 4.1.2 Training

We train our models for 15 epochs using SGD with an initial learning rate of 0.005, which is divided by 10 at epochs 11, and 14. During training, we first resize the shorter side of the video to a random value in [256, 320]. We then randomly sample a  $224 \times 224$  crop from the resized video. We randomly sample clips from the full-length videos with a frame rate of 1/32. The batch size is set to 64. The momentum is set to 0.9, while the weight decay is set to 0.0001. We use a patch size  $P = 16$ , and the number of prompts ( $num$ ) added in AR-VPT are 5.

In all our experiments, we use the “Base” ViT model with layers  $L = 12$  (Dosovitskiy et al., 2021). Attention layers in each block are initialized with the same weights.

#### 4.1.3 Baselines

We define two baselines for this paper. The first is the *Frozen-ViT* baseline. In this method, all the layers of the standard ViT model, pretrained on ImageNet, are

Table 1: Comparison with baseline methods on UCF-101 (Soomro et al., 2012) dataset.

Method	Top-1
Frozen-ViT (ICCV'21)	84.69
VPT-Deep (ECCV'22)	87.30
AR-VPT	88.10

Table 2: Comparison with baseline methods on Diving-48 (Li et al., 2018) dataset.

Method	Top-1
Frozen-ViT (ICLR'21)	63.8
VPT-Deep (ECCV'22)	67.5
AR-VPT	69.7

Table 3: Comparison with baseline methods on SSv2 (Goyal et al., 2017) dataset.

Method	Top-1
Frozen-ViT (ICCV'21)	15.5
VPT-Deep (ECCV'22)	16.1
AR-VPT	37.8

Table 4: Comparison with baseline methods on computational efficiency on videos with a spatial resolution of  $256 \times 256$ , temporal resolution of 8, 101 classes and batch size of 1.

Method	Trainable Parameters (K)	FLOPs (G)	Throughput (FPS)	GPU Memory Used (GB/GPU)
Frozen-ViT (ICLR'21)	77.6	140.65	32.87	11.61
VPT-Deep (ECCV'22)	123.6	144.34	31.87	12.3
AR-VPT	123.6	164.84	22.21	12.3

frozen, and only a new head is trained. The second is the VPT-Deep baseline where we add additional trainable prompt tokens to the otherwise frozen ViT model.

## 4.2 Comparison with Baselines

### 4.2.1 UCF-101

On the UCF-101 dataset, we report results for AR-VPT comparing against Frozen-ViT model and vanilla VPT-Deep in Table 1. It can be observed that our model AR-VPT surpasses the Frozen-ViT model and VPT-Deep model by 3.41% and 0.8% respectively.

### 4.2.2 Diving-48

On the Diving-48 dataset, we report results for AR-VPT comparing against the Frozen-ViT and vanilla VPT-Deep in Table 2. Compared to Frozen-ViT and VPT-Deep models, AR-VPT again surpasses both these models by 5.9% and 2.2% respectively.

### 4.2.3 Something-Something-v2

On the SSv2 dataset, we again report results for AR-VPT comparing against Frozen-ViT and VPT-Deep model in Table 3. On this temporally challenging dataset, our model surpasses both Frozen-ViT and VPT-Deep models by a significant difference of 22.3% and 21.7% respectively. This shows that our model is able to learn sophisticated long-range temporal dependencies effectively.

The reason why AR-VPT is performing significantly better than the baselines on SSv2 is because there is no temporal modelling in the baselines which leads to their comparatively poorer performance on temporally fine-grained datasets such as SSv2. However, we perform explicit temporal modelling in AR-VPT via the recurrence of the prompts which enables our architecture to perform better on fine-grained video datasets.

### 4.2.4 Computational Efficiency Comparison

We evaluate each of the method on various computational efficiency metrics including No. of learnable parameters, GFLOPs, Throughput, and Average GPU Mem. Used, and report results in Table 4 to provide a fair and transparent comparison between the models. Each of the efficiency metric was calculated with the batch size of 1, except for GPU Mem. Used (it was calculated with batch size of 64 while training). Throughput was calculated with batch size of 1 and averaged over 30 times on NVIDIA® TITAN RTX™ (24 GB/GPU). It can be observed that the number of learnable parameters are identical for VPT-Deep and AR-VPT, which is a unique feature of our model as we extended the Vanilla VPT to perform temporal modelling, yet there is no increase in the number of learnable parameters. However, increase in GFLOPs and less Throughput can be observed in AR-VPT which can be attributed to the regressive nature of the model. In terms of GPU memory used, there is no significant difference in the performance of the models which again shows the architectural superiority of AR-VPT when we take its performance into account.

## 5 CONCLUSIONS

This paper presents a method to perform temporal modeling effectively and efficiently for video recognition tasks. This architecture, AR-VPT, is an extension of the original VPT architecture and adapts the prompt-tuning technique in visual space to perform temporal feature learning. We demonstrate that our model is able to effectively learn long-range dependencies in the spatiotemporal dimension via the evaluation on both coarse and fine-grained video datasets. This method shows how effective a simple prompting mechanism can be when incorporating information sharing across frames auto-regressively.

## REFERENCES

- Arnab, A., Deghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). Vivit: A video vision transformer. In *ICCV*.
- Bahng, H., Jahanian, A., Sankaranarayanan, S., and Isola, P. (2022). Visual prompting: Modifying pixel space to adapt pre-trained models. *arXiv:2203.17274*.
- Bertasius, G., Wang, H., and Torresani, L. (2021). Is space-time attention all you need for video understanding? In *ICML*.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*.
- Dollár, P., Rabaud, V., Cottrell, G., and S. (2005). Behavior recognition via sparse spatio-temporal features. In *2005 IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance*, pages 65–72. IEEE.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Deghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houshy, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*.
- Duan, H., Zhao, Y., Xiong, Y., Liu, W., and Lin, D. (2020). Omni-sourced webly-supervised learning for video recognition. In *ECCV*.
- Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., and Feichtenhofer, C. (2021). Multiscale vision transformers. In *ICCV*.
- Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. In *CVPR*.
- Feichtenhofer, C., Fan, H., Malik, J., and He, K. (2019). Slowfast networks for video recognition. In *ICCV*.
- Feichtenhofer, C., Pinz, A., and Wildes, R. (2016). Spatiotemporal residual networks for video action recognition. In *NeurIPS*.
- Gao, T., Fisch, A., and Chen, D. (2020). Making pre-trained language models better few-shot learners. *arXiv:2012.15723*.
- Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Freund, I., Yianilos, P., Mueller-Freitag, M., et al. (2017). The “something something” video database for learning and evaluating visual common sense. In *ICCV*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.
- Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. (2022a). Visual prompt tuning. In *ECCV*.
- Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N. (2022b). Visual prompt tuning. *arXiv:2203.12119*.
- Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. (2020). How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.
- Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al. (2017). The kinetics human action video dataset. In *arXiv:1705.06950*.
- Khattak, M. U., Rasheed, H., Maaz, M., Khan, S., and Khan, F. S. (2023). Maple: Multi-modal prompt learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Klaser, A., Marszałek, M., and Schmid, C. (2008). A spatiotemporal descriptor based on 3d-gradients. In *BMVC*.
- Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., and Gong, B. (2021). Movinets: Mobile video networks for efficient video recognition. In *CVPR*.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NeurIPS*.
- Lester, B., Al-Rfou, R., and Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *arXiv:2104.08691*.
- Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., and Qiao, Y. (2022a). Uniformer: Unified transformer for efficient spatiotemporal representation learning. In *ICLR*.
- Li, X. L. and Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv:2101.00190*.
- Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., and Wang, L. (2020). Tea: Temporal excitation and aggregation for action recognition. In *CVPR*.
- Li, Y., Li, Y., and Vasconcelos, N. (2018). Resound: Towards action recognition without representation bias. In *ECCV*.
- Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., and Feichtenhofer, C. (2022b). Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*.
- Lin, J., Gan, C., and Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. In *ICCV*.

- Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., and Neubig, G. (2021a). Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv:2107.13586*.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. (2021b). Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*.
- Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., and Hu, H. (2022). Video swin transformer. In *CVPR*.
- Patrick, M., Campbell, D., Asano, Y. M., Metze, I. M. F., Feichtenhofer, C., Vedaldi, A., Henriques, J., et al. (2021). Keeping your eye on the ball: Trajectory attention in video transformers. In *NeurIPS*.
- Qiu, Z., Yao, T., Ngo, C.-W., Tian, X., and Mei, T. (2019). Learning spatio-temporal representation with local and global diffusion. In *CVPR*.
- Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. In *ICML*.
- Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., and Schiele, B. (2017). Movie description. *IJCV*, 123(1):94–120.
- Schick, T. and Schütze, H. (2020). Exploiting cloze questions for few shot text classification and natural language inference. *arXiv:2001.07676*.
- Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *ICLR*.
- Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*.
- Sun, L., Jia, K., Yeung, D.-Y., and Shi, B. E. (2015). Human action recognition using factorized spatio-temporal convolutional networks. In *ICCV*.
- Sun, X., Hu, P., and Saenko, K. (2022). Dualcoop: Fast adaptation to multi-label recognition with limited annotations. *arXiv:2206.09541*.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *CVPR*.
- Tan, M. and Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*.
- Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.
- Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *CVPR*.
- Wang, H., Kläser, A., Schmid, C., and Liu, C.-L. (2013). Dense trajectories and motion boundary descriptors for action recognition. In *IJCV*.
- Wang, X., Girshick, R., Gupta, A., and He, K. (2018). Non-local neural networks. In *CVPR*.
- Wang, X., Xiong, X., Neumann, M., Piergiovanni, A., Ryoo, M. S., Angelova, A., Kitani, K. M., and Hua, W. (2020). Attentionnas: Spatiotemporal attention cell search for video classification. In *ECCV*.
- Wasim, S. T., Khattak, M. U., Naseer, M., Khan, S., Shah, M., and Khan, F. S. (2023a). Video-focalnets: Spatio-temporal focal modulation for video action recognition. *arXiv:2307.06947*.
- Wasim, S. T., Naseer, M., Khan, S., Khan, F. S., and Shah, M. (2023b). Vita-clip: Video and text adaptive clip via multimodal prompting. In *CVPR*.
- Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*.
- Yan, S., Xiong, X., Arnab, A., Lu, Z., Zhang, M., Sun, C., and Schmid, C. (2022). Multiview transformers for video recognition. In *CVPR*.
- Zhang, Y., Li, X., Liu, C., Shuai, B., Zhu, Y., Brattoli, B., Chen, H., Marsic, I., and Tighe, J. (2021). Vidtr: Video transformer without convolutions. In *ICCV*.
- Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2022a). Conditional prompt learning for vision-language models. In *CVPR*.
- Zhou, K., Yang, J., Loy, C. C., and Liu, Z. (2022b). Learning to prompt for vision-language models. *IJCV*.