2024 18th International Conference on Automatic Face and Gesture Recognition (FG)

Attention Prompt Tuning: Parameter-efficient Adaptation of Pre-trained Models for Action Recognition

Wele Gedara Chaminda Bandara and Vishal M. Patel
Johns Hopkins University, Baltimore, Maryland, USA
https://github.com/wgcban/apt

Abstract— **In this paper, we introduce Attention Prompt Tuning (APT) – a computationally efficient variant of prompt tuning for video-based applications such as action recognition. Prompt tuning approaches involve injecting a set of learnable prompts along with data tokens during fine-tuning while keeping the backbone frozen. This approach greatly reduces the number of learnable parameters compared to full tuning. For image-based downstream tasks, normally a couple of learnable prompts achieve results close to those of full tuning. However, videos, which contain more complex spatiotemporal information, require hundreds of tunable prompts to achieve reasonably good results. This reduces the parameter efficiency observed in images and significantly increases latency and the number of floating-point operations (FLOPs) during inference. To tackle these issues, we directly inject the prompts into the keys and values of the non-local attention mechanism within the transformer block. Additionally, we introduce a novel prompt reparameterization technique to make APT more robust against hyperparameter selection. The proposed APT approach greatly reduces the number of FLOPs and latency while achieving a significant performance boost over the existing parameter-efficient tuning methods on UCF101, HMDB51, and SSv2 datasets for action recognition. The code and pre-trained models are available at https://github.com/wgcban/apt**

I. Introduction

Video-based action recognition is the task of identifying human activities, including gestures, simple actions, interactions involving humans and objects, from videos [1], [2], [3], [4]. Encoding temporal information is crucial for recognizing different activities [5]. The early action recognition models primarily relied on deep neural networks (DNNs) [5], owing much to the success of convolutional neural networks (CNNs) in encoding the spatiotempoal information. However, with the introduction of Transformer [6] networks in natural language processing (NLP) to model sequences of data with non-local self-attention, their ability to capture spatiotemporal information from videos for downstream applications like action recognition has been explored [5], [1], yielding better results compared to their CNN counterparts. The adoption of Transformer networks has been further accelerated by the advancements in efficient self-supervised representation learning approaches for pre-training, such as masked autoencoders (MAEs) [7], [8], [9].

Transformers [6], [10], [11] have revolutionized a wide range of fields, including natural language, audio, image, and video processing [12]. They have been applied to various downstream tasks, such as language translation [6], audio classification [13], image classification [10], and action recognition [9], [8]. The *de facto* standard when utilizing transformer backbones for downstream tasks is full fine-tuning, i.e. tuning the backbone and a task-specific head by initializing the backbone with pre-trained weights [9], [8], [14]. Although full fine-tuning has demonstrated strong performance in most cases, it presents certain disadvantages. These include:

1) *parameter inefficiency*—full fine-tuning can lead to a significant increase in the number of parameters, especially when multiple tasks are involved, as it does not share parameters between tasks;
2) *large labeled data requirements*—full fine-tuning necessitates a substantial volume of labeled data to avoid overfitting and ensure good generalization;
3) *computational demands*—full fine-tuning is computationally intensive and time-consuming, requiring significant computational resources;
4) *overfitting*—there is a risk of overfitting to the new data, particularly if the training data is limited;
5) *catastrophic forgetting*—fine-tuning on a new task may cause the model to forget previously learned knowledge from the pre-training task.

As a result, alternative methods, including linear probing, adapter tuning [15], and visual prompt tuning (VPT) [16], have been proposed, each with its own set of trade-offs (see Fig. 1). *Linear probing* involves tuning only a task-specific head while keeping the backbone parameters fixed (frozen). It is highly parameter-efficient and computationally less demanding. However, its performance is usually limited by the fixed nature of the backbone. On the other hand, *adapter tuning* [15] introduces learnable Multi-Layer Perceptron (MLP) layers in each transformer block, providing a balance between parameter efficiency and model flexibility. It allows for task-specific adaptations without changing the entire backbone and has shown better performance than linear probing. The added adapter layers increase the number of trainable parameters (but far less than full fine-tuning) and FLOPS (hence latency) compared to linear probing. In contrast, *VPT* [16] adopts tunable prompts that are fed into the model along with the input tokens. These prompts act as task-specific guides, allowing the model to retain the pre-trained knowledge of the backbone while adapting to the new task. VPT is highly flexible, parameter efficient, and has been shown to achieve better performance than linear probing and

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-21-2-0211.

979-8-3503-9494-8/24/$31.00 ©2024 IEEE

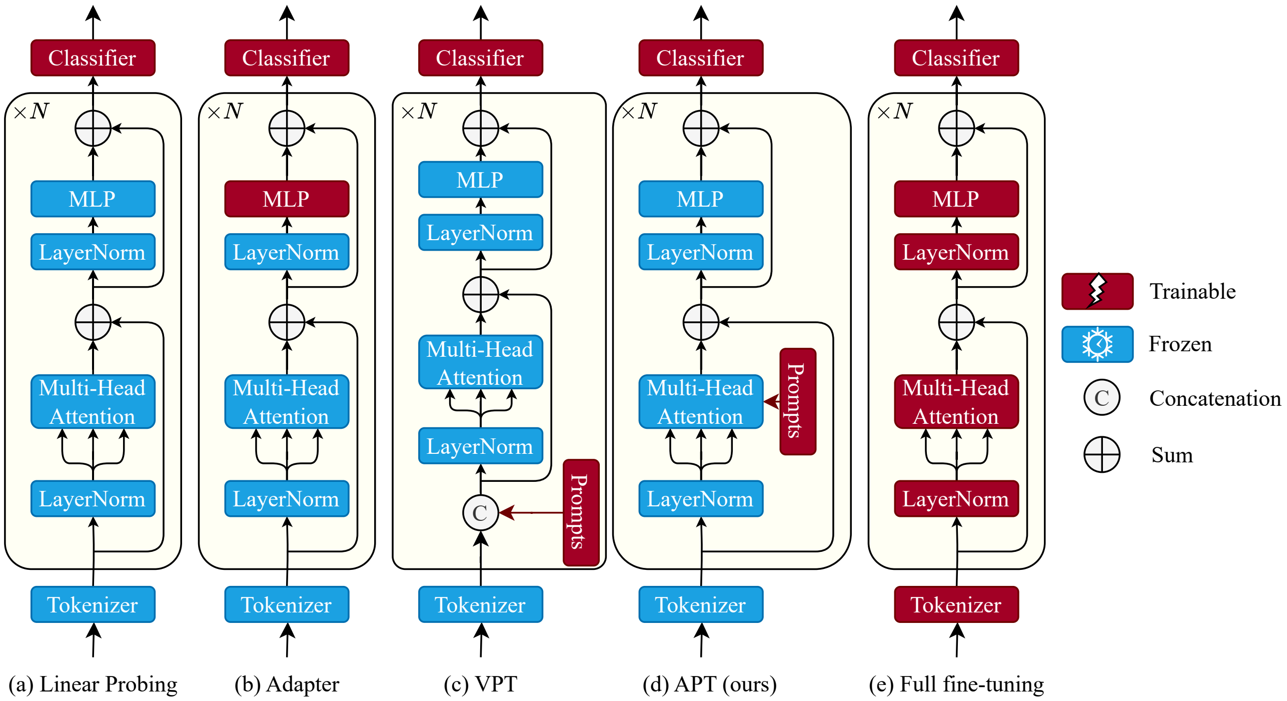


Fig. 1: Comparison of our Attention Prompt Tuning (APT) for videos action classification with other existing tuning methods: linear probing, adapter tuning [15], visual prompt tuning (VPT) [16], and full fine-tuning.

prompt tuning across a wide variety of downstream tasks specially those involving images [16]. However, we have observed that direct adaptation of VPT for downstream tasks involving videos, such as action recognition, leads to high computation requirements, unstable training, and increased FLOPs during training and inference.

Videos inherently consist of a significantly larger number of tokens compared to images (typically 1568 *vs.* 196 tokens) [9], [8], [7]. Consequently, VPT for videos often necessitates appending a substantial number of prompts (approximately 800 prompts) to the input tokens to achieve performance levels comparable to full fine-tuning. However, this approach presents challenges. The large number of appended prompts increases computational complexity (FLOPs) and latency during both training and inference. Elevated latency is particularly concerning for video-based downstream applications, which already demand significant time and computational resources. As such, the increased latency and FLOPs compromise the flexibility and parameter efficiency of visual prompt tuning for videos, highlighting the importance of reducing computational complexity while maintaining parameter efficiency.

In our experiments we have observed that interactions between video token representations and prompts primarily occur during the multi-head attention (MHA) in the transformer block. This is because both the MLPs and Layer Normalization (LayerNorm) [17] operations before and after the attention mechanism are applied along the embedding dimension separately for each token/prompt. As a result, the layers preceding the MHA serve as fixed reparameterization for the prompts, rendering the computations on prompts following the attention mechanism redundant. This redundancy arises because the current prompts are discarded at the end of each transformer layer, and new prompts are subsequently appended for the following layer. Motivated by this observation, we propose introducing prompts directly into the attention mechanism, thereby reducing redundancy and computational complexity. Our approach, named Attention Prompt Tuning (APT), minimizes extraneous computations, reduces latency, and enhances the effectiveness of visual prompt tuning for video-based tasks.

Moreover, we noted that VPT is highly sensitive to hyperparameter selection, such as learning rate and weight decay [18], and often requires a longer fine-tuning time (approximately twice as long) compared to full fine-tuning. This sensitivity is observed in other fields as well, including natural language processing and image processing [19], [20]. Since video fine-tuning generally takes longer than image fine-tuning, accelerating convergence and enhancing robustness to hyperparameter selection are of great interest. Previous works on prompt tuning [19] suggest prompt reparameterization as a solution to these challenges, wherein prompts are processed through a reparameterization network consisting of a small MLP layer. However, adding an MLP layer significantly increases the number of parameters and introduces additional computational cost. To address this, we introduce scaling-based reparameterization as a solution that adds a significantly lower number of parameters while incurring negligible computational cost. In this paper, we explore the benefits and effectiveness of this approach for video-based action recognition.

In summary, our contributions are as follows:

1) **We introduce Attention Prompt Tuning (APT)** – a parameter-efficient fine-tuning technique specifically designed for video-based action recognition.
2) **APT innovatively injects learnable prompts directly into the non-local attention mechanisms of Transformers.** This significantly enhances parameter efficiency and reduces redundant computations compared to existing methods such as Visual Prompt Tuning (VPT).
3) **We also introduce a novel prompt reparameterization technique**, making the training process less sensitive to hyperparameter selection.
4) **The effectiveness of our proposed APT is validated on three public action recognition benchmarks**: UCF101, HMDB51, and Something-Something-v2. We demonstrate that our method achieves superior action recognition performance while utilizing fewer tunable parameters.

II. Related Work

A. Action Recognition

Human action recognition is an important task in video understanding. It aims to recognize human actions in video clips. Researchers have proposed various approaches to recognizing actions in videos by learning spatio-temporal representations. In the early phase, this was accomplished by training specific models on action recognition datasets such as HMDB51 [4], UCF101 [3], and Something-Something-v2 (SSv2) [2]. Some of these early works on action recognition include DeepVideo [21], which attempts to utilize convolutional neural networks, and Two-Stream Networks, which introduced a second pathway to learn temporal information in videos by training a convolutional neural network on the optical flow stream. Two-Stream Networks marked a significant success in action recognition, leading to many follow-up works such as TDD [22], LRCN [23], Fusion [24], TSN [25], etc. However, the temporal dimension typically makes action recognition challenging. Additionally, existing deep architectures generally encode temporal information with limited solutions such as 3D filters (examples include I3D [26], R3D [27], S3D [28], non-local [29], and SlowFast [30]), pre-computed motion features, and recurrent neural networks (RNNs) [31]. These models are typically limited in simultaneously capturing local and global variations of temporal features. In contrast, the Transformer is a novel architecture that employs the attention mechanism to differentially weigh each input patches. Although Transformers are designed to handle sequential input data, they do not necessarily process it in order. Instead, the attention mechanism provides context for any position in the input sequence.

B. Transformers

The Transformer architecture [6], [10], [32], [33], [34], originally introduced for natural language processing (NLP) tasks, has been successfully applied to computer vision with the development of the Vision Transformer (ViT) [10]. ViT converts images into small patches (tokens) and processes them through multi-head attention modules, enabling tasks such as classification [10], [8], [35], segmentation [36], [37], [38], detection [38], synthesis [39], [40], [41], and more. With the development of self-supervised pre-training methods such as masked autoencoders for transformers (MAEs) [42], [9], [8], [43], [44], the use cases of transformers have expanded by leveraging unlabeled data for pre-training, which provides excellent transfer learning capability for various downstream applications with limited supervised labels [42], [9], [8]. Despite the success of pre-trained transformer models, fine-tuning the entire backbone for downstream tasks is commonly practiced but often inefficient in terms of the number of tunable parameters and computational resources. Consequently, there has been significant interest in exploring parameter-efficient techniques for adopting pre-trained models in downstream applications [16], [15], [19], [20].

C. Efficient Fine-tuning

Efficient tuning methods can be divided into two categories: (1) network-based methods and (2) prompting-based methods. Network-based methods, such as AdaptFormer [15], [45], ST-Adapter [46], and AIM [47], introduce a small, learnable network before or after each multi-head attention module, which are fine-tuned while keeping the rest of the parameters fixed. This approach significantly improves parameter efficiency compared to end-to-end tuning. However, the addition of these networks slightly alters the backbone architecture, which is a drawback. In contrast, prompting-based methods [16], [48], [49], [50], [51], [52], [53], originally inspired by prompt tuning in NLP [19], [20], [54], involve inserting a set of learnable prompts along with the input tokens during tuning.

Prompt tuning is relatively simple and easily applicable to downstream tasks. However, it has been observed that prompt tuning achieves significantly lower performance than full-tuning in complex video-based applications like action classification. Moreover, prompt tuning requires a higher number of tunable prompts for video applications, resulting in increased latency and computational requirements. In this paper, we eliminate redundant computations of prompt tuning by directly injecting prompts into the attention mechanism.

III. Method

A. Visual Prompt Tuning (VPT)

As depicted in Fig. 1, VPT [16] appends a set of learnable prompts to the input tokens. The combined input (tokens + prompts) is then processed by the Transformer Block, which consists of a sequence of components: LayerNorm, Multilayer Perceptrons (MLPs), and Multi-Head Attention (MHA). Notably, LayerNorm and MLPs operate along the embedding dimension, meaning that the interaction between the prompts and the input token features occurs primarily within the MHA component. Consequently, the LayerNorm and MLP components preceding the attention mechanism can be viewed as a fixed reparameterization network for the prompts. However, the LayerNorm and MLP components following the MHA are redundant because the current prompts are discarded at the end of the current Transformer Block, and a new set of prompts is appended to the next block. Motivated by this observation, we introduce an optimization to reduce the extra computational redundancy introduced by VPT. Specifically, we introduce the prompts directly into the attention mechanism, thereby saving significant computation – a benefit that is particularly valuable for video-based downstream tasks.

B. Attention Prompt Tuning (APT)

Attention Prompt Tuning (APT) injects learnable prompts directly into the MHA unlike VPT, as shown in Fig. 2. Specifically, we concatenate learnable prompts only with Keys (K) and Values (V), ensuring that the output from the MHA has the same number of tokens as the input. This avoids the redundant computations by MLP and LayerNorm layers on prompt tokens following the MHA, unlike in VPT.

Formally, the non-local attention operation in MHA is defined as,

$$x = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{1}$$

where Q, K, and V denote the Queries, Keys, and Values, respectively, and d is the size of the embedding dimension. In APT, we modify the Keys and Values of the input by appending a set of learnable prompts, denoted by K_p and V_p, as follows,

$$K = K_x \oplus K_p, \text{ and } \; V = V_x \oplus V_p, \tag{2}$$

where $\oplus$ represents concatenation. Here, K_x and V_x are the original Keys and Values derived from the input tokens. The concatenation results in Keys and Values with dimensions of size $(n_x + n_p, d)$, where n_x is the number of input tokens and n_p is the number of prompts. The dimension of Queries (Q) remains unchanged at (n_x, d). Consequently, the output after the MHA operation retains the same size as the input, which is (n_x, d).

The efficient design of APT improves computational efficiency by reducing the overhead introduced by VPT, making it particularly advantageous when dealing with large numbers of prompts, as in the case in video-based applications.

C. Prompt Reparameterization

During our experiments, we observed that the performance of prompt tuning methods is highly sensitive to the proper selection of hyperparameter values, especially learning rate and weight decay. This sensitivity has also been observed in prompt tuning applied to other applications in NLP and image classification. Furthermore, we noticed that prompt tuning typically requires a slightly higher number of fine-tuning epochs compared to full-model fine-tuning. A common practice to make prompt tuning more robust to variations in learning rate and other hyperparameters is to use a reparameterization network, which often involves processing the prompts through an additional MLP network. However, adding an MLP layer to a large number of prompts increases computational complexity, latency, and the number of tunable parameters. Therefore, we introduce a scaled reparameterization method in which each prompt in K_p and V_p is scaled by a learnable scalar. More precisely, the reparameterization of the Key and Value prompts is as follows,

$$\tilde{K_p} = \max(s_k, 1.0) \times K_p, \tag{3}$$

$$\tilde{V_p} = \max(s_v, 1.0) \times V_p, \tag{4}$$

where s_k and s_v are the learnable scalars of size $\mathbb{R}^{n_p}$. Importantly, s_k and s_v are initialized to 1.0. The proposed scaled reparameterization enables the direct adjustment of prompt values through a tunable scaling factor, leading to faster convergence without sacrificing performance. The number of GLOPS are calculated using fvcore flop count tool[1].

IV. Experimental Setup

For all of our experiments, we utilize vanilla transformer architectures, specifically ViT-Small and ViT-Base [10], [9], [8]. We initialize the model using pretrained weights from the VideoMAE method [9], [8]. Unless otherwise noted, we employ the AdamW optimizer [55]. The models are fine-tuned for 100 epochs, which includes a 10-epoch warm-up period, followed by a cosine scheduler to reduce the learning rate to 0. As part of our data augmentation strategy, we apply Random Augment [56]. The base learning rate is specified for a batch size of 256, and it is automatically scaled for different batch sizes as needed. To expedite fine-tuning, we use a batch augmentation factor of 2, meaning that for each loaded video, we create two random views. For evaluation, we adhere to the common practice of multi-view testing. We use K temporal clips to span the video length (with K=3 for SSv2, 10 for HMDB51, and 5 for UCF101 [9], [8], [44]), and for each clip, we take three spatial views to cover the entire image. The final prediction is obtained by averaging the results across all views.

V. Results

A. Ablation Experiments

For all ablation experiments, we employ the ViT-Small (21.947M params.) [10] model pretrained on the SSv2 (Something-Something v2) dataset [2]. The pretrained weights are obtained from VideoMAE [9]. We report both top-1 and top-5 accuracy metrics for action classification on the test set of the SSv2 dataset [2], unless specified otherwise. The best (i.e., default) configuration is highlighted in grey color and used in the main analysis with ViT-Base architecture. During evaluation, we adhere to a common protocol that involves a 2-view, 3-temporal-clip evaluation. This protocol involves extracting multiple temporal clips and views from each video and aggregating the predictions to produce the final classification result.

[1] https://github.com/facebookresearch/fvcore/blob/main/fvcore/nn/flop_count.py

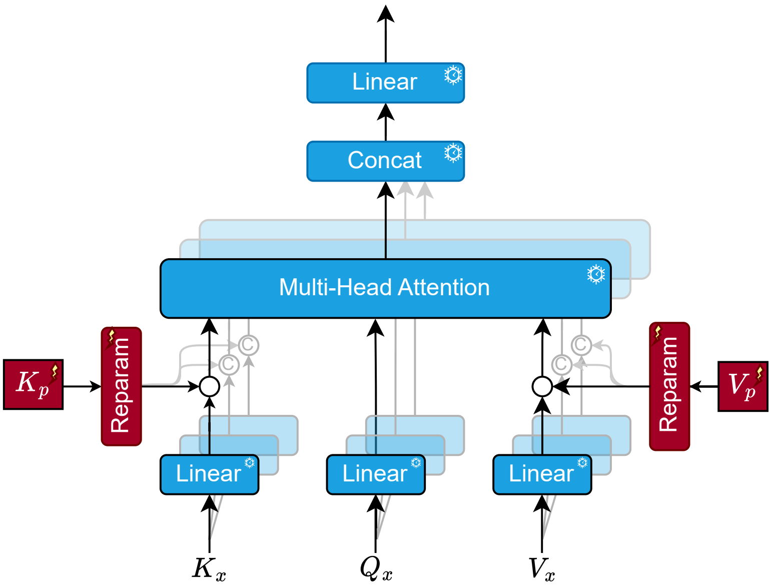


Fig. 2: The proposed APT architecture. Modules in **cyan** color are frozen and not updated during fine-tuning. Modules in **purple** color are learned and updated during fine-tuning. Learnable prompts (K_p and V_p) are directly injected into the MHA by concatenating them with the Keys (K_x) and Values (V_x), while the Queries (Q_x) remain unchanged.

TABLE I: Effect of prompt length (n_p) on APT with ViT-Small backbone on SSv2.

n_p	Params. (M)	Top-1 (%)	Top-5 (%)
400	0.692 (3.2%)	55.81	83.17
600	1.003 (4.5%)	56.52	83.76
800	1.316 (6.0%)	56.74	83.87
1000	1.628 (7.4%)	57.18	84.21
1200	1.940 (8.8%)	57.44	84.24
1400	2.252 (10.3%)	**57.57**	**84.15**
1600	2.564 (11.7%)	57.07	84.14
2000	3.188 (14.5%)	57.55	84.03

a) Effect of Prompt Length: Table I shows the impact of varying the number of prompts (i.e., prompt length n_p) in the proposed APT on the tunable parameters, GFLOPs, and Top-1/Top-5 accuracy on SSv2. Increasing the prompt length initially leads to an improvement in both Top-1 and Top-5 accuracy. The peak Top-1 accuracy of 57.57% and Top-5 accuracy of 84.15% are achieved with 1400 prompts. However, further increasing the number of prompts to 1600 or 2000 does not lead to a significant improvement in accuracy, and in some cases, the Top-1 accuracy slightly decreases. This suggests that there may be an optimal number of prompts for this model and dataset, beyond which the additional prompts do not contribute significantly to performance improvement.

b) Effect of Prompt Reparameterization: As previously discussed, prompt tuning methods are highly sensitive to hyperparameter selection, especially with regards to the choice of learning rate and weight decay parameters. To enhance the robustness of APT to hyperparameter selection, we introduce a prompt reparameterization technique that requires minimal computational overhead and introduces only a small number of tunable parameters. In this analysis, we examine the convergence of training loss during fine-tuning both with and without the application of prompt reparameterization. As illustrated in Fig. 3, utilizing scaled reparameterization leads to faster convergence and a lower training loss compared to the absence of reparameterization. This approach results in improved performance, as evidenced by a **1.02%** gain in top-1 accuracy (**55.81%** vs. **54.85%**) and a **0.93%** gain in top-5 accuracy (**83.17%** vs. **82.24%**). Notably, the observed improvement is achieved with a negligible increase in tunable parameters ($2 \times 400 \times 12 = 9600$, representing only **1.4%** of the total tunable parameters). This enhancement in both performance and convergence speed is particularly valuable for video-based downstream applications, where fine-tuning on

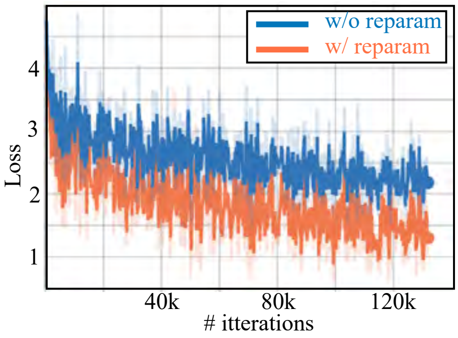


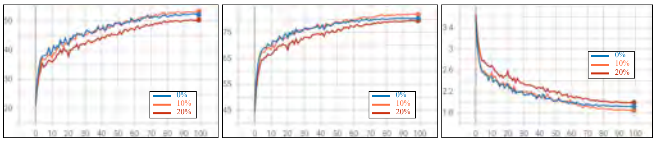
Fig. 3: Training loss comparison: with and without prompt reparameterization. The use of scaled reparameterization results in faster convergence and a lower training loss.

video datasets is both time-consuming and computationally demanding.

TABLE II: Performance for different dropout rates applied to K_p and V_p.

Dropout	Top-1 (%)	Top-5 (%)
0%	55.21	82.66
10%	**57.57**	**84.15**
20%	52.14	80.80

c) Effect of Dropout on Attention Prompts: Through experimentation, we found that applying dropout to attention prompts K_p and V_p improved both top-1 and top-5 accuracy. Dropout serves as a regularization technique, reducing overfitting and promoting diverse prompt representations. Table II shows action recognition performance for different dropout rates, with the best results achieved at a **10% rate**, which balanced informativeness and variability in the prompts. Overall, dropout in prompt tuning enhances classification performance and generalization.



(a) Top-1 (val-set) (b) Top-5 (val-set) (c) Val-loss

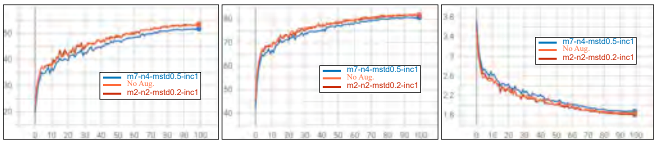
Fig. 4: Effect of dropout on attention prompts. Applying 10% dropout to attention prompts results in the best performance on both validation and test sets.

Furthermore, Figure 4 illustrates the curves for Top-1 accuracy, Top-5 accuracy, and validation loss at different dropout levels. The figures reveal that, initially, the accuracies for both Top-1 and Top-5 were higher when no dropout was applied. However, as training progressed, the results improved significantly with a dropout rate of 10%. This indicates that utilizing dropout in attention prompts effectively prevents the network from overfitting to the prompts and plays a crucial role in optimizing attention prompt tuning to achieve superior test results.

TABLE III: Effect of Random Augmentations (RA).

Configuration	Top-1 (%)	Top-5 (%)
No Aug.	55.81	83.17
RA (m2-n2-mstd0.2-inc1)	**55.82**	**83.42**
RA(m7-n4-mstd0.5-inc1)	53.94	82.23

d) Effect of Random Augmentations: During full-fine tuning, extensive data augmentations are typically applied through Random Augment [56] to enhance the model's generalization ability by introducing diversity into the training data. However, our experiments with APT, as well as with linear probing and VPT, revealed a different behavior. We observed that applying the same extent of Random Augmentations during APT, linear probing, and VPT did not lead to performance improvements and, in some instances, resulted in diminished performance. We hypothesize that this behavior is attributable to the fact that the backbone remains frozen during APT, linear probing, and VPT, thereby limiting the capacity for overfitting. As training relies on a relatively small fraction of tunable parameters, the need for aggressive data augmentation is reduced.



(a) Top-1 (val-set) (b) Top-5 (val-set) (c) Val-loss

Fig. 5: Effect of random augmentations on training APT. Higher magnitude of augmentations in training results in lower action recognition performance while light / no augmentations results in good action recognition performance.

TABLE IV: Effect of WD.

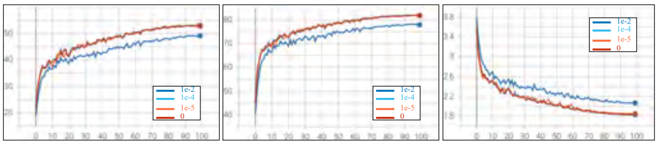
WD	Top-1 (%)	Top-5 (%)
1e-2	51.43	79.98
1e-4	55.47	**83.35**
1e-5	**55.81**	83.17
0	55.24	83.05

e) Effect of Weight Decay (WD): During the fine-tuning of APT, we investigated the impact of weight decay regularization on model performance. Our experiments revealed that the choice of weight decay value also plays a crucial role in the effectiveness of APT.

Specifically, as shown in Figure 6, applying an excessively high weight decay value (like $1e-2$ utilized in full-tuning) led to a deterioration in performance, likely due to over-constraining the attention prompts values and which hinders its ability to change the input token representations in the desired direction. Conversely, completely removing weight decay regularization (i.e., setting weight decay to zero) also resulted in sub-optimal performance, as the attention prompts become more susceptible to overfitting to the training data.

Through hyperparameter tuning, we determined that a moderate weight decay value of $1e-5$ yielded improved performance, achieving an optimal balance between preventing overfitting and allowing flexibility to attention prompts. The performance results for different weight decay values are summarized in Table IV.

f) Effect of Attention Prompt Placement: In the default configuration of APT, we append attention prompts to the MHA at each Transformer Block within the ViT. While this placement strategy yields the best performance, our experiments reveal that the placement of attention prompts at different depths of the Transformer Blocks has varying contributions to the overall performance. Specifically, we find that attention prompts placed at deeper Transformer Blocks



(a) Top-1 (val-set)

(b) Top-5 (val-set)

(c) Val-loss

Fig. 6: Effect of weight decay regularization on training APT. Applying small weight decay regularization results in better performance.

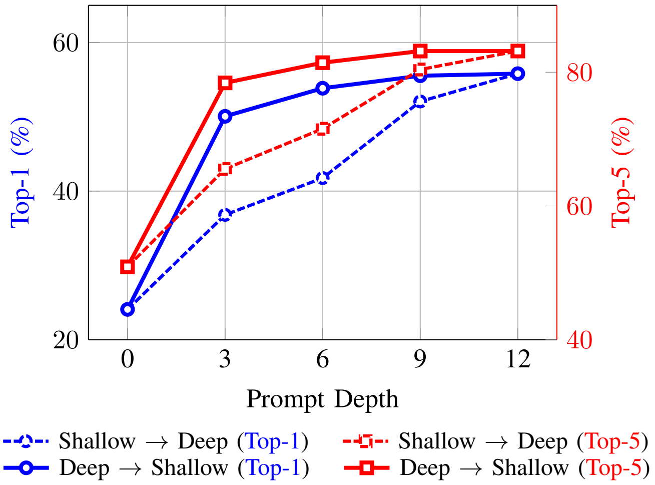


Fig. 7: Effect of prompt depth with APT.

(i.e., blocks closer to the output layer) have a more substantial impact on performance compared to prompts placed at shallower Transformer Blocks (i.e., blocks closer to the input layer). This observation suggests that refining feature representations at deeper transformer layers is more important for downstream performance than the features at shallow transformer layers. Figure 7 illustrates the performance improvement achieved by incrementally appending attention prompts, starting from the deepest Transformer Blocks and progressively extending the placement to shallower blocks. The results highlight the importance of attention prompt placement in the APT framework and its influence on model performance.

B. Computational Complexity

We compare computational and parameter efficiency of APT with VPT [16] in Fig. 8. The results demonstrate that injecting prompts directly into the multi-head attention significantly reduces the number of tunable parameters and GFLOPs, unlike VPT, which concatenates prompts with the input tokens. The reduced number of GFOPs of APT is beneficial for more efficient and scalable deployment of pre-trained transformer models for video applications.

C. Main Analysis

1) Results with VideoMAE pre-trained backbones: Table V compares the results of our APT with linear probing, full-tuning, and existing parameter-efficient methods (VPT [16] and AdaptFormer [15]) on three action recognition datasets: SSv2 [2], UCF101 [3], and HMDB51 [4]. Note that, for these experiments, we utilize VideoMAE [9] pre-trained backbones which is currently the state-of-the-art pre-trained models for ViT backbones.

For the UCF101 and HMDB51 datasets, our APT achieves results close to full-tuning while using less than 1% of tunable parameters. Notably, for UCF101, APT achieves higher accuracy than full-tuning (97.7% vs. 96.1% top-1) with only 200 attention prompts, equivalent to 0.45% of tunable parameters. On the HMDB51 dataset, APT results are slightly lower than full-tuning (70.12% vs. 73.3%), but still very close. However, APT outperforms VPT and AdaptFormer by a significant margin, reducing the performance gap between full-tuning and parameter-efficient methods.

When comparing results on the SSv2 dataset, which is larger and has more complex action classes, there is approximately a 10% performance gap between APT and full-tuning. This difference may be attributed to the complexity of the action classes in the dataset. However, when comparing APT with VPT and AdaptFormer, APT achieves better results with significantly fewer tunable parameters (60.79% top-1 with 0.81% tunable parameters vs. 59.20% top-1 with 1.46% tunable parameters).

Overall, APT significantly reduces the gap between full-tuning and existing parameter-efficient methods, establishing itself as the new state-of-the-art for parameter-efficient tuning in action recognition. Additionally, APT further reduces the number of tunable parameters required to adopt pre-trained transformer backbones for video-based downstream applications.

2) Results with other pre-trained backbones: To demonstrate that the proposed APT is not dependent on a specific pre-trained method and performs well with other pre-trained approaches, such as CLIP and supervised pre-trained models, we conducted experiments on SSv2 using the ViT-Base backbone. The results of these experiments are summarized in Table VI. From the data presented in the table, it is evident that the proposed APT achieves SOTA results even when compared to other pre-trained approaches, all while maintaining parameter efficiency.

Taken together, all the aforementioned results affirm APT's performance as a better parameter-efficient training method than existing approaches, particularly well-suited for handling spatiotemporal data – which often involves training with a large number of tokens than images.

VI. Conclusion and Future Work

In this paper, we proposed APT, a variant of VPT, for efficient adaptation of pre-trained transformer models in downstream tasks. While VPT offers flexibility and can be easily adapted to various applications, it requires a significant number of prompts to achieve satisfactory results in video-based tasks like action classification, leading to increased FLOPs and latency during inference and hindering parameter efficiency. In contrast, our APT approach injects attention prompts directly into the attention mechanism, resulting in reduced FLOPs and improved performance compared to VPT

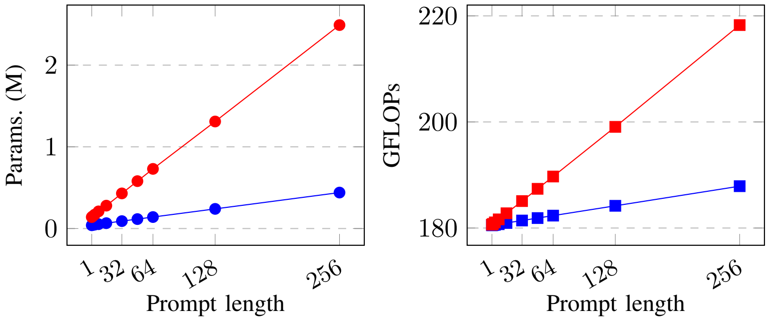


Fig. 8: Comparison of tunable params. (left) and GFLOPs (right) with prompt length for **VPT** and **APT (Ours)**. The proposed APT model demonstrates a significant reduction in both the number of tunable parameters and GFLOPs, highlighting its efficiency and optimized performance compared to VPT.

TABLE V: Comparison of APT performance with other parameter efficient tuning methods. We use VideoMAE [9], [8] pre-trained ViT-Base backbone as the initialization.

Method	Tuned Params. (M)	GFLOPs	**SSv2**		**UCF101**		**HMDB51**	
			top-1 %	top-5 %	top-1 %	top-5 %	top-1 %	top-5 %
Linear-Probing	0.07 (0.08%)	180.03	29.23	N/A	94.3	99.5	49.84	75.9
VPT-1	0.08 (0.09%)	180.50	43.73	N/A	94.1	N/A	52.76	N/A
VPT-5	0.12 (0.14%)	180.60	45.87	N/A	94.7	N/A	53.66	N/A
AdaptFomrer-1	0.10 (0.12%)	180.51	50.03	N/A	94.3	99.5	51.68	78.3
AdaptFomrer-4	0.15 (0.17%)	180.60	54.70	N/A	94.5	99.5	51.81	78.9
AdaptFomrer-64	1.26 (1.46%)	182.33	59.02	N/A	94.5	99.7	55.69	79.4
APT-1 (ours)	0.081 (0.09%)	180.51	42.92	72.18	96.4	99.8	58.23	83.09
APT-5 (ours)	0.087 (0.09%)	180.63	49.15	78.50	96.5	99.9	61.43	87.12
APT-100 (ours)	0.235 (0.27%)	183.38	57.15	84.85	97.6	100.0	67.58	89.67
APT-200 (ours)	0.391 (0.45%)	186.27	59.43	86.39	**97.7**	**100.0**	68.17	88.63
APT-400 (ours)	0.703 (0.81%)	192.05	**60.79**	**87.15**	97.7	99.9	**70.12**	**90.78**
Full-tuning	86.36 (100%)	180.03	70.8	92.4	96.1	99.4	73.3	N/A

TABLE VI: The comparison of APT with other parameter-efficient methods using other pre-trained approaches, such as CLIP [57] and supervised pre-training on K400 [58] with the ViT-Base backbone on the SSv2 dataset. Reported results indicate top-1 accuracy.

Method	Tune Params. (M)	Pre-training method	
		CLIP (Top-1 %)	Supervised K400 (Top-1 %)
Linear Probing	0.07 (0.08%)	21.9	29.3
VPT	0.08 (0.09%)	45.1	33.1
AdaptFormer-4	0.15 (0.17%)	59.1	42.3
AdaptFormer-64	1.3 (1.46%)	63.4	46.0
ST-Adapter	7.20 (8.33%)	66.3	NA
AIM	14.3 (16.56%)	66.4	NA
APT-200 (ours)	0.39 (0.45%)	64.8	47.2
APT-400 (ours)	0.70 (0.81%)	**66.8**	50.3
APT-600 (ours)	1.01 (1.17%)	**66.8**	**50.4**
Full-tuning	86.36 (100%)	66.9	55.7

and other parameter-efficient methods like AdaptFormer. We also introduced a novel prompt reparameterization technique to enhance tuning convergence and robustness to hyperparameter selection. Overall, APT achieves superior results, sometimes surpassing full-tuning, with high parameter efficiency and minimal increase in GLOPs compared to existing approaches. However, a limitation of APT and other parameter-efficient methods is the potential increase in FLOPs and latency during deployment, which warrants further exploration as future work.

References

[1] Elham Shabaninia, Hossein Nezamabadi-pour, and Fatemeh Shafizadegan. Transformers in action recognition: A review on temporal modeling. *arXiv preprint arXiv:2302.01921*, 2022.

[2] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyńska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense, June 2017. arXiv:1706.04261 [cs].

[3] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild, December 2012. arXiv:1212.0402 [cs].

[4] H Kuehne, H Jhuang, E Garrote, T Poggio, and T Serre. HMDB: A Large Video Database for Human Motion Recognition, 2011.

[5] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, December 2017. arXiv:1706.03762 [cs].

[7] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal Contrastive Video Representation Learning. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6960–6970, Nashville, TN, USA, June 2021. IEEE.

[8] Wele Gedara Chaminda Bandara, Naman Patel, Ali Gholami, Mehdi Nikkhah, Motilal Agrawal, and Vishal M. Patel. AdaMAE: Adaptive Masking for Efficient Spatiotemporal Learning with Masked Autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Canada, November 2022. arXiv:2211.09120 [cs].

[9] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked Autoencoders are Data-Efficient Learners for Self-Supervised Video Pre-Training. In *Advances in Neural Information Processing Systems*, July 2022. arXiv:2203.12602 [cs].

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. arXiv:2010.11929 [cs].

[11] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[12] Kai Han, Yunhe Wang, Hanting Chen, Xinghao Chen, Jianyuan Guo, Zhenhua Liu, Yehui Tang, An Xiao, Chunjing Xu, Yixing Xu, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1):87–110, 2022.

[13] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10699–10709, 2022.

[14] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked Autoencoders As Spatiotemporal Learners, May 2022. arXiv:2205.09113 [cs].

[15] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition, October 2022. arXiv:2205.13535 [cs].

[16] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning, July 2022. arXiv:2203.12119 [cs].

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

[18] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[19] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[20] Xiang Lisa Li and Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation, January 2021. arXiv:2101.00190 [cs].

[21] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[22] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4305–4314, 2015.

[23] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):677–691, 2017.

[24] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1933–1941, 2016.

[25] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[26] João Carreira and Andrew 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, 2017.

[27] 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.

[28] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin 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.

[29] 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.

[30] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[31] Vivek Veeriah, Naifan Zhuang, and Guo-Jun Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015.

[32] Wele Gedara Chaminda Bandara and Vishal M. Patel. A transformer-based siamese network for change detection. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210, 2022.

[33] Wele Gedara Chaminda Bandara and Vishal M. Patel. Hypertransformer: A textural and spectral feature fusion transformer for pansharpening. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1767–1777, June 2022.

[34] Malsha V. Perera, Wele Gedara Chaminda Bandara, Jeya Maria Jose Valanarasu, and Vishal M. Patel. Transformer-based sar image despeckling. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 751–754, 2022.

[35] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding Robustness of Transformers for Image Classification, 2021.

[36] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation, 2021.

[37] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L. Yuille, and Yuyin Zhou. TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation, February 2021. arXiv:2102.04306 [cs].

[38] Wele Gedara Chaminda Bandara and Vishal M. Patel. A Transformer-Based Siamese Network for Change Detection. In *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, pages 207–210, July 2022. ISSN: 2153-7003.

[39] Bowen Zhang, Shuyang Gu, Bo Zhang, Jianmin Bao, Dong Chen, Fang Wen, Yong Wang, and Baining Guo. StyleSwin: Transformer-Based GAN for High-Resolution Image Generation, 2022.

[40] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image Transformer. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4055–4064. PMLR, July 2018. ISSN: 2640-3498.

[41] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative Pretraining From Pixels. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1691–1703. PMLR, July 2020.

[42] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked Autoencoders Are Scalable Vision Learners, December 2021. arXiv:2111.06377 [cs].

[43] Rohit Girdhar, Alaaeldin El-Nouby, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. OmniMAE: Single Model Masked Pretraining on Images and Videos, June 2022. arXiv:2206.08356 [cs, stat].

[44] Sepehr Sameni, Simon Jenni, and Paolo Favaro. Spatio-Temporal Crop Aggregation for Video Representation Learning, November 2022. arXiv:2211.17042 [cs].

[45] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. ST-Adapter: Parameter-Efficient Image-to-Video Transfer Learning, October 2022. arXiv:2206.13559 [cs].

[46] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022.

[47] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. *arXiv preprint arXiv:2302.03024*, 2023.

[48] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning, 2022.

[49] Guangyu Sun, Matias Mendieta, Jun Luo, Shandong Wu, and Chen Chen. Fedperfix: Towards partial model personalization of vision transformers in federated learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4988–4998, 2023.

[50] Bruce Yu, Jianlong Chang, Lingbo Liu, Qi Tian, and Chang Wen Chen. Towards a unified view on visual parameter-efficient transfer learning, 2023.

[51] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[52] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning, 2021.

[53] Cheng Han, Qifan Wang, Yiming Cui, Zhiwen Cao, Wenguan Wang, Siyuan Qi, and Dongfang Liu. E^2vpt: An effective and efficient approach for visual prompt tuning, 2023.

[54] Zhixing Tan, Xiangwen Zhang, Shuo Wang, and Yang Liu. Msp: Multi-stage prompting for making pre-trained language models better translators. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6131–6142, 2022.

[55] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, January 2019. arXiv:1711.05101 [cs, math].

[56] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space, November 2019. arXiv:1909.13719 [cs].

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[58] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The Kinetics Human Action Video Dataset, May 2017. arXiv:1705.06950 [cs].