Dual-path Adaptation from Image to Video Transformers

Jungin Park[1∗] Jiyoung Lee[2∗] Kwanghoon Sohn[1,3†]
[1]Yonsei University [2]NAVER AI Lab [3]Korea Institute of Science and Technology (KIST)
{newrun, khsohn}@yonsei.ac.kr lee.j@navercorp.com

Abstract

In this paper, we efficiently transfer the surpassing representation power of the vision foundation models, such as ViT and Swin, for video understanding with only a few trainable parameters. Previous adaptation methods have simultaneously considered spatial and temporal modeling with a unified learnable module but still suffered from fully leveraging the representative capabilities of image transformers. We argue that the popular dual-path (two-stream) architecture in video models can mitigate this problem. We propose a novel DUALPATH *adaptation separated into spatial and temporal adaptation paths, where a lightweight bottleneck adapter is employed in each transformer block. Especially for temporal dynamic modeling, we incorporate consecutive frames into a grid-like frameset to precisely imitate vision transformers' capability that extrapolates relationships between tokens. In addition, we extensively investigate the multiple baselines from a unified perspective in video understanding and compare them with* DUALPATH*. Experimental results on four action recognition benchmarks prove that pretrained image transformers with* DUALPATH *can be effectively generalized beyond the data domain.*

1. Introduction

Recognizing *when*, *where*, and *what* happened is a fundamental capability in the human cognition system to understand our natural world. The research for video understanding inspires such capability for machine intelligence to comprehend scenes over time flow. Over the last decade, the development of deep neural networks [10,33,66,73] has contributed towards advances in video understanding.

Vision Transformer (ViT) [17] has recently emerged, making an upheaval in the research field of computer vision. ViT and its variants [16, 46, 72, 78] have demonstrated remarkable generalizability and transferability of their representations with scaled-up foundation models [34,53,63,69, 77,79] and large-scale web-collected image data (*e.g.* JFT-3B [79], LAION-5B [58]). To capitalize on well-trained visual foundation models, finetuning entire parameters of the pretrained models with task-specific objectives has been the most popular transfer technique. However, it requires high-quality training data and plenty of computational resources to update the whole parameters for each downstream task, making overwhelming efforts for training. While partial finetuning [28], which trains additional multilayer perceptron (MLP) layers to the top of the model, has also been widely used for affordable training costs, unsatisfactory performance has been pointed out as a problem.

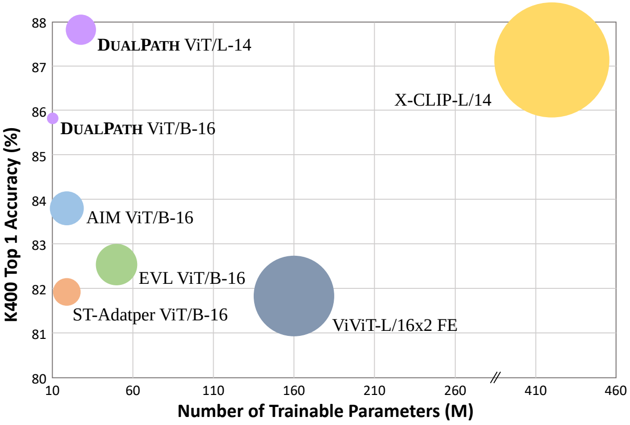


Figure 1. Performance comparison on the Kinetics-400 [38] dataset. We depict the action recognition performance (vertical axis, %) with respect to the number of trainable parameters (horizontal axis). The size of circles indicates GFLOPs for inference.

Most recently, parameter-efficient transfer learning (PETL) methods [25, 26, 31, 32, 41, 61] have been proposed as an alternative to finetuning in the natural language processing area to adapt the large-scale language model, such as GPT series [8, 54, 55] and T5 [56], for each task. They have successfully attained comparable or even surpassing performance to full-tuning parameters by learning a small number of extra trainable parameters only while keeping the original parameters of the pretrained model frozen. Thanks to their effectiveness and simplicity, they have been ex-

∗ Equal contributions. † Corresponding author.
Official code: https://github.com/park-jungin/DualPath

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (NRF2021R1A2C2006703).

tended to vision models by applying prompt-based methods [3, 35] and adapter-based methods [11, 49, 64]. They have efficiently adapted pretrained models to downstream tasks with significantly reduced tuning parameters, but most of these works mainly focus on transferring image models to image tasks [3, 11, 35, 49] and vision-language models to vision-language tasks [64]. Inspired by the advances of the prior arts, we raise two conceivable questions: **(1)** Is it possible to transfer the parameters of the image foundation model to another video domain? **(2)** Is it also possible the transferred model performs comparably to the carefully designed video models that take the spatiotemporal nature of the video into account?

While image models have demonstrated strong spatial context modeling capabilities [34, 53, 63, 79], video transformer models [1, 47, 57, 76] require a more complex architecture (*e.g.* 539 vs 48912 GFLOPs [57]) with a large number of parameters (*e.g.* 84M vs 876M parameters [76]) than ViT for temporal context reasoning. Therefore, the challenge in transferring image models for video understanding is to encode the temporal context of videos while leveraging the discriminative spatial context of the pretrained image models. A naive solution is to finetune image models on a video dataset by directly applying previous prompt-/adapter-based approaches [3, 11, 35, 49]. However, these approaches inevitably ignore the temporal context in videos because they bridge only the spatial contexts between image and video data.

In this paper, we propose a novel adapter-based dual-path parameter efficient tuning method for video understanding, namely **DUALPATH**, which consists of two distinct paths (*spatial* path and *temporal* path). For both paths, we freeze the pretrained image model and train only additional bottleneck adapters for tuning. The **spatial path** is designed to encode the spatial contexts that can be inferred from the appearance of individual frames with the minimum tuning of the pretrained image model. To reduce the computation burden, we sparsely use the frames with a low frame rate in the spatial path. The **temporal path** corresponds to the temporal context that should be encoded by grasping the dynamic relationship over several frames sampled with a high frame rate. Especially for two reasons, we construct a grid-like frameset that consists of consecutive low-resolution frames as an input of the temporal path: (i) preventing computational efficiency loss caused by calculating multiple frames simultaneously; (ii) precisely imitating the ViT's ability for extrapolating global dependencies between input tokens. To compare our DUALPATH with existing methods broadly, we implement several baselines with a unified perspective on recent domain-specific PETL approaches [11, 35, 49]. Extensive experiments on several action recognition benchmarks [23, 38, 39, 42] demonstrate the effectiveness and high efficiency of our DUALPATH, achieving comparable and even better performance than the baselines and prior video models [1, 7, 18, 22, 40, 43, 47, 57, 68, 76]. We achieve these results with extremely low computational costs for both training and inference, as demonstrated in Fig. 1.

2. Related Work

Pretraining vision models. To address the burdens of collecting large-scale labeled datasets for supervised learning [15, 63, 79], self-supervised learning methods [12, 13, 24, 27, 28, 74] have been introduced to learn general-purpose visual representations from unlabeled data. Similarly, self-supervised learning methods for videos have also been proposed with large-scale unlabeled video/video-language data [20, 51, 52, 62, 65, 68, 75]. However, collecting even unlabeled video-language pairs is still quite costly compared to image-language pairs. In addition, pretraining video models require more computational power than images. We thus take advantage of the powerful pretrained image-based models for efficient video understanding.

Video action recognition. Action recognition is one of the most fundamental research topics for video understanding. Early works have been built upon convolution neural networks (CNNs) [10, 21, 44, 66, 73] to effectively infer the spatiotemporal context for action recognition. Since Vision Transformer (ViT) [17] has become a new paradigm in computer vision, transformers for video understanding have been actively studied by extending pretrained image models. The pretrained image transformers have been used to initialize the part of the video transformers [1, 7, 76, 80] or inflated to the video transformers [47]. While transformers have demonstrated superior performance on video action recognition, they require full finetuning on video datasets, making the training inefficient.

Parameter-efficient transfer learning (PETL). To address the memory and parameter inefficiency of full-/partial-finetuning, PETL has first introduced in natural language processing (NLP) [6, 25, 26, 31, 32, 41, 61]. The main objective of PETL is to attain comparable or surpassing performance on downstream tasks by finetuning with only a small number of trainable parameters. Although PETL approaches [3, 11, 11, 35, 36, 49, 64] have recently been studied in computer vision, they are 'blind' to other modalities such that image models are used for image tasks, and so are the other modalities. In contrast, we share the same objective as recent works for image-to-video transfer learning [37, 45, 48, 50], demonstrating the pretrained image models can be good video learners. However, they have several limitations in terms of parameter and computational efficiency. For example, [45] learned an extra decoder that contains 3D convolution layers and cross-frame attention, and [50] inserted additional depth-wise 3D convolution layers between the down-/up-projection layers of the adapter to perform temporal reasoning, inducing computational inefficiency. The most recent works [37,48] require an additional text encoder branch as a classifier. Moreover, they have computational efficiency proportional to the temporal resolution. Our DUALPATH accomplishes more efficient spatiotemporal modeling while achieving higher performance.

3. Preliminaries and Baselines

3.1. Vision transformers for video

We briefly describe how to apply vision transformers for video understanding below. Following [67], given a set of T frames in a video, we split each frame into N patches of size $(P \times P)$ and tokenize them using a linear projection, such that

$$\mathbf{X}_t = [\mathbf{x}_t\{\texttt{[CLS]}\}, \mathbf{x}_t^1, \mathbf{x}_t^2, \cdots, \mathbf{x}_t^N] + \mathbf{p}, \tag{1}$$

where $\mathbf{X}_t$ is a set of tokens for the t-th frame, and $\mathbf{x}_t\{\texttt{[CLS]}\}$ and $\mathbf{p}$ denote a learnable class token and a learned positional embedding respectively. We feed $(N+1)$ tokens of each frame to a sequence of L transformer blocks, and the output of the l-th block $\mathbf{h}_{l,t}$ can be derived by the following equations:

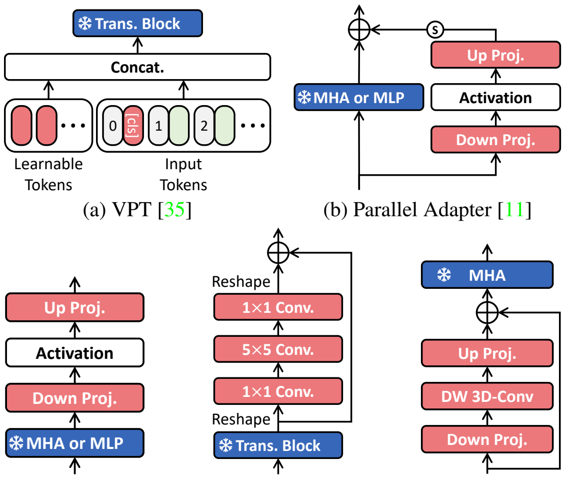
$$\begin{aligned} \mathbf{z}_{l,t} &= \mathbf{h}_{l-1,t} + \texttt{MHA}(\texttt{LN}(\mathbf{h}_{l-1,t})), \\ \mathbf{h}_{l,t} &= \mathbf{z}_{l,t} + \texttt{MLP}(\texttt{LN}(\mathbf{z}_{l,t})), \end{aligned} \tag{2}$$

where $\texttt{LN}, \texttt{MHA}$, and $\texttt{MLP}$ denote a layer normalization [2], multi-head attention [67], and a multilayer perceptron operation, respectively. We apply layer normalization to the learned T class tokens from the final transformer block and treat them as a set of frame representations.

To take minimal temporal modeling into account the following baselines [11, 35, 49], we employ a temporal transformer block followed by a full-connected (FC) layer as a classifier for video action recognition, similar to [37]. We add learnable temporal positional embeddings $\mathbf{p}_{\text{temp}}$ to the frame representations (*i.e.*, $\mathbf{x}_t\{\texttt{[cls]}\} \leftarrow \mathbf{x}_t\{\texttt{[cls]}\} + \mathbf{p}_{\text{temp}}$) and feed them into the transformer classifier. For the ST-adapter [50], we use a single FC layer as a classifier.

3.2. Baselines

The objective of our work is to transfer the superiority of vision transformers pretrained on large-scale image datasets to the video domain through efficient finetuning with a small number of learnable parameters, while freezing the pretrained parameters. To compare with other methods, we generalize four recent PETL methods to the video domain only with the least possible transformation; (1) VPT [35] (2) AdaptFormer [11] (3) Pro-tuning [49] (4) ST-adapter [50]. The most of works have been originally proposed to adapt pretrained image models to downstream image tasks [11, 35, 49] and video models to video tasks [11], by learning visual prompt tokens [35], adapter blocks [11] and prompt prediction blocks [49]. Only the ST-adapter [50] has proposed image-to-video transfer learning. In this section, we describe baselines for image-to-video transfer learning in detail. For brevity, we leave out the subscripts in Eq. 2 and arouse them as needed. In addition, we represent the **learnable** and **frozen** parameters in red and blue colors, respectively.



(a) VPT [35] (b) Parallel Adapter [11]

(c) Serial Adapter [11] (d) Prompt Block [49] (e) ST-Adapter [50]

Figure 2. Baselines for image-to-video transfer learning. (a) Visual Prompt Tuning [35]; (b) Parallel adapter and (c) serial adapter [11]; (d) Pro-Tuning [49]; and (e) ST-Adapter [50].

Visual prompt tuning (VPT) [35] prepends K trainable prompt tokens to the input space of every transformer block[1] while keeping pretrained parameters frozen. The input tokens for each transformer block can be written as:

$$\tilde{\mathbf{h}} = [\mathbf{e}; \mathbf{h}], \tag{3}$$

where $\mathbf{e} \in \mathbb{R}^{K \times d}$ is a set of trainable visual prompt tokens and d is a channel dimension of the original token.

AdaptFormer [11] learns a trainable bottleneck module [26, 31]. The intermediate feature $\mathbf{z}$ in Eq. 2 of each transformer block is fed into the AdapterMLP that consists of the original MLP layers and a bottleneck structure (parallel adapter in Fig. 2b). The output of the AdaptFormer block can be formulated by:

$$\begin{aligned} \tilde{\mathbf{z}} &= \sigma(\texttt{LN}(\mathbf{z}) \cdot \mathbf{W}_{\text{down}}) \cdot \mathbf{W}_{\text{up}}, \\ \mathbf{h} &= \mathbf{z} + \texttt{MLP}(\texttt{LN}(\mathbf{z})) + s \cdot \tilde{\mathbf{z}}, \end{aligned} \tag{4}$$

where $\mathbf{W}_{\text{down}}$, $\mathbf{W}_{\text{up}}$ are trainable down- and up-projection matrices, $\sigma(\cdot)$ is an activation function, and s is a scaling factor.

Pro-tuning [49] predicts task-specific vision prompts $\mathbf{v}$ from the output of each transformer block using consecu-

[1] While the original work also presented a shallow version (VPT-Shallow) that inserts prompt tokens to the first layer, we explore a deep version (VPT-Deep) only.

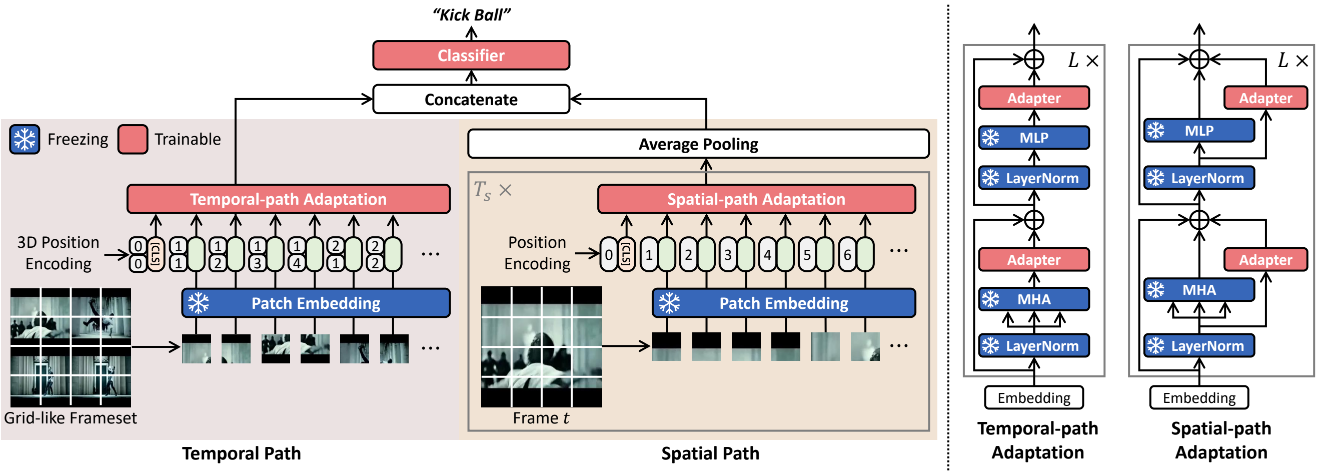


Figure 3. Overall architecture of DUALPATH. The pretrained image transformer takes a grid-like frameset and T_s frames as inputs. We learn the temporal and spatial contexts separately through two distinct paths. During training, we only update newly introduced adapters and the classifier while freezing the pretrained layers.

tive 2D convolution layers. The output of each block is reshaped as $\mathbb{R}^{P \times P \times C}$ to apply 2D convolutions and the final representation is derived by adding $\mathbf{v}$ and $\mathbf{h}$:

$$\begin{gathered} \mathbf{v} = \texttt{Reshape}(\sigma(\texttt{Conv2d}(\texttt{Reshape}(\mathbf{h})))), \\ \tilde{\mathbf{h}} = \mathbf{h} + \mathbf{v}, \end{gathered} \tag{5}$$

where $\texttt{Conv2D}$ consists of 1×1 convolution layer followed by 5×5 depth-wise convolution [14] and 1×1 convolution. **ST-adapter** [50] inserts a depth-wise 3D convolution layer between the down-projection layer and the activation function of the adapter. Different from the conventional adapters (*e.g.* AdaptFormer [11]), the ST-adapter takes tokens for all frames to enable the model to capture temporality in videos. The output of the ST-adapter can be represented as:

$$\begin{gathered} \tilde{\mathbf{z}}_t = \texttt{LN}(\mathbf{z}_t), \\ \hat{\mathbf{z}} = \sigma(\texttt{D-Conv3d}([\tilde{\mathbf{z}}_1 \cdot \mathbf{W}_{\text{down}}, \cdots, \tilde{\mathbf{z}}_T \cdot \mathbf{W}_{\text{down}}])) \cdot \mathbf{W}_{\text{up}}, \\ \mathbf{h}_t = \mathbf{z}_t + \texttt{MLP}(\tilde{\mathbf{z}}_t) + s \cdot \hat{\mathbf{z}}_t, \end{gathered} \tag{6}$$

where $\texttt{D-Conv3d}$ denotes the depth-wise 3D convolution layer. Note that the same down-projection matrix $\mathbf{W}_{\text{down}}$ is applied to all tokens regardless of the frame index t.

We emphasize that most of the baselines [11, 35, 49] have *not* concerned with temporal modeling. Even though ST-Adapter [50] has employed depth-wise 3D convolution layers between linear projections, it results in high computational cost. To entirely leverage a simple and efficient architecture of the adapter [11], we incorporate the dual-path design into the pretrained image transformers.

4. Method

The dual-path design (also called two-stream) is well-known architecture in CNN-based models for video recognition [10, 21, 60]. They have commonly used an optical flow [10, 60] or multiple frames with a high temporal resolution [21] to capture rapidly changing motion. Despite the effectiveness of dual-path architecture, it has yet to be explored with the transformer due to high computational costs. In this work, we propose a novel PETL method, called DUALPATH, comprised of *spatial* and *temporal* path adaptation. To the best of our knowledge, our DUALPATH is the first attempt to explicitly build the two-stream architecture upon the transformer while maintaining the computational cost similar to the single-stream architecture. The overall framework is depicted in Fig. 3.

4.1. Spatial adaptation

Since the image foundation models have been trained on large amounts of web datasets, we can intuitively speculate that they might be powerful to encode the spatial context even in videos. In order to make the outstanding ability of spatial modeling to be more suitable for video understanding with a slight parameter tuning, we adopt two parallel adapters for $\texttt{MHA}$ and $\texttt{MLP}$ in each transformer block. The parallel adapters allow the model to learn the spatial context for action recognition from the appearance of target video data while maintaining the original contexts for object recognition.

Specifically, we sample T_S frames from a video and tokenize each frame. Similar to Eq. 1, the set of spatial tokens $\mathbf{X}_t^{\text{SP}}$ includes the learnable positional encodings $\mathbf{p}^{\text{SP}}$ and the spatial class token $\mathbf{x}_t^{\text{SP}}\{\texttt{[CLS]}\}$. The spatial adaptation in the l-th transformer block can be formulated by the following equations:

$$\begin{aligned} \mathbf{z}_l^{\text{SP}} &= \mathbf{h}_{l-1}^{\text{SP}} + \texttt{MHA}(\texttt{LN}(\mathbf{h}_{l-1}^{\text{SP}})) + \texttt{Adapter}(\texttt{LN}(\mathbf{h}_{l-1}^{\text{SP}})), \\ \mathbf{h}_l^{\text{SP}} &= \mathbf{z}_l^{\text{SP}} + \texttt{MLP}(\texttt{LN}(\mathbf{z}_l^{\text{SP}})) + \texttt{Adapter}(\texttt{LN}(\mathbf{z}_l^{\text{SP}})), \end{aligned} \tag{7}$$

where $\mathbf{h}_0^{\text{SP}} = [\mathbf{x}_t^{\text{SP}}\{\texttt{[CLS]}\}, \mathbf{X}_t^{\text{SP}}] + \mathbf{p}^{\text{SP}}$. We average the set of the spatial `[CLS]` tokens from the final transformer block to obtain a global spatial representation $\mathbf{y}^{\text{SP}}$, such that,

$$\mathbf{y}^{\text{SP}} = \frac{1}{T_S} \sum_{t=1}^{T_S} \mathbf{x}_t^{\text{SP}}\{\texttt{[CLS]}\}. \tag{8}$$

Recent methods have discussed that a high frame rate only increases the computation volume and is unnecessary to understand the semantics of appearance [9, 21]. We thus sparsely sample T_S frames with a low frame rate (*e.g.* 8 frames per clip).

4.2. Temporal adaptation

While spatial adaptation enables the models to take the spatial contexts in video data, the image models are still incapable of modeling the temporal dynamics. The key component that allows video transformers to model the solid temporal context is to learn relationships between local patches across *frames* in the video [1, 7]. To make image models capable of effectively establishing this component, we suggest a novel *grid-like frameset* transform technique that aggregates multiple frames into a unified *grid-like frameset*. Our grid-like frameset design is inspired by recent visual prompting research [4, 5]. It is simple yet surprisingly effective in imitating temporal modeling as spatial modeling and certainly reduces the computation. In each transformer block, we adopt two additional serial adapters for `MHA` and `MLP`, respectively.

More concretely, we sample T frames from a video and scale them with factors of w and h, such that the scaled frame size is $[W/w \times H/h \times 3]$. We stack $w \times h$ scaled frames according to temporal ordering and reshape a stacked frame to construct a set of frames in a grid form of the same size as the original frame (*i.e.*, $[W \times H \times 3]$). Note that the total number of grid-like framesets is $T_G = T/wh$. The set of temporal tokens $\mathbf{X}_g^{\text{TP}}$ for the g-th frameset is obtained in the same way in Eq. 1 and combined with the learnable temporal class token $\mathbf{x}_g^{\text{TP}}\{\texttt{[CLS]}\}$. Unlike the spatial adaptation, we use fixed 3D positional encodings [70], $\mathbf{p}^{\text{TP}}$, to the tokens to take the absolute temporal order and spatial positions of patches into account. The input transformation allows transformers to observe multiple frames at the same level. In experiments, we mainly construct a grid-like frameset from 16 original frames (*i.e.*, scaling factors $w = h = 4$) to take the computational efficiency and promising performance.

Whereas the parallel adapter is used in the spatial path, we sequentially append adapters to the top of `MHA` and `MLP` layers in each transformer block. Formally, temporal adaptation in the l-th block can be described as:

$$\begin{aligned} \mathbf{z}_l^{\text{TP}} &= \mathbf{h}_{l-1}^{\text{TP}} + \texttt{Adapter}(\texttt{MHA}(\texttt{LN}(\mathbf{h}_{l-1}^{\text{TP}}))), \\ \mathbf{h}_l^{\text{TP}} &= \mathbf{z}_l^{\text{TP}} + \texttt{Adapter}(\texttt{MLP}(\texttt{LN}(\mathbf{z}_l^{\text{TP}}))), \end{aligned} \tag{9}$$

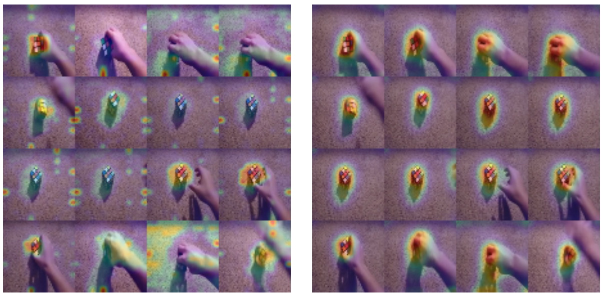

(a) DUALPATH w/o TA (b) DUALPATH w/ TA

Figure 4. Visualization of attention maps (a) without temporal adaptation (TA) and (b) with temporal adaptation for the action 'Spinning [something] that quickly stops spinning' in SSv2 [23].

where $\mathbf{h}_0^{\text{TP}} = [\mathbf{x}_g^{\text{TP}}\{\texttt{[CLS]}\}, \mathbf{X}_g^{\text{TP}}] + \mathbf{p}^{\text{TP}}$. Similar to spatial adaptation, a global temporal representation $\mathbf{y}^{\text{TP}}$ can be derived by averaging the set of the temporal `[CLS]` tokens from the final transformer block, *i.e.*,

$$\mathbf{y}^{\text{TP}} = \frac{1}{T_G} \sum_{g=1}^{T_G} \mathbf{x}_g^{\text{TP}}\{\texttt{[CLS]}\}. \tag{10}$$

For the final prediction, we concatenate the global spatial and temporal representations and feed them into the classifier with GeLU activation [29] between two FC layers.

4.3. Does a grid-like frameset really help to encode temporal context?

This work presents a new standpoint to perform video action recognition with the pretrained image transformer by transforming multiple frames into a unified grid-like frameset. However, it is still questionable whether the temporal path of DUALPATH can really capture the temporal context of videos. In this section, we provide some qualitative examples of the attention map. To validate the ability of precise temporal modeling, we sample videos from the SSv2 [23] dataset. Following [50], we depict the attention map of $\mathbf{x}_g^{\text{TP}}\{\texttt{[CLS]}\}$ from the final transformer block of the temporal path. As shown in Fig. 4, the model with the temporal adaptation (TA) of DUALPATH tends to concentrate on action-related regions, contrary to the model without TA that focuses on the irrelevant background. This result suggests that the temporal adaptation of DUALPATH strengthens the temporal modeling ability of the image model. More examples are shown in Fig. A1 of Appendix.

5. Experiment

5.1. Evaluation setup

Datasets. We evaluate the proposed method on four standard action recognition datasets, including the Kinetics-400 (K400) [38], HMDB51 [39], Something-something-v2 (SSv2) [23], and Diving-48 [42].

Method & Arch.	Pretrain	Model # Params	Trainable # Params	GFLOPs	R@1	R@5	Views
Full-tuning							
SlowFast+NL [21]	-	60M	60M	7020	79.8	93.9	16×3×10
MViT-B [18]	-	37M	37M	4095	81.2	95.1	64×3×3
UniFormer-B [40]	IN-1K	50M	50M	3108	83.0	95.4	32×4×3
TimeSformer-L [7]	IN-21K	121M	121M	7140	80.7	94.7	64×1×3
ViViT-L/16×2 FE [1]	IN-1K	311M	311M	3980	80.6	92.7	32×1×1
VideoSwin-L [47]	IN-21K	197M	197M	7248	83.1	95.9	32×4×3
MViTv2-L [43]	IN-21K	218M	218M	42420	86.1	97.0	32×3×5
MTV-L [76]	JFT	876M	876M	18050	84.3	96.3	32×4×3
TokenLearner-L/10 [57]	JFT	450M	450M	48912	85.4	96.3	64×4×3
ActionCLIP [68]	CLIP	142M	142M	16890	83.8	97.1	32×10×3
X-CLIP-L/14 [48]	CLIP	420M	420M	7890	87.1	97.6	8×4×3
Parameter Efficient Tuning							
EVL [45] w/ ViT-L/14	CLIP	368M	59M	8088	87.3	-	32×3×1
ST-Adapter [50] w/ ViT-B/16	CLIP	93M	7M	1821	82.7	96.2	32×3×1
DualPath w/ ViT-B/16	CLIP	96M	10M	710	85.4	97.1	32×3×1
DualPath w/ ViT-L/14	CLIP	330M	27M	1868	87.7	97.8	32×3×1

Table 1. Performance comparisons for action recognition on the Kinetics-400 [38] dataset. Note that Views = #frames × #clips × #spatial.

- **Kinetics-400 (K400)** contains about 240K training videos and 20K validation videos for 400 human action categories. Each video is trimmed to have a length of 10 seconds. While the K400 dataset provides a wide range of categories, they are known to be highly biased in spatial appearance [59].
- **Somthing-something-v2 (SSv2)** is a more challenging dataset since they require strong temporal modeling [59]. They contain about 168.9K training videos and 24.7K validation videos for 174 classes.
- **HMDB51** is a small dataset that provides about 7K videos of 51 action categories. We use all three splits, each split of which consists of 3570 and 1530 videos for training and evaluation, respectively. We report the average accuracy for three splits.
- **Diving-48** is a fine-grained diving action dataset. We use train-test split v2 that contains about 15K training videos and 2K validation videos of 48 diving actions. Since the action can not be determined by only the static representations (*e.g*. objects or background), stronger temporal modeling is required for this dataset.

Pretrained image backbone. We employ CLIP pretrained ViT-B/16 and ViT-L/14 as backbones. The results with Swin-B [46] are provided in Tab. A2 of the Appendix.

- **ViT-B/16** [17] consists of 12 transformer blocks with 86M parameters and takes patches of size 16 × 16 as inputs.
- **ViT-L/14** [17], a larger model than ViT-B/16, contains 24 transformer blocks with 303M parameters. It takes 14 × 14 patches as inputs.

Implementation details. For the K400, HMDB51, and Diving-48 datasets, we uniformly sample 8 frames (*i.e*., $T_s = 8$) with the sampling interval 8 in the spatial path. In the temporal path, we uniformly sample 16, 32, and 48 frames with the sampling intervals 4, 2, and 1 to construct 1, 2, and 3 grid-like framesets (*i.e*., $T_g = 1, 2, 3$), respectively. For the SSv2 dataset, we sample the same number of frames as in other datasets, but with a dynamic sampling interval to cover the whole video. Note that the frames for the spatial path are the subset of the temporal path frames. Please refer to more implementation details in Appendix A.

5.2. Comparison with state-of-the-art

In this section, we compare the proposed method with baselines [11, 35, 49, 50] in Sec. 3 and state-of-the-art video transformers [1, 7, 18, 40, 43, 45, 47, 48, 57, 68, 76] to demonstrate the effectiveness of the proposed method on four video action recognition datasets. Note that the number of frames of the spatial adaptation path of DualPath is fixed to 8 for all experiments, *i.e*., $T_S = 8$.

Results on Kinetics-400. We report the results evaluated on K400 [38] in Tab. 1. We first compare the proposed method with state-of-the-art video models that are pretrained on the large-scale image dataset and fully finetuned on K400. In terms of memory and computational efficiency, video models require a huge number of parameters (∼450M [57]) and computations (∼48912 GFLOPs [57]). On the other hand, we require only 10M trainable parameters which are newly stored, and 710 GFLOPs for inference using 32 frames with ViT-B/16 [17] backbone. Compared to X-CLIP-L/14 [48] which leverages the additional text branch, our DualPath achieves state-of-the-art performance with ViT-L/14 back-

Method & Arch.	Pretrain	Model #Params	Trainable #Params	GFLOPs	R@1	R@5	Views
Full-tuning							
Full-tuning ViT-B/16 [17]	CLIP	86M	86M	419	44.0	77.0	8×1×3
Full-tuning ViT-L/14 [17]	CLIP	303M	303M	1941	48.7	77.5	8×1×3
TimeSformer-L [7]	IN-21K	121M	121M	7140	62.4	-	64×1×3
MTV-B [76]	IN-21K	310M	310M	4790	67.6	90.4	32×4×3
MViT-B [18]	K400	37M	37M	510	67.1	90.8	32×1×3
MViTv2-B [43]	K400	51M	51M	675	70.5	92.7	40×1×3
ViViT-L/16×2 [1]	IN-21K/K400	311M	311M	11892	65.4	89.8	16×4×3
VideoSwin-B [47]	IN-21K/K400	89M	89M	963	69.6	92.7	32×1×1
Omnivore [22]	IN-21K/K400	-	-	-	71.4	93.5	32×1×3
MViTv2-B [43]	IN-21K/K400	213M	213M	8484	73.3	94.1	32×1×3
UniFormer-B [40]	IN-21K/K600	50M	50M	777	71.2	92.8	32×1×3
Parameter Efficient Tuning							
VideoPrompt* [37] w/ ViT-B/16	CLIP	92M	6M	537	31.0	60.3	8×1×3
VPT [35] w/ ViT-B/16	CLIP	92M	6M	537	36.2	61.1	8×1×3
AdaptFormer [11] w/ ViT-B/16	CLIP	94M	8M	544	51.3	70.6	8×1×3
Pro-tuning [49] w/ ViT-B/16	CLIP	95M	9M	538	50.8	69.9	8×1×3
EVL [45] w/ ViT-L/14	CLIP	484M	175M	9641	66.7	-	32×1×3
ST-Adapter [50] w/ ViT-B/16	CLIP	97M	11M	1955	69.5	92.6	32×3×1
DualPath w/ ViT-B/16	CLIP	99M	13M	642	69.6	92.5	16×1×3
DualPath w/ ViT-B/16	CLIP	99M	13M	716	70.3	92.9	32×1×3
DualPath w/ ViT-B/16	CLIP	99M	13M	791	71.2	93.2	48×1×3
DualPath w/ ViT-L/14	CLIP	336M	33M	1713	70.2	92.7	16×1×3
DualPath w/ ViT-L/14	CLIP	336M	33M	1932	71.4	93.4	32×1×3
DualPath w/ ViT-L/14	CLIP	336M	33M	2151	72.2	93.7	48×1×3

Table 2. Performance comparisons for action recognition on the SSv2 [23] dataset. Note that we reproduce VideoPrompt [37] without the additional text branch for fair comparison (denoting with $*$).

bone. The comparisons with parameter-efficient tuning methods [45,50] show that our DualPath achieves higher performance while requiring much lower burdens in computations under the same conditions.

Results on Something-Something-v2. We present the performance comparisons on SSv2 [23] in Tab. 2. The results show that our DualPath with ViT-B/16 achieves comparable or better performance than the prior supervised video models [1, 7, 18, 76], requiring a much smaller number of trainable parameters and GFLOPs. Our DualPath with ViT-L/14 shows more competitive performance, outperforming most prior works [22, 40, 43]. The baselines [11, 35, 37, 49], which have relatively weak temporal modeling ability, show significantly poor performance, implying that strong temporal modeling is mandatory for the SSv2 dataset. The comparisons to the CLIP pretrained PET approaches [45,50] with ViT-B/16 demonstrate the effectiveness and efficiency of DualPath, achieving higher performance (70.3 vs 69.5 [50]) with significantly low computations (716 vs 9641 [45] GFLOPs) using 32 frames. Thanks to the extreme computational efficiency, our DualPath comprises more competitive performance using 48 frames ($T_G = 3$) with low computation requirements.

Method & Arch.	Classifier	Params	HMDB51
Full-tuning w/ ViT-B/16 [17]	Lin.	86M	59.4
Linear w/ ViT-B/16	Lin.	0.1M	61.2
VPT [35] w/ ViT-B/16	Trans.	7M	62.4
AdaptFormer [11] w/ ViT-B/16	Trans.	8M	63.7
Pro-tuning [49] w/ ViT-B/16	Trans.	9M	63.3
VideoPrompt [37] w/ ViT-B/16	Trans.	6M	66.4
ST-Adapter* [50] w/ ViT-B/16	Lin.	7M	65.9
DualPath w/ ViT-B/16	MLPs.	10M	75.6

Table 3. Performance comparisons for action recognition on the HMDB51 [39] dataset with the CLIP pretrained ViT-B/16 [53]. We report the type of classifier and the number of learnable parameters for baselines and ours. **Lin.** and **Trans.** denote the linear classifier and temporal transformer, respectively. Our DualPath uses two MLP layers as the classifier. Note that we reproduce ST-Adapter [50] for fair comparison in terms of the pretrained dataset (denoting with $*$).

Results on HMDB51. In Tab. 3, we compare the results with baselines [11, 35, 37, 49, 50] on HMDB51 [39] that dominantly requires strong spatial modeling for action recognition. Surprisingly, our DualPath significantly outperforms baselines by large margins. This result demonstrates DualPath fully capitalizes on the exceptional spatial modeling ability of the pretrained image model for ac-

Method & Arch.	Pretrain	Params	Diving48
Supervised			
TimeSformer-L [7]	IN-21K	121M	81.0
VideoSwin-B [47]	IN-21K	88M	81.9
SIFAR-B-14 [19]	IN-21K	87M	87.3
ORViT [30]	IN-21K	160M	88.0
Parameter Efficient Tuning			
DUALPATH w/ ViT-B/16	CLIP	10M	88.7

Table 4. Performance comparisons for action recognition on the Diving-48 [42] dataset. We report the pretrained dataset, the number of learnable parameters (M) for each method, and the accuracy.

tion recognition. The comparison with VideoPrompt [37] that uses the additional text branch demonstrates the effectiveness of DUALPATH, improving 9.2% performance improvement.

Results on Diving-48. Tab. 4 shows performance comparisons on Diving-48 [42] that requires fine-grained action recognition. Our DUALPATH consistently outperforms all video models with only 10M trainable parameters. Particularly, we obtain a better performance than ORViT [30] which utilizes the additional tracking model. It indicates the utility of DUALPATH in fine-grained action recognition and the superiority of temporal modeling of DUALPATH.

5.3. Components analysis

Impact of dual-path. In the top panel of Tab. 5, we train the model by ablating each path and evaluate the performance on SSv2. Without the temporal path (DUALPATH w/o TA), the performance is significantly degraded despite using a larger number of frames ($T_S = 16$ vs 8). Without the spatial path (DUALPATH w/o SA), we can obtain slightly better performance than the result without the temporal adaptation. Since the SSv2 requires strong temporal modeling, we speculate that two ablation studies show comparison results. However, it still shows a substantial performance gap compared to the full model of DUALPATH, demonstrating the effectiveness of the dual-path design.

Frame rates in spatial path. In the middle panel of Tab. 5, we analyze the effect according to the number of frames used in the spatial path. The temporal path identically uses 16 frames to construct a grid-like frameset, and frames used in the spatial path are sampled from such 16 frames. A large number of T_S slightly improves performance, however, requires more computational costs. Considering the performance improvement compared to the computation increase, we mainly set T_S to 8.

Number of frames in grid. We scale down original frames with scaling factors w and h to construct grid-like framesets. These factors thus determine the number of frames the model observes within one grid-like frameset. While a large value of factors increases the temporal resolution, the information of each frame is inevitably reduced. For example, the original frame is scaled down to the size of 28×28 with $w = h = 8$. Meanwhile, a small value of factors retains richer information from the original frame, however, makes the temporal resolution small. As shown in the bottom panel of Tab. 5, we attain the best performance with $w = h = 4$.

Method	Params	GFLOPs	SSv2	Views
Effectiveness of Each Path				
DUALPATH w/o TA	5M	1016	53.7	16×3×1
DUALPATH w/o SA	8M	134	55.1	16×3×1
Effectiveness of T_S				
$T_S = 8$	13M	642	69.3	16×3×1
$T_S = 12$	13M	896	69.6	16×3×1
$T_S = 16$	13M	1150	69.8	16×3×1
Effectiveness of scaling factors				
w=h=2 (T_G=16)	13M	1752	66.4	64×3×1
w=h=4 (T_G=4)	13M	864	71.8	64×3×1
w=h=8 (T_G=1)	13M	642	61.5	64×3×1
DUALPATH	13M	642	69.3	16×3×1

Table 5. Performance with respect to variants of the components.

6. Conclusion and Future Work

In this paper, we have introduced the novel image-to-video transfer learning approach, DUALPATH. By incorporating a dual-path design into image transformers, DUALPATH adapts image models to the video task (*i.e.*, action recognition) with a small number of trainable parameters. The spatial path adaptation strengthens the inherent spatial context modeling of the pretrained image transformers for video data. The temporal path adaptation transforms multiple frames into a unified grid-like frameset, enabling the image model to capture relationships between frames. We appropriately employ the bottlenecked adapters in each path to adapt the pretrained features to target video data. In addition, we present several baselines transforming recent PETL approaches [11, 35, 49] for image-to-video adaptation. Experimental results demonstrated the superiority of the dual-path design and the grid-like frameset prompting, outperforming several baselines and supervised video models.

There are many possible directions for future work, encompassing cross-domain transfer learning. While we have explored image-to-video transfer learning, large foundation vision-language models are available. It would also be interesting to expand the superior pretrained 2D knowledge to 3D spatial modeling [71]. We hope our study will foster research and provide a foundation for cross-domain transfer learning.

Acknowledgement. This research was supported by the Yonsei Signature Research Cluster Program of 2022 (2022-22-0002) and the KIST Institutional Program (Project No.2E32283-23-064).

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 2, 5, 6, 7

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

[3] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint: arXiv:2203.17274*, 2022. 2

[4] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. *arXiv preprint: arXiv:2203.17274*, 2022. 5

[5] Amir Bar, Yossi Gandelsman, Trevor Darrell, Amir Globerson, and Alexei A. Efros. Visual prompting via image inpainting. *arXiv preprint: arXiv:2209.00647*, 2022. 5

[6] Elad Ben-Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *ACL*, 2022. 2

[7] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021. 2, 5, 6, 7, 8

[8] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *NeurIPS*, 2020. 1

[9] Shyamal Buch, Cristóbal Eyzaguirre, Adrien Gaidon, Jiajun Wu, Li Fei-Fei, and Juan Carlos Niebles. Revisiting the "videos" in video-language understanding. In *CVPR*, 2022. 5

[10] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 2, 4

[11] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. In *NeurIPS*, 2022. 2, 3, 4, 6, 7, 8

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2

[13] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 2

[14] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 4

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2

[16] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022. 1

[17] 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. In *ICML*, 2021. 1, 2, 6, 7

[18] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021. 2, 6, 7

[19] Quanfu Fan, Chun-Fu (Richard) Chen, and Rameswar Panda. Can an image classifier suffice for action recognition? In *ICLR*, 2022. 8

[20] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. *arXiv preprint: arXiv:2205.09113*, 2022. 2

[21] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2, 4, 5, 6

[22] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *CVPR*, 2022. 2, 7

[23] 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. In *ICCV*, 2017. 2, 5, 7

[24] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 2

[25] Demi Guo, Alexander M. Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. In *ACL*, 2021. 1, 2

[26] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. In *ICLR*, 2022. 1, 2, 3

[27] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2

[28] Kaining He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2

[29] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint: arXiv:1606.08415*, 2016. 5

[30] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *CVPR*, 2022. 8

[31] Neil Houlsby, Andrei Giurgiu, Stanisław Jastrzębski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019. 1, 2, 3

[32] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 2

[33] Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010. 1

[34] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yunhsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 1, 2

[35] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie1, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*, 2022. 2, 3, 4, 6, 7, 8

[36] Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint. arXiv:2207.07039*, 2022. 2

[37] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *ECCV*, 2022. 2, 3, 7, 8

[38] 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. *arXiv preprint: arXiv:1705.06950*, 2017. 1, 2, 5, 6

[39] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, 2011. 2, 5, 7

[40] Kunchang Li, YaliWang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *ICLR*, 2021. 2, 6, 7

[41] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *ACL*, 2021. 1, 2

[42] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, 2018. 2, 5, 8

[43] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 2, 6, 7

[44] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2

[45] Ziyi Lin, Shijie Geng, Renrui Zhang, Peng Gao, Gerard de Melo, Xiaogang Wang, Jifeng Dai, Yu Qiao, and Hongsheng Li. Frozen clip models are efficient video learners. In *ECCV*, 2022. 2, 6, 7

[46] 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 *ICCV*, 2021. 1, 6

[47] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, 2022. 2, 6, 7, 8

[48] Bolin Ni, Houwen Peng, Minghao Chen, and Songyang Zhang. Expanding language-image pretrained models for general video recognition. In *ECCV*, 2022. 2, 3, 6

[49] Xing Nie, Bolin Ni, Jianlong Chang, Gaomeng Meng, Chunlei Huo, Zhaoxiang Zhang, Shiming Xiang, Qi Tian, and Chunhong Pan. Pro-tuning: Unified prompt tuning for vision tasks. *arXiv preprint: arXiv:2207.14381*, 2022. 2, 3, 4, 6, 7, 8

[50] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning for action recognition. In *NeurIPS*, 2022. 2, 3, 4, 5, 6, 7

[51] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: contrastive video representation learning with temporally adversarial examples. In *CVPR*, 2021. 2

[52] Jungin Park, Jiyoung Lee, Ig-Jae Kim, and Kwanghoon Sohn. Probabilistic representations for video contrastive learning. In *CVPR*, 2022. 2

[53] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICLR*, 2021. 1, 2, 7

[54] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. Technical report, OpenAI, 2018. 1

[55] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. 1

[56] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Research*, 2020. 1

[57] Michael S Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. In *NeurIPS*, 2021. 2, 6

[58] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In *NeurIPS*, 2022. 1

[59] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *WACV*, 2021. 6

[60] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 4

[61] Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu,

Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. On transferability of prompt tuning for natural language processing. In *NAACL*, 2022. 1, 2

[62] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 2

[63] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 1, 2

[64] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-language tasks. In *CVPR*, 2022. 2

[65] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 2

[66] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1, 2

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3

[68] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint: arXiv:2109.08472*, 2021. 2, 6

[69] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint: arXiv:2208.10442*, 2022. 1

[70] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 5

[71] Ziyi Wang, XuminYu, YongmingRao, and Jie Zhou JiwenLu. P2p: Tuning pre-trained image models for point cloud analysis with point-to-pixel prompting. In *NeurIPS*, 2022. 8

[72] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, 2021. 1

[73] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 1, 2

[74] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022. 2

[75] Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. Videoclip: Contrastive pre-training for zero-shot video-text understanding. In *EMNLP*, 2021. 2

[76] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022. 2, 6, 7

[77] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv preprint: arXiv:2111.11432*, 2021. 1

[78] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021. 1

[79] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022. 1, 2

[80] Yanyi Zhang, Xinyu Li, Bing Shuai Chunhui Liu, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *ICCV*, 2021. 2