D^2ST-Adapter: Disentangled-and-Deformable Spatio-Temporal Adapter for Few-shot Action Recognition

Wenjie Pei[1,⋆], Qizhong Tan[1,⋆], Guangming Lu[1], and Jiandong Tian[2]

[1] Harbin Institute of Technology, Shenzhen
[2] Shenyang Institute of Automation, Chinese Academy of Sciences
wenjiecoder@outlook.com, 200110929@stu.hit.edu.cn, luguangm@hit.edu.cn, tianjd@sia.cn

Abstract. Adapting large pre-trained image models to few-shot action recognition has proven to be an effective and efficient strategy for learning robust feature extractors, which is essential for few-shot learning. Typical fine-tuning based adaptation paradigm is prone to overfitting in the few-shot learning scenarios and offers little modeling flexibility for learning temporal features in video data. In this work we present the Disentangled-and-Deformable Spatio-Temporal Adapter (D^2ST-Adapter), which is a novel adapter tuning framework well-suited for few-shot action recognition due to lightweight design and low parameter-learning overhead. It is designed in a dual-pathway architecture to encode spatial and temporal features in a disentangled manner. In particular, we devise the anisotropic Deformable Spatio-Temporal Attention module as the core component of D^2ST-Adapter, which can be tailored with anisotropic sampling densities along spatial and temporal domains to learn spatial and temporal features specifically in corresponding pathways, allowing our D^2ST-Adapter to encode features in a global view in 3D spatio-temporal space while maintaining a lightweight design. Extensive experiments with instantiations of our method on both pre-trained ResNet and ViT demonstrate the superiority of our method over state-of-the-art methods for few-shot action recognition. Our method is particularly well-suited to challenging scenarios where temporal dynamics are critical for action recognition.

Keywords: Action recognition · Few-shot learning · Adapter tuning

1 Introduction

Few-shot action recognition aims to learn an action recognition model from a set of base classes of video data, which has the capability to recognize novel

[⋆] Equal contribution.
Code is available at `https://github.com/qizhongtan/D2ST-Adapter`.

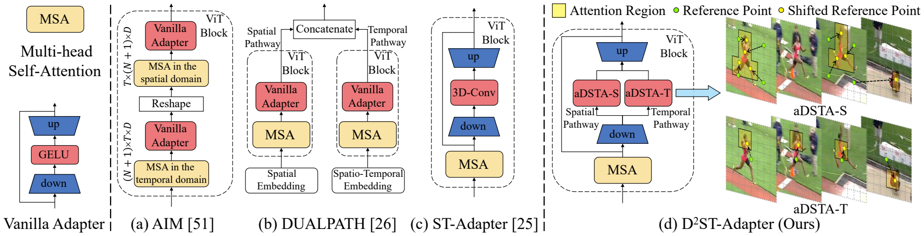


Fig. 1: Comparison with previous adapter-based action recognition methods. Both AIM [51] and DUALPATH [26] pre-learn the spatial and temporal features separately by duplicating the MSA modules, and insert vanilla adapter into each duplicated module for specific feature adaptation. ST-Adapter [25] first attempts to design the specialized adapter for video data using 3D convolution. Unlike these models, our D^2*ST-Adapter* is designed in a dual-pathway architecture to encode spatial and temporal features in a disentangled manner. Furthermore, we design the anisotropic Deformable Spatio-Temporal Attention (aDSTA) as the essential component, which configures different sampling density in the spatial and temporal domains to model two pathways specifically and enables D^2*ST-Adapter* to perform feature adaptation in a global view in 3D spatio-temporal space while maintaining a lightweight design.

categories of actions using only a few support samples. To this end, learning an effective feature extractor that is generalizable across different classes is crucial to the task. A typical way [1, 2, 42, 44, 54] is to leverage pre-trained large vision models [13, 29, 37] for feature learning by task adaptation.

Most existing methods [28, 42, 43] seek to adapt large vision models to few-shot action recognition by fine-tuning either the entire or partial model. While such methods have achieved promising performance, there are two important limitations that hamper them from exploiting the full potential of the pre-trained models. First, fully fine-tuning the pre-trained models is prone to overfitting in the few-shot learning scenarios whilst partial fine-tuning acquires limited effectiveness due to limited capacity for adaptation. Second, most large vision models are trained on image data due to the difficulty of collecting a large scale of video data. Thus, these fine-tuning based methods require to learn the temporal features as a post-process after feature learning by the pre-trained backbone, typically by constructing an auxiliary module to model the temporal dynamics [36,43,44,46], which has limited effectiveness compared to the learning manner throughout the whole feature encoding stage.

As a prominent parameter-efficient fine-tuning technique [12,14,17,58], adapter tuning [4,14,41] embeds learnable adapters with a negligible amount of parameters into pre-trained models flexibly while keeping the model frozen, conducting model adaptation efficiently and effectively. Compared to the fine-tuning adaptation paradigm, adapter tuning is particularly better-suited to few-shot learning scenarios. A straightforward way to apply adapter tuning to image-to-video transfer learning is to employ the vanilla adapter for image data directly to adapt the pre-learned spatial features and temporal features separately, which is exactly adopted by AIM [51] and DUALPATH [26]. As shown in Figure 1, both AIM and DUALPATH first pre-learn the spatial and temporal features separately by duplicating the multi-head self-attention modules either in cascaded (AIM) or parallel (DUALPATH) manners, and insert vanilla adapter in each duplicated module for specific feature adaptation. However, such methods inevitably incur heavyweight model architecture and massive computation overhead. ST-Adapter [25] first attempts to design the specialized adapter for video data by inserting a 3D convolution layer into the vanilla adapter for learning spatio-temporal features while maintaining a lightweight structure. Nevertheless, we investigate two potential limitations of ST-Adapter. First, it learns the spatial and temporal features jointly by 3D convolution, whereas it has been shown that in the low-data scenarios, such a joint learning paradigm of spatio-temporal features for video data is inferior to learning the spatial and temporal features in a disentangled way, such as SlowFast [8] or two-stream design [31,40]. Second, the convolutional operation, especially with shallow layers for lightweight design, has a limited local receptive field in both the spatial and temporal domains, which also limits the performance of ST-Adapter.

In this work we design the Disentangled-and-Deformable Spatio-Temporal Adapter (*D^2ST-Adapter*) to tackle the aforementioned limitations. Specifically, we design our *D^2ST-Adapter* as a dual-pathway architecture, as illustrated in Figure 1, in which the spatial pathway is responsible for capturing the spatial appearance features while the temporal pathway focuses on learning the temporal dynamics. Thus, our model can encode the spatio-temporal features in a disentangled manner. Moreover, we devise the anisotropic Deformable Spatio-Temporal Attention (aDSTA) module as an essential component to model both the spatial and temporal pathways. It adapts the deformable attention [48] from 2D image space to 3D spatio-temporal space, which first samples a group of reference points in the spatio-temporal space and then learns the offset for each point to shift them to more informative regions in the 3D space. These shifted points are used as key and value pairs for aDSTA to perform sparse self-attention, which allows our *D^2ST-Adapter* to perform feature adaptation in a global view while maintaining a lightweight design.

A novel design of our aDSTA is that we make the sampling density anisotropic along the spatial and temporal domains by specific configuration, which allows for tailoring specialized versions of aDSTA to model the spatial and temporal pathways separately. In particular, we tailor aDSTA-T with denser sampling along the temporal domain than the spatial domain for the temporal pathway since it focuses on capturing temporal features. In contrast, the tailored aDSTA-S for the spatial pathway samples denser reference points along the spatial domain. To conclude, we make the following contributions:

- We propose *D^2ST-Adapter*, a novel adapter tuning method for few-shot action recognition, which is designed in a dual-pathway architecture to adaptively encode the spatial and temporal features in a disentangled manner. It can adapt large vision models pre-trained on image data to video data efficiently and effectively in few-shot learning scenarios due to the lightweight design.
- We devise the anisotropic Deformable Spatio-Temporal Attention (aDSTA), which can be tailored with anisotropic sampling densities along spatial and

temporal domains to model the spatial and temporal pathways separately, allowing our *D^2ST-Adapter* to encode features in a global view while maintaining a lightweight design.
- Extensive experiments on five benchmarks with instantiations of our method on pre-trained ResNet [13] and ViT [29], demonstrate the superiority of our method over other methods, particularly in challenging scenarios like SSv2 benchmark where the temporal dynamics are critical for action recognition.

2 Related Work

Few-shot image classification. Existing works for few-shot image classification can be categorized into three groups: augmentation-based, optimization-based, and metric-based. Augmentation-based methods mainly hallucinate [11, 45] and generate [20,55] new training samples adversarially to prevent the model from over-fitting. Optimization-based algorithms [9,16,30] aim to train a model that generalizes well and can be quickly adapted to new tasks through a few gradient update steps. Metric-based approaches recognize novel samples through nearest neighbor classification using a specific distance metric, such as cosine similarity [39,52], Euclidean distance [6,32,53], and learnable metrics [34]. Our work follows the metric-based category due to its simplicity and effectiveness, and we focus on the more challenging video setting.

Few-shot action recognition. Most existing few-shot action recognition methods adopt the metric-based paradigm to classify videos, and primarily focus on two directions to deal with this task. The first one is to investigate the spatio-temporal modeling [27,36,43,46,47,50,54,56,59,60]. For example, MTFAN [46] proposes a motion encoder to learn global motion patterns and injects them into each video representations using a motion modulator. STRM [36] enriches local patch features and global frame features for joint spatio-temporal modeling. MoLo [43] designs a motion autodecoder to explicitly extract motion dynamics in a unified network. Our method also aims to model spatio-temporal features effectively and efficiently based on adapter-tuning technique. Another direction focuses on designing effective metric learning strategy [1,2,15,24,28,44,46,57]. OTAM [2] utilizes the DTW [23] algorithm to calculate video distances with strict temporal alignment. TRX [28] exhaustively enumerates all sub-sequences of support and query videos and matches them using an attention mechanism. HyRSM [44] applies a novel bidirectional mean Hausdorff metric (referred to as Bi-MHM) to alleviate the strictly ordered constraints. We evaluate our *D^2ST-Adapter* using the above three classical matching strategies to demonstrate its effectiveness and robustness.

Adapter tuning. As a classical parameter-efficient fine-tuning method, adapter tuning is first proposed in [14], and quickly draws attention in many other research areas [4,12,35,41]. A common practice is to build up a lightweight module (named *Adapter*) which only consisting of negligible learnable parameters, and selectively plug it into a pre-trained model. During training, only the parameters of inserted adapters are updated while the original model remains frozen, leading to efficient task adaptation. Recently, some works apply this method to adapt image models for action recognition. AIM [51] duplicates multi-head self-attention modules and plugs adapters after them to separately learn the spatial and temporal features in a cascaded manner. Similarly, DUALPATH [26] explicitly builds the two-stream architecture upon ViT and utilizes adapters to parallelly learn the spatial and temporal features. ST-Adapter [25] employs depth-wise 3D convolution to construct adapter for feature adaptation, which endows it with spatio-temporal modeling capability. Our *D^2ST-Adapter* follows the basic framework of ST-Adapter, and meanwhile optimize the essential technical designs for few-shot action recognition.

3 Method

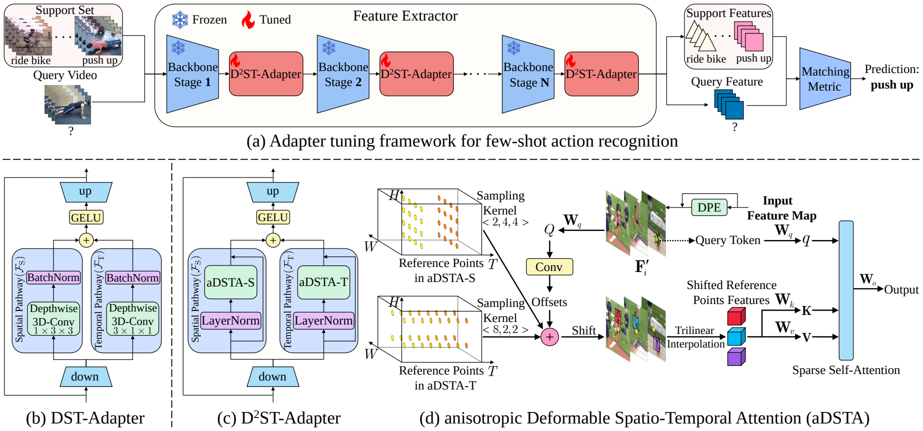
3.1 Overview

Problem Formulation of Few-shot Action Recognition. The aim of few-shot action recognition is to learn an action recognizer from a set of base classes, which is able to recognize novel classes using only a few labeled samples. The task typically follows episodic paradigm [2, 39, 43, 44], which is consistent with general application scenarios for few-shot learning. An episode includes a support set $\mathcal{S}$ consisting of N classes and K labeled samples for each class (referred to as the N-way K-shot task), as well as a query set $\mathcal{Q}$ that contains unlabeled samples to be classified. In each episode, we aim to classify every query into one of the N classes with the guidance of the support set. Such episodic task setting is consistently adopted during all the training, validation, and test stages.

Adapter Tuning Framework. We design a plug-and-play and lightweight adapter, dubbed *D^2ST-Adapter*, which can be integrated into most existing large vision models. Thus we can leverage the powerful feature encoding capability of pre-trained large vision models by efficient image-to-video model adaptation while incurring only a small amount of parameter-tuning overhead. It is particularly well-suited to few-shot learning scenarios.

Figure 2 (a) illustrates the overall adapter tuning framework of our method. Given a pre-trained feature extractor from a large model, our designed lightweight *D^2ST-Adapter* can be selectively plugged into the middle layers of the feature extractor to perform feature transformation for task adaptation while maintaining the shape of feature maps consistent before and after each adaptation. Only the inserted adapters are tuned during training while the pre-trained backbone keeps frozen, leading to efficient task adaptation. The learned features by such adapter tuning framework are further used for few-shot action recognition based on the metric-based strategy [2, 32, 43].

Our *D^2ST-Adapter* is designed independently from the backbone structure of the feature extractor, thus it can be readily integrated into most of classical pre-trained deep vision models. In particular, we instantiate the backbone of feature extractor with two dominant deep visual learning frameworks, namely ResNet [13] and ViT [7], respectively.



(a) Adapter tuning framework for few-shot action recognition

(b) DST-Adapter

(c) D²ST-Adapter

(d) anisotropic Deformable Spatio-Temporal Attention (aDSTA)

Fig. 2: (a) Overall adapter tuning framework of our method. (b) DST-Adapter, a convolutional version of our model, which also follows dual-pathway architecture, whereas both pathways are constructed based on 3D convolution. (c) D^{2}ST-Adapter, which is designed in a dual-pathway architecture and both pathways are modeled based on the proposed aDSTA illustrated in (d).

3.2 D^{2}ST-Adapter

Dual-pathway Adapter Architecture for Spatio-Temporal Disentanglement. Successful action recognition from video data entails effective feature learning of both spatial semantic and temporal dynamic features. Most large vision models are typically pre-trained on image data, thus our *D^{2}ST-Adapter* should be capable of capturing both the spatial features and temporal features. To this end, we design the *D^{2}ST-Adapter* as a dual-pathway architecture shown in Figure 2 (c), in which the spatial pathway is responsible for capturing the spatial semantics while the temporal pathway focuses on learning the temporal dynamics. As a result, our model is able to encode the spatio-temporal features for video data in a disentangled manner.

As a common practice of typical Adapters [14,25], our *D^{2}ST-Adapter* adopts the bottleneck architecture for reducing the computational complexity. It first downsamples the feature map into a low-dimensional feature space, then the downsampled features are fed into the spatial and temporal pathways concurrently to perform disentangled feature adaptation. Finally, both the adapted spatial and temporal features are fused by simple element-wise addition and upsampled back to the initial size. Formally, given the feature maps $\mathbf{F}_i^{\text{in}} \in \mathbb{R}^{T\times H\times W\times C}$ obtained from the i-th stage of the pre-trained backbone, containing C channels of feature maps with spatial size $H \times W$ for each of T frames, the feature adaptation by *D^{2}ST-Adapter* can be formulated as:

$$\mathbf{F}_i^{\text{out}} = \text{GELU}\left(\mathcal{F}_{\text{S}}(\mathbf{F}_i^{\text{in}} \cdot \mathbf{W}_{\text{down}}) \oplus \mathcal{F}_{\text{T}}(\mathbf{F}_i^{\text{in}} \cdot \mathbf{W}_{\text{down}})\right) \cdot \mathbf{W}_{\text{up}}. \tag{1}$$

Herein, $\mathbf{W}_{\text{down}} \in \mathbb{R}^{C \times C'}$ and $\mathbf{W}_{\text{up}} \in \mathbb{R}^{C' \times C}$ are the transformation matrices of two linear layers for downsampling and upsampling, respectively. $\mathcal{F}_{\text{S}}$ and $\mathcal{F}_{\text{T}}$ denote the disentangled feature adaptation by the spatial and temporal pathways respectively. Below we discuss two feasible modeling mechanisms for $\mathcal{F}_{\text{S}}$ and $\mathcal{F}_{\text{T}}$, which leverages 3D convolution and our proposed Anisotropic Deformable Spatio-Temporal Attention, respectively.

Modeling with 3D Convolution. A straightforward way to model the disentangled spatio-temporal feature adaptation ($\mathcal{F}_{\text{S}}$ and $\mathcal{F}_{\text{T}}$ in Equation 1) is to employ 3D convolutional network, which can be tailored by configuring the shape of convolutional kernels to focus on learning either the spatial or the temporal features. Specifically, we model both $\mathcal{F}_{\text{S}}$ and $\mathcal{F}_{\text{T}}$ using 3D depth-wise convolutional layers followed by a batch normalization layer and a GELU layer, as shown in Figure 2 (b). The only difference between the modeling of them is that $\mathcal{F}_{\text{S}}$ uses $1 \times 3 \times 3$ convolutional kernel for capturing spatial features while $\mathcal{F}_{\text{T}}$ is constructed with $3 \times 1 \times 1$ convolutional layer to learn the temporal dynamics. The resulting version is called *DST-Adapter*.

While it has been validated that leveraging 3D convolutions with different kernel shapes can learn the spatial and temporal features separately [8,38], An important limitation of using 3D convolutions to construct *D^{2}ST-Adapter*, which is also suffered by ST-Adapter, is that the convolutional operation has limited local receptive field either in the spatial or the temporal domain. Constructing deep convolutional network by stacking convolutional layers can enlarge the receptive field whilst increasing the model size and incurring heavy computational burden, which is inconsistent with the lightweight design principle for adapters.

Modeling with Anisotropic Deformable Spatio-Temporal Attention. To address the limitation of the modeling mechanism based on 3D convolution (DST-Adapter), we devise anisotropic Deformable Spatio-Temporal Attention (aDSTA) which performs feature adaptation in a global view by sparse self-attention while maintaining high computational efficiency. It adapts the deformable attention [48] from 2D image space to 3D spatio-temporal space. As shown in Figure 2 (d), aDSTA first samples a group of reference points with anisotropic sampling density in the spatio-temporal space and then learns the offset for each reference point to shift them to the more informative regions. As a result, aDSTA can learn a set of informative tokens which are further used as key and value pairs shared for all queries for sparse self-attention.

We conduct several adaptations from the deformable attention in 2D space to our aDSTA in spatio-temporal 3D space. First, we employ Dynamic Position Embedding (DPE) [21] implemented as a 3D depth-wise convolution layer to learn the semantic-conditioned spatio-temporal position information for each token and incorporate it into the semantic features by simple element-wise addition. Thus, the feature map $\mathbf{F}'_i \in \mathbb{R}^{T \times H \times W \times C'}$ fused with the position information at the i-th stage is obtained by:

$$\mathbf{F}'_i = \mathcal{F}_{\text{DPE}}(\mathbf{F}^{\text{in}}_i \cdot \mathbf{W}_{\text{down}}) \oplus (\mathbf{F}^{\text{in}}_i \cdot \mathbf{W}_{\text{down}}). \tag{2}$$

Our aDSTA learns the offsets in 3D space for the reference points by a 3D convolutional network consisting of a 3D depth-wise convolution layer and a $1 \times 1 \times 1$

3D convolution layer as well as a GELU in between. After shifting the reference points according to the learned offsets, we derive the features for each shifted reference point by performing trilinear interpolation among neighboring tokens within a 3D volume space around the point. For instance, the features for the shifted point $\mathbf{p}$ located at (p_t, p_h, p_w) is calculated via the trilinear interpolation $\mathcal{F}_{\text{tri-int}}$ by:

$$\mathcal{F}_{\text{tri-int}}(\mathbf{p}) = \sum_{\mathbf{r}} g(p_t, r_t) \cdot g(p_h, r_h) \cdot g(p_w, r_w) \cdot \mathbf{F}'_i(\mathbf{r}), \tag{3}$$

where $g(a, b) = \max(0, 1 - |a - b|)$ defines an interpolating cubic volume space centered at a and $\mathbf{r} = (r_t, r_h, r_w)$ indexes all tokens in the whole spatio-temporal 3D space. All shifted reference points are used as the keys and values shared for sparse self-attention. For instance, a token in $\mathbf{F}'_i$ located at (u_t, u_h, u_w) serves as a query and the output $\mathbf{Z}_{i,(u_t,u_h,u_w)}$ of aDSTA is calculated by:

$$\begin{aligned} &\mathbf{q} = \mathbf{F}'_{i,(u_t,u_h,u_w)} \cdot \mathbf{W}_q, \mathbf{K} = \mathbf{P} \cdot \mathbf{W}_k, \mathbf{V} = \mathbf{P} \cdot \mathbf{W}_v, \\ &\mathbf{Z}_{i,(u_t,u_h,u_w)} = \text{softmax}(\mathbf{qK}/\sqrt{C'})\mathbf{V} \cdot \mathbf{W}_o, \end{aligned} \tag{4}$$

where $\mathbf{P} \in \mathbb{R}^{M \times C'}$ is feature matrix of M shifted points and $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ are projection matrices for the query, key, value and output respectively.

Anisotropic Sampling Density. The learned offsets for reference points are restricted in a limited range for two reasons. First, it can prevent the potential training collapse that all reference points are shifted to the same point. Second, such restriction can ensure that each reference point can be shifted to a unique location within a local region. As a result, more sampling of reference points typically leads to more powerful and fine-grained representations by the shifted points. The deformable attention samples reference points uniformly in the 2D feature space with the same sampling density along two spatial dimensions. In contrast, we configure anisotropic sampling density in the spatial and temporal domains. Consequently, our *D^2ST-Adapter* can leverage aDSTA to model both spatial feature adaptation $\mathcal{F}_{\text{S}}$ in the spatial pathway and temporal feature adaptation $\mathcal{F}_{\text{T}}$ in the temporal pathway. Specifically, we define the '*Sampling kernel*' for aDSTA as:

Definition 1. ***Sampling kernel**: formulated as the sampling densities of reference points along the dimensions of time, height and width, respectively, denoted as $< n_t, n_s, n_s >$, s.t. $M = n_t \times n_s \times n_s$. M is the total number of sampled reference points.*

Intuitively, an aDSTA module whose sampling kernel has large n_t and small n_s are more capable of learning the temporal features than learning the spatial features. On the other hand, sampling more reference points in the spatial domain than the temporal domain ($n_s > n_t$) generally makes aDSTA focus on learning the spatial features. Thus, we can tailor aDSTA by configuring the sampling kernel to model $\mathcal{F}_{\text{S}}$ and $\mathcal{F}_{\text{T}}$ correspondingly. In particular, we tailor two versions of aDSTA:

- **aDSTA-S** for modeling $\mathcal{F}_\text{S}$ in the spatial pathway, whose sampling kernel has relatively large n_s and small n_t ($n_s > n_t$).
- **aDSTA-T** for modeling $\mathcal{F}_\text{T}$ in the temporal pathway, which samples denser reference points in temporal domain than in spatial domain ($n_t > n_s$).

In practice, the values of n_t and n_s are tuned as hyper-parameters. Such modeling is analogous to classical property of 3D convolutional network that configuring the shape of the convolutional kernel can steer the convolutional operation to focus on learning features in different domain.

3.3 End-to-End Adapter Tuning

The learned features by our adapter tuning framework are further used for few-shot action recognition based on the metric-based strategy. It first calculates the frame-wise L_2 distance matrix between the input query video and each class prototype derived by simply averaging the corresponding support samples. Then the distance matrix is used to calculate the matching similarity between the query and each class for prediction. Formally, the similarity between query q and class prototype c can be expressed as:

$$s(\mathbf{F}_q, \mathbf{F}_c) = \mathbf{M}([\mathbf{F}_q^1, \cdots, \mathbf{F}_q^T], [\mathbf{F}_c^1, \cdots, \mathbf{F}_c^T]), \tag{5}$$

where $\mathbf{F}_q^i$ and $\mathbf{F}_c^j$ denote the features of the i-th frame in query q and the j-th frame in class prototype c while $\mathbf{M}$ is the matching metric such as OTAM [2] and Bi-MHM [44]. The whole model can be optimized in an end-to-end manner using the cross-entropy loss.

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on five standard few-shot action recognition benchmarks, including SSv2-Full [10], SSv2-Small [10], Kinetics [3], HMDB51 [19], and UCF101 [33]. The temporal features are crucial to the action recognition in SSv2-Full and SSv2-Small while the actions in other datasets relies more on the spatial features than the dynamic features. Following the typical data split [2, 43, 59], For SSv2-Full, SSv2-Small and Kinetics, we select 64/24/12 classes from the datasets as the training/validation/test set, respectively. As for HMDB51 and UCF101, we adopt the same data split as [43, 54].

Pre-trained backbone. Most prior arts employs the ResNet-50 [13] pre-trained on ImageNet [5] as backbone to extract frame features. Recently, CLIP-FSAR [42] conduct experiments on CLIP-ViT-B [7] and get remarkable performance. To verify the effectiveness of our *D^{2}ST-Adapter*, we instantiate our method with both pre-trained backbones.

Implementation details. We insert one *D^{2}ST-Adapter* module in each stage of the pre-trained model (4 stages for ResNet-50 and 12 stages for CLIP-ViT-B) as shown in Figure 2 (a). We tune the sampling kernel for aDSTA-S and

Table 1: Classification accuracy (%) of different methods on SSv2-Full and SSv2-Small that are challenging benchmarks where both the temporal and spatial features are crucial to action recognition. The highest and the second best results of both backbones are highlighted in **bold** and underline, respectively.

Method	Backbone	Adaptation paradigm	SSv2-Full					SSv2-Small		
			1-shot	2-shot	3-shot	4-shot	5-shot	1-shot	3-shot	5-shot
CMN [59]	ResNet-50	Full fine-tuning	34.4	–	–	–	43.8	33.4	42.5	46.5
CMN-J [60]	ResNet-50	Full fine-tuning	–	–	–	–	–	36.2	44.6	48.8
OTAM [2]	ResNet-50	Full fine-tuning	42.8	49.1	51.5	52.0	52.3	36.4	45.9	48.0
ITANet [56]	ResNet-50	Full fine-tuning	49.2	55.5	59.1	61.0	62.3	39.8	49.4	53.7
TRX [28]	ResNet-50	Full fine-tuning	42.0	53.1	57.6	61.1	64.6	36.0	51.9	59.1
TA^{2}N [22]	ResNet-50	Full fine-tuning	47.6	–	–	–	61.0	–	–	–
MTFAN [46]	ResNet-50	Full fine-tuning	45.7	–	–	–	60.4	–	–	–
STRM [36]	ResNet-50	Full fine-tuning	43.1	53.3	59.1	61.7	68.1	37.1	49.2	55.3
HyRSM [44]	ResNet-50	Full fine-tuning	54.3	62.2	65.1	67.9	69.0	40.6	52.3	56.1
Nguyen *et al.* [24]	ResNet-50	Full fine-tuning	43.8	–	–	–	61.1	–	–	–
Huang *et al.* [15]	ResNet-50	Full fine-tuning	49.3	–	–	–	66.7	38.9	–	**61.6**
HCL [57]	ResNet-50	Full fine-tuning	47.3	54.5	59.0	62.4	64.9	38.7	49.1	55.4
SloshNet [49]	ResNet-50	Full fine-tuning	46.5	–	–	–	68.3	–	–	–
GgHM [50]	ResNet-50	Full fine-tuning	54.5	–	–	–	69.2	–	–	–
MoLo [43]	ResNet-50	Full fine-tuning	56.6	62.3	67.0	68.5	70.6	42.7	52.9	56.4
ST-Adapter [25]	ResNet-50	Adapter tuning	52.2	59.9	64.1	67.1	68.7	41.9	51.5	55.7
D^{2}ST-Adapter (Ours)	ResNet-50	Adapter tuning	**57.0**	**65.2**	**69.5**	**71.4**	**73.6**	**45.8**	**56.6**	60.9
CLIP-FSAR [42]	CLIP-ViT-B	Full fine-tuning	61.9	64.9	68.1	70.9	72.1	54.5	58.6	61.8
AIM [51]	CLIP-ViT-B	Adapter tuning	63.7	72.0	75.2	77.3	79.2	52.8	64.0	67.5
DUALPATH [26]	CLIP-ViT-B	Adapter tuning	64.5	72.6	76.0	78.0	79.8	53.5	64.4	68.1
ST-Adapter [25]	CLIP-ViT-B	Adapter tuning	64.2	72.4	76.0	77.4	79.5	53.1	64.2	68.0
D^{2}ST-Adapter (Ours)	CLIP-ViT-B	Adapter tuning	**66.7**	**75.3**	**78.3**	**80.1**	**81.9**	**55.0**	**65.8**	**69.3**

aDSTA-T for each backbone and the details are provided in the supplementary material. For a fair comparison with previous methods [2, 43, 44], we uniformly sample 8 frames (*i.e.*, $T = 8$) as the input of a video. We employ several standard techniques for data augmentation such as random crop and color jitter in the training stage. Bi-MHM [44] is used as the matching metric in comparison with other methods. Besides, OTAM [2] and TRX [28] are additionally used in the ablation study. We train the model episodically with Adam [18] optimizer. Following the common practice [43, 44], in the test stage, we randomly select 10,000 episodes from the test set and report the average accuracy.

4.2 Comparison with State-of-the-Art Methods

Benchmarks sensitive to temporal features. Table 1 shows the comparative results of different methods on SSv2-Full and SSv2-Small, which are quite challenging since both the temporal and spatial features are crucial to the action recognition. Our *D^{2}ST-Adapter* achieves the best performance and outperforms other methods by a large margin with both backbones in all few-shot settings except '5-shot' on SSv2-Small. These results demonstrate the advantages of our method over other methods in dealing with challenging scenarios where the temporal features are critical for action recognition. In particular, our *D^{2}ST-Adapter* outperforms all existing adapter tuning methods for video

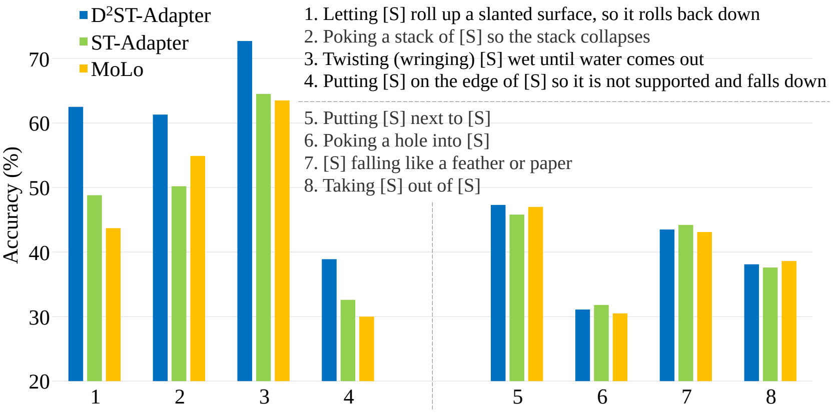


Fig. 3: Illustration of actions our method excels at (1-4) and performs poorly at (5-8), respectively. The performance is evaluated on SSv2-Small benchmark in 1-shot setting.

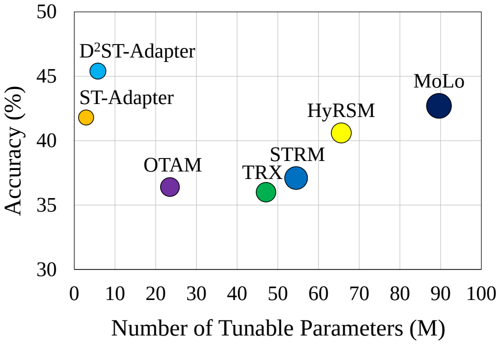


Fig. 4: Performance and efficiency of different methods with ResNet-50 backbone on SSv2-Small benchmark in 1-shot setting. Circle size indicates the memory usage.

data, including AIM, DUALPATH, and ST-Adapter, substantially on all settings, which reveals the benefit of two core technical designs of *D^{2}ST-Adapter*, namely the dual-pathway architecture for disentangled learning of spatial and temporal features and anisotropic Deformable Spatio-Temporal Attention (aDSTA) for learning features in a global view.

To gain more insight into the strengths and weaknesses of our *D^{2}ST-Adapter*, we select four categories of actions where our method excels, as well as four categories where our model struggles in comparison to the state-of-the-art MoLo and ST-Adapter. Figure 3 shows that our model achieves large performance superiority over other methods on the relatively complex actions requiring careful reasoning via learning temporal features for recognition, whilst performing poorly on simple actions that can be recognized primarily based on spatial features. Such results are consistent with the previous results that our model excels at recognizing actions involving complex temporal dynamics.

Table 2: Classification accuracy (%) of different methods on Kinetics, HMDB51 and UCF101 where spatial features are more crucial for action recognition. The highest and the second best results are highlighted in **bold** and underline, respectively.

Method	Backbone	Adaptation paradigm	Kinetics					HMDB51			UCF101		
			1-shot	2-shot	3-shot	4-shot	5-shot	1-shot	3-shot	5-shot	1-shot	3-shot	5-shot
CMN [59]	ResNet-50	Full fine-tuning	57.3	67.5	72.5	74.7	76.0	–	–	–	–	–	–
CMN-J [60]	ResNet-50	Full fine-tuning	60.5	70.0	75.6	77.3	78.9	–	–	–	–	–	–
OTAM [2]	ResNet-50	Full fine-tuning	73.0	75.9	78.7	81.9	85.8	54.5	65.7	68.0	79.9	87.0	88.9
ITANet [56]	ResNet-50	Full fine-tuning	73.6	–	–	–	84.3	–	–	–	–	–	–
TRX [28]	ResNet-50	Full fine-tuning	63.6	76.2	81.8	83.4	85.9	53.1	66.8	75.6	78.2	92.4	96.1
TA^{2}N [22]	ResNet-50	Full fine-tuning	72.8	–	–	–	85.8	59.7	–	73.9	81.9	–	95.1
MTFAN [46]	ResNet-50	Full fine-tuning	74.6	–	–	–	**87.4**	59.0	–	74.6	84.8	–	95.1
STRM [36]	ResNet-50	Full fine-tuning	62.9	76.4	81.1	83.8	86.7	52.3	67.4	77.3	80.5	92.7	96.9
HyRSM [44]	ResNet-50	Full fine-tuning	73.7	80.0	83.5	84.6	86.1	60.3	71.7	76.0	83.9	93.0	94.7
Nguyen *et al.* [24]	ResNet-50	Full fine-tuning	74.3	–	–	–	**87.4**	59.6	–	76.9	84.9	–	95.9
Huang *et al.* [15]	ResNet-50	Full fine-tuning	73.3	–	–	–	86.4	60.1	–	77.0	71.4	–	91.0
HCL [57]	ResNet-50	Full fine-tuning	73.7	79.1	82.4	84.0	85.8	59.1	71.2	76.3	82.5	91.0	93.9
SloshNet [49]	ResNet-50	Full fine-tuning	70.4	–	–	–	87.0	59.4	–	**77.5**	86.0	–	**97.1**
GgHM [50]	ResNet-50	Full fine-tuning	74.9	–	–	–	**87.4**	61.2	–	76.9	85.2	–	96.3
MoLo [43]	ResNet-50	Full fine-tuning	74.0	80.4	83.7	84.7	85.6	60.8	72.0	77.4	86.0	93.5	95.5
ST-Adapter [25]	ResNet-50	Adapter tuning	73.0	79.9	82.8	84.6	85.1	60.3	71.4	74.7	84.6	92.9	94.5
D^{2}ST-Adapter (Ours)	ResNet-50	Adapter tuning	**75.0**	**81.4**	**84.3**	**85.8**	87.0	**61.6**	**73.0**	76.6	**86.9**	**94.4**	95.6
CLIP-FSAR [42]	CLIP-ViT-B	Full fine-tuning	**89.7**	92.9	94.2	94.8	95.0	75.8	84.1	87.7	**96.6**	98.4	99.0
AIM [51]	CLIP-ViT-B	Adapter tuning	88.4	92.7	94.2	94.6	95.3	74.2	83.8	86.9	95.4	98.2	98.5
DUALPATH [26]	CLIP-ViT-B	Adapter tuning	88.8	92.8	94.3	94.7	95.4	74.9	84.6	87.5	95.7	98.4	98.7
ST-Adapter [25]	CLIP-ViT-B	Adapter tuning	88.5	92.6	94.0	94.6	95.1	74.1	84.3	87.3	95.9	98.5	98.9
D^{2}ST-Adapter (Ours)	CLIP-ViT-B	Adapter tuning	89.3	**93.1**	**94.4**	**95.2**	**95.5**	**77.1**	**86.4**	**88.2**	96.4	**98.7**	**99.1**

Benchmarks relying on spatial features. Table 2 presents the experimental results on the benchmarks relying more on spatial features, including Kinetics, HMDB51 and UCF101. Our method still performs best on most of the settings, which manifest the effectiveness and robustness of our method in these scenarios. We also observe that the performance improvement by our *D^{2}ST-Adapter* over other method, especially ST-Adapter, is smaller than that on SSv2 datasets. It is reasonable since the action recognition in these datasets relies less on the temporal features and thus the disentangled feature encoding of our method yields limited performance gain.

Comparison of efficiency. We compare the efficiency between our *D^{2}ST-Adapter* and other methods in terms of tunable parameter size and memory usage in Figure 4. It shows that the tunable parameter size of both our model and ST-Adapter is significantly smaller than other methods that are based on full fine-tuning, which reveals the advantage of the adapter tuning paradigm over the full fine-tuning paradigm for task adaptation. Meanwhile, our method outperforms these full fine-tuning based methods substantially while enjoying much fewer tunable parameters, which validates the superiority of our method.

4.3 Ablation study

We conduct ablation study on SSv2-Full and Kinetics datasets, using ResNet-50 as the pre-trained backbone and Bi-MHM [44] as the matching metric.

Comparison between different Adapters. We first compare the performance of different Adapters in Table 3. As shown in Figure 1, Vanilla-Adapter [14] only contains the downsampling and upsampling layers with a GELU nonlinearity in

Table 3: Comparison between different adapters as well as full fing-tuning.

Method	SSv2-Full		Kinetics	
	1-shot	5-shot	1-shot	5-shot
Full Fine-tuning	44.9	57.0	72.4	84.3
Vanilla-Adapter	45.3	57.6	72.6	84.8
ST-Adapter	52.2	68.7	73.0	85.1
DST-Adapter (Ours)	53.7	69.6	73.3	85.5
D^{2}ST-Adapter (Ours)	**57.0**	**73.6**	**75.0**	**87.0**

Table 4: Ablation study on the configuration of sampling kernel of aDSTA.

Configuration	SSv2-Full		Kinetics	
	1-shot	5-shot	1-shot	5-shot
aDSTA-Uniform	55.8	72.2	73.9	86.0
aDSTA-S & aDSTA-T	**57.0**	**73.6**	**75.0**	**87.0**

between. As described in Section 3.2, DST-Adapter is the convolutional version of our *D^{2}ST-Adapter* which follows the dual-pathway structure but models spatial and temporal pathways with 3D convolution. The performance of full fine-tuning is also provided for reference. We make following observations.

- Effect of learning temporal features by adapters. ST-Adapter outperforms Vanilla-Adapter distinctly, which indicates that using 3D convolution to learn joint spatio-temporal features facilitates the feature adaptation.
- Effect of disentangled encoding of the spatial and temporal features. The performance improvement from ST-Adapter to DST-Adapter demonstrates the effectiveness of dual-pathway adapter architecture, which allows for encoding the spatial and temporal features in a disentangled manner.
- Effect of the proposed anisotropic Deformable Spatial-Temporal Attention (aDSTA). Comparing our *D^{2}ST-Adapter* with DST-Adapter, the proposed aDSTA yields a large performance gain, especially on the SSv2-Full dataset, which is more challenging due to the necessity of encoding temporal features.

Effect of configuration of the sampling kernel of aDSTA. To investigate the effect of configuring specialized sampling kernels for aDSTA-S in the spatial pathway and aDSTA-T in the temporal pathway respectively, we additionally configure an aDSTA-Uniform module, which performs sampling with uniform density in both the spatial and temporal domains. Then we compare the performance of our *D^{2}ST-Adapter* using two different settings, including 1) 'aDSTA-Uniform', in which aDSTA-Uniform is used to model both the spatial and temporal pathways and 2) 'aDSTA-S & aDSTA-T', in which *D^{2}ST-Adapter* uses aDSTA-S and aDSTA-T to model the spatial and temporal pathways respectively. As shown in Table 4, the distinct performance gap between these two settings reveals the advantage of configuring specialized sampling kernels for the spatial and temporal pathways.

Generalization across different matching metrics. To evaluate the generalization of our *D^{2}ST-Adapter* across different matching metrics, we conduct

Table 5: Evaluation of generalization of our *D*^{2}*ST-Adapter* across different matching metrics on SSv2-Full and Kinetics. The state-of-the-art methods for each metric are involved into comparison.

Matching Metric	Method	SSv2-Full		Kinetics	
		1-shot	5-shot	1-shot	5-shot
OTAM [2]	RFPL [47]	47.0	61.0	74.6	**86.8**
	MoLo [43]	55.0	69.6	73.8	85.1
	D^{2}ST-Adapter	**56.0**	**72.8**	**74.7**	**86.8**
TRX [28]	RFPL [47]	44.6	64.6	66.2	87.3
	MoLo [43]	45.6	66.1	64.8	86.3
	D^{2}ST-Adapter	**47.7**	**68.6**	**66.4**	**87.6**
Bi-MHM [44]	HyRSM [44]	54.3	69.0	73.7	86.1
	MoLo [43]	56.6	70.6	74.0	85.6
	D^{2}ST-Adapter	**57.0**	**73.6**	**75.0**	**87.0**

Table 6: Performance of different methods using smaller backbones (*i.e.*, ResNet-18 and ResNet-34) on SSv2-Full dataset.

Method	ResNet-18					ResNet-34				
	1-shot	2-shot	3-shot	4-shot	5-shot	1-shot	2-shot	3-shot	4-shot	5-shot
OTAM [2]	39.4	45.0	46.6	47.4	49.0	40.6	45.2	48.0	48.9	49.2
TRX [28]	29.9	38.2	44.0	48.2	50.3	32.4	41.6	47.7	52.0	53.5
HyRSM [44]	46.6	54.7	58.7	60.7	61.1	50.0	57.5	61.9	63.3	64.8
MoLo [43]	50.0	57.2	61.6	63.6	64.6	54.1	61.1	65.9	67.3	67.8
ST-Adapter [25]	47.3	54.3	58.1	61.3	62.2	48.9	55.6	59.4	62.5	64.1
DST-Adapter (Ours)	50.0	56.9	61.7	63.4	65.3	52.1	59.3	63.4	65.8	67.5
D^{2}ST-Adapter (Ours)	**53.0**	**60.4**	**65.0**	**67.1**	**68.6**	**54.4**	**62.5**	**66.0**	**69.1**	**70.6**

the experiments using three classical matching metrics on SSv2-Full and Kinetics, including OTAM [2], TRX [28], and Bi-MHM [44], and compare our model with the state-of-the-art methods corresponding to each metric. Table 5 presents the results, which show that our *D*^{2}*ST-Adapter* consistently outperforms other methods, especially on the challenging SSv2-Full benchmark, validating the well generalization of the proposed *D*^{2}*ST-Adapter* across different matching metrics.

Instantiation on ResNet-18 and ResNet-34. To investigate the performance of our *D*^{2}*ST-Adapter* on smaller backbones, we instantiate our model with ResNet-18 and ResNet-34 pre-trained on ImageNet, and conduct experiments to compare our model with other methods under the same experimental settings. Table 6 show that our *D*^{2}*ST-Adapter* achieves the best performance using both backbones in all settings, which demonstrates the robustness of our method. While DST-Adapter, the convolutional version of our model, also performs well, it is still inferior to *D*^{2}*ST-Adapter*, which manifests the effectiveness of the proposed anisotropic Deformable Spatio-Temporal Attention (aDSTA).

4.4 Visualization

We visualize the shifted reference points learned by our *D*^{2}*ST-Adapter* to investigate the effectiveness of the proposed aDSTA qualitatively. Following Deformable

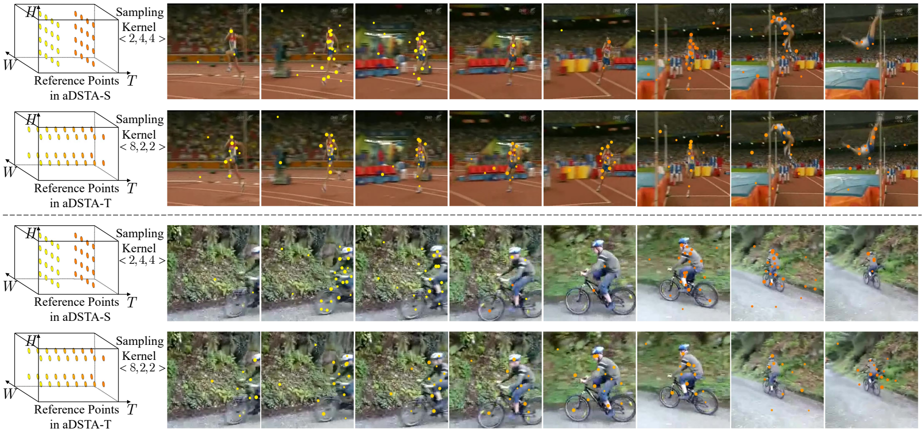

H
W
T
Sampling Kernel < 2,4,4 >
Reference Points in aDSTA-S
H
W
T
Sampling Kernel < 8,2,2 >
Reference Points in aDSTA-T
H
W
T
Sampling Kernel < 2,4,4 >
Reference Points in aDSTA-S
H
W
T
Sampling Kernel < 8,2,2 >
Reference Points in aDSTA-T

Fig. 5: Visualization of shifted reference points in both pathways for two video samples. Circle size indicates the importance for each point calculated by aggregating the attention from all queries.

Attention [48], for each shifted reference point serving as key and value, we accumulate the attention weights assigned by all queries as its relative importance. Then we visualize top-100 most important shifted reference points in each pathway for a video sample, as shown in Figure 5. Note that we sample 8 frames for all videos, if a reference point is shifted to a temporal position between two sampled frames, we visualize it on the nearest frame approximately.

The results show that the shifted reference points learned by our model can always focus on the salient targets that are critical for action recognition in both the spatial and temporal pathways. Another interesting observation is that the shifted points in the spatial pathway are distributed densely in some specific frames to capture the spatial appearance features for the targets, while the shifted points in the temporal pathway are distributed uniformly among all frames to capture the temporal dynamic features. This is consistent with the design of specialized tailored aDSTA modules for different pathways by configuring the sampling kernels accordingly.

5 Conclusion

We present *D^{2}ST-Adapter*, which is a novel adapter tuning method for few-shot action recognition. It is designed in a dual-pathway architecture, which allows for disentangled encoding of spatial and temporal features. Moreover, we design the anisotropic Deformable Spatio-Temporal Attention (aDSTA) as the essential component of *D^{2}ST-Adapter*, which enables it to encode features in a global view while maintaining lightweight design. Extensive experiments demonstrate that the superiority of our model over state-of-the-art methods, especially in challenging scenarios involving complex temporal features.

Supplementary Material

A Tuning of the Sampling Kernels of aDSTA

The sampling kernels of aDSTA-S in the spatial pathway and aDSTA-T in the temporal pathway of our D^2*ST-Adapter* can be tuned on a held-out validation set. Generally, aDSTA-S should sample denser reference points in the spatial domain while aDSTA-T samples more points in the temporal domain. For the instantiation of our model with ResNet, we configure the sampling kernel of aDSTA in each convolutional stage individually since the feature maps in different convolutional stages may have different size. In contrast, we only need to tune one configuration for sampling kernel when using ViT as the backbone since the feature map always has fixed size in different stages. Table 7 shows the tuned configurations of the sampling kernels of aDSTA-S and aDSTA-T with different backbones. Besides, we also provide the configurations of sampling kernel for aDSTA-Uniform module which is constructed to validate the effect of configuring the sampling kernel of aDSTA in Table 4.

Table 7: Tuned sampling kernels (in the form of (T, H, W)) for aDSTA-S in the spatial pathway and aDSTA-T in the temporal pathway of our D^2*ST-Adapter*. Besides, the sampling kernel for aDSTA-Uniform used in Table 4 for ablation study is also provided.

Backbone	Feature Map	Sampling Kernel in aDSTA-S	Sampling Kernel in aDSTA-T	Sampling Kernel in aDSTA-Uniform
ResNet-50	(8, 56, 56)	(2, 8, 8)	(8, 4, 4)	(4, 4, 4)
ResNet-50	(8, 28, 28)	(2, 4, 4)	(8, 2, 2)	(4, 4, 4)
ResNet-50	(8, 14, 14)	(2, 4, 4)	(8, 2, 2)	(4, 4, 4)
ResNet-50	(8, 7, 7)	(2, 2, 2)	(8, 1, 1)	(4, 4, 4)
CLIP-ViT-B	(8, 14, 14)	(2, 4, 4)	(8, 2, 2)	(4, 4, 4)

B Effect of the Inserted Position of D^2*ST-Adapter*

Theoretically, our D^2*ST-Adapter* can be inserted into any position of the backbone flexibly. To investigate the effect of the inserted position of D^2*ST-Adapter* on the performance of the model, we conduct experiments with four different ways of inserting D^2*ST-Adapters* into the pre-trained CLIP-ViT-B backbone (comprising 12 learning stages) on SSv2-Small dataset: a) early-insertion, which inserts the D^2*ST-Adapter* into each of first 6 stages (close to the input), b) late-insertion that inserts the D^2*ST-Adapter* into each of last 6 stages (close to the output), c) skip-insertion, which inserts the adapter into the backbone every two stages and d) full-insertion that inserts the adapter into each learning stage.

As shown in Table 8, late-insertion, namely inserting the proposed D^2*ST-Adapters* into the last 6 stages, yields better performance than early-insertion,

Table 8: Effect of the inserted position of *D^{2}ST-Adapter* in CLIP-ViT-B on SSv2-Small dataset. Skip means adding *D^{2}ST-Adapter* every other stage.

Insertion Position	Tunable Parameters (%)	Memory Usage	1-shot	5-shot
Early-insertion	4.1%	15.5GB	48.4	64.6
Late-insertion	4.1%	15.2GB	54.2	68.5
Skip-insertion	4.1%	15.2GB	53.3	67.9
Full-insertion	7.9%	16.7GB	**55.0**	**69.3**

Table 9: Effect of the bottleneck ratio of *D^{2}ST-Adapter* in CLIP-ViT-B on SSv2-Small dataset.

Ratio	Tunable Parameters (%)	Memory Usage	1-shot	5-shot
0.0625	1.4%	15.2GB	53.4	68.1
0.125	3.2%	15.7GB	54.2	68.6
0.25	7.9%	16.7GB	**55.0**	69.3
0.5	20.2%	18.7GB	54.7	**69.4**

which implies that adapter tuning is more effective for task adaptation in deeper layers than in the shallower layers. It is reasonable since deeper layers generally capture the high-level semantic features, which are more relevant to task adaptation. Besides, full-insertion achieves the best performance at the expense of slightly more tunable parameters and memory usage.

C Effect of the Bottleneck Ratio of *D^{2}ST-Adapter*

The tunable parameter size is mainly determined by the bottleneck ratio of *D^{2}ST-Adapter*, defined as the ratio of downsampled channel numbers to the initial size. Thus, we can balance between the model efficiency in terms of tunable parameter size and model effectiveness in terms of classification accuracy by tuning the bottleneck ratio. As shown in Table 9, larger bottleneck ratios typically yield more performance improvement while introducing more tunable parameters, and we set the bottleneck ratio to 0.25 in all the experiments based on the results.

D More Visualizations

To provide more insight into the anisotropic Deformable Spatio-Temporal Attention (aDSTA), we provide more visualization results in Figure 6. To be specific, we manually select a point within the salient object as the query token and visualize top-50 relevant shifted reference points to the query in terms of attention

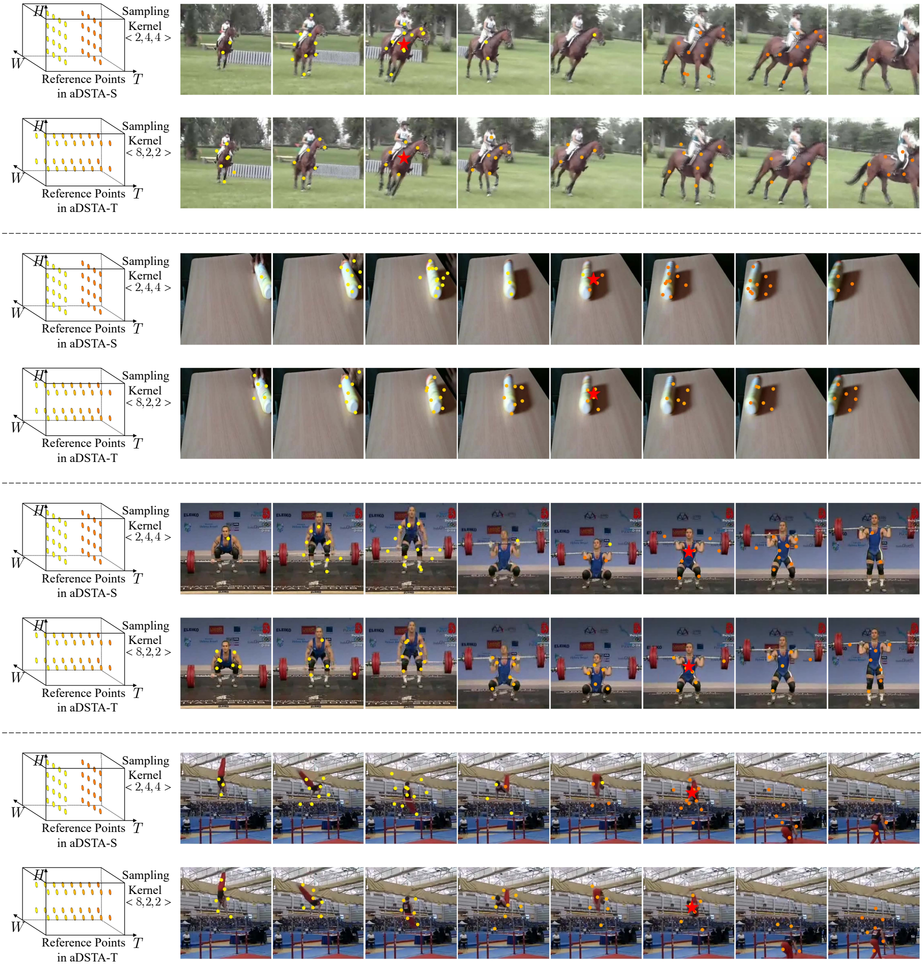

H
W
T
Sampling Kernel < 2,4,4 >
Reference Points in aDSTA-S
H
W
T
Sampling Kernel < 8,2,2 >
Reference Points in aDSTA-T
H
W
T
Sampling Kernel < 2,4,4 >
Reference Points in aDSTA-S
H
W
T
Sampling Kernel < 8,2,2 >
Reference Points in aDSTA-T
H
W
T
Sampling Kernel < 2,4,4 >
Reference Points in aDSTA-S
H
W
T
Sampling Kernel < 8,2,2 >
Reference Points in aDSTA-T
H
W
T
Sampling Kernel < 2,4,4 >
Reference Points in aDSTA-S
H
W
T
Sampling Kernel < 8,2,2 >
Reference Points in aDSTA-T

Fig. 6: Given a selected query within the salient object, top-50 relevant shifted reference points in terms of attention weights in both pathways are visualized for three video samples. The red stars denote the selected query tokens while the circles denote relevant shifted reference points.

weights. The results show that our model is able to capture the salient object through the shifted reference points in both spatial and temporal domains.

References

1. Bishay, M., Zoumpourlis, G., Patras, I.: TARN: Temporal attentive relation network for few-shot and zero-shot action recognition. In: BMVC (2019)
2. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR. pp. 10618–10627 (2020)

3. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR. pp. 6299–6308 (2017)
4. Chen, S., Ge, C., Tong, Z., Wang, J., Song, Y., Wang, J., Luo, P.: Adaptformer: Adapting vision transformers for scalable visual recognition. NeurIPS **35**, 16664–16678 (2022)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009)
6. Doersch, C., Gupta, A., Zisserman, A.: Crosstransformers: spatially-aware few-shot transfer. NeurIPS **33**, 21981–21993 (2020)
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
8. Feichtenhofer, C., Fan, H., Malik, J., He, K.: SlowFast networks for video recognition. In: ICCV. pp. 6202–6211 (2019)
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML. pp. 1126–1135 (2017)
10. Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fruend, I., Yianilos, P., Mueller-Freitag, M., et al.: The" something something" video database for learning and evaluating visual common sense. In: ICCV. pp. 5842–5850 (2017)
11. Hariharan, B., Girshick, R.: Low-shot visual recognition by shrinking and hallucinating features. In: ICCV. pp. 3018–3027 (2017)
12. He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., Neubig, G.: Towards a unified view of parameter-efficient transfer learning. In: ICLR (2021)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
14. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: ICML. pp. 2790–2799 (2019)
15. Huang, Y., Yang, L., Sato, Y.: Compound prototype matching for few-shot action recognition. In: ECCV. pp. 351–368 (2022)
16. Jamal, M.A., Qi, G.J.: Task agnostic meta-learning for few-shot learning. In: CVPR. pp. 11719–11727 (2019)
17. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV. pp. 709–727 (2022)
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
19. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: ICCV. pp. 2556–2563 (2011)
20. Li, K., Zhang, Y., Li, K., Fu, Y.: Adversarial feature hallucination networks for few-shot learning. In: CVPR. pp. 13470–13479 (2020)
21. Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatiotemporal representation learning. In: ICLR (2022)
22. Li, S., Liu, H., Qian, R., Li, Y., See, J., Fei, M., Yu, X., Lin, W.: TA2N: Two-stage action alignment network for few-shot action recognition. In: AAAI. pp. 1404–1411 (2022)
23. Müller, M.: Dynamic time warping. Information Retrieval for Music and Motion pp. 69–84 (2007)
24. Nguyen, K.D., Tran, Q.H., Nguyen, K., Hua, B.S., Nguyen, R.: Inductive and transductive few-shot video classification via appearance and temporal alignments. In: ECCV. pp. 471–487 (2022)

25. Pan, J., Lin, Z., Zhu, X., Shao, J., Li, H.: ST-Adapter: Parameter-efficient image-to-video transfer learning. In: NeurIPS (2022)
26. Park, J., Lee, J., Sohn, K.: Dual-path adaptation from image to video transformers. In: CVPR. pp. 2203–2213 (2023)
27. Pei, W., Feng, X., Fu, C., Lu, G., Yu-Wing, T.: Learning sequence representations by non-local recurrent neural memory. IJCV **130**(10), 2532–2552 (2022)
28. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational CrossTransformers for few-shot action recognition. In: CVPR. pp. 475–484 (2021)
29. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021)
30. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: ICLR (2017)
31. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. NeurIPS **27** (2014)
32. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. NeurIPS **30** (2017)
33. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
34. Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P.H., Hospedales, T.M.: Learning to compare: Relation network for few-shot learning. In: CVPR. pp. 1199–1208 (2018)
35. Sung, Y.L., Cho, J., Bansal, M.: Lst: Ladder side-tuning for parameter and memory efficient transfer learning. NeurIPS **35**, 12991–13005 (2022)
36. Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S., Ghanem, B.: Spatio-temporal relation modeling for few-shot action recognition. In: CVPR. pp. 19958–19967 (2022)
37. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: ICCV. pp. 4489–4497 (2015)
38. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. pp. 6450–6459 (2018)
39. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. NeurIPS **29** (2016)
40. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV. pp. 20–36 (2016)
41. Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X.J., Ji, J., Cao, G., Jiang, D., Zhou, M.: K-adapter: Infusing knowledge into pre-trained models with adapters. In: ACL. pp. 1405–1418 (2021)
42. Wang, X., Zhang, S., Cen, J., Gao, C., Zhang, Y., Zhao, D., Sang, N.: Clip-guided prototype modulating for few-shot action recognition. arXiv preprint arXiv:2303.02982 (2023)
43. Wang, X., Zhang, S., Qing, Z., Gao, C., Zhang, Y., Zhao, D., Sang, N.: Molo: Motion-augmented long-short contrastive learning for few-shot action recognition. In: CVPR. pp. 18011–18021 (2023)
44. Wang, X., Zhang, S., Qing, Z., Tang, M., Zuo, Z., Gao, C., Jin, R., Sang, N.: Hybrid relation guided set matching for few-shot action recognition. In: CVPR. pp. 19948–19957 (2022)
45. Wang, Y.X., Girshick, R., Hebert, M., Hariharan, B.: Low-shot learning from imaginary data. In: CVPR. pp. 7278–7286 (2018)

46. Wu, J., Zhang, T., Zhang, Z., Wu, F., Zhang, Y.: Motion-modulated temporal fragment alignment network for few-shot action recognition. In: CVPR. pp. 9151–9160 (2022)
47. Xia, H., Li, K., Min, M.R., Ding, Z.: Few-shot video classification via representation fusion and promotion learning. In: ICCV. pp. 19311–19320 (2023)
48. Xia, Z., Pan, X., Song, S., Li, L.E., Huang, G.: Vision transformer with deformable attention. In: CVPR. pp. 4794–4803 (2022)
49. Xing, J., Wang, M., Liu, Y., Mu, B.: Revisiting the spatial and temporal modeling for few-shot action recognition. In: AAAI. pp. 3001–3009 (2023)
50. Xing, J., Wang, M., Ruan, Y., Chen, B., Guo, Y., Mu, B., Dai, G., Wang, J., Liu, Y.: Boosting few-shot action recognition with graph-guided hybrid matching. In: ICCV. pp. 1740–1750 (2023)
51. Yang, T., Zhu, Y., Xie, Y., Zhang, A., Chen, C., Li, M.: AIM: Adapting image models for efficient video action recognition. In: ICLR (2022)
52. Ye, H.J., Hu, H., Zhan, D.C., Sha, F.: Few-shot learning via embedding adaptation with set-to-set functions. In: CVPR. pp. 8808–8817 (2020)
53. Yoon, S.W., Seo, J., Moon, J.: TAPNet: Neural network augmented with task-adaptive projection for few-shot learning. In: ICML. pp. 7115–7123 (2019)
54. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: ECCV. pp. 525–542 (2020)
55. Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., Song, Y.: MetaGAN: An adversarial approach to few-shot learning. NeurIPS **31** (2018)
56. Zhang, S., Zhou, J., He, X.: Learning implicit temporal alignment for few-shot video classification. In: IJCAI (2021)
57. Zheng, S., Chen, S., Jin, Q.: Few-shot action recognition with hierarchical matching and contrastive learning. In: ECCV. pp. 297–313 (2022)
58. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV **130**(9), 2337–2348 (2022)
59. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: ECCV. pp. 751–766 (2018)
60. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. IEEE TPAMI **44**(1), 273–285 (2020)