Pattern Recognition 169 (2026) 111902

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

MA-FSAR: Multimodal Adaptation of CLIP for few-shot action recognition

Jiazheng Xing [a,1], Jian Zhao [b,c,1], Chao Xu [a], Mengmeng Wang [d], Guang Dai [e], Yong Liu [a,*], Jingdong Wang [f], Xuelong Li [b,c]

[a] State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, China
[b] Institute of AI (TeleAI), China
[c] School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University (NWPU), China
[d] Zhejiang University of Technology, China
[e] SGIT AI Lab, State Grid Corporation of China, China
[f] Baidu Inc., China

ARTICLE INFO

Keywords:
Few-shot action recognition
Large model application
Parameter-efficient Fine-tuning
Multimodal learning

ABSTRACT

Applying large-scale vision-language pre-trained models like CLIP to few-shot action recognition (FSAR) can significantly enhance both performance and efficiency. While several studies have recognized this advantage, most rely on full-parameter fine-tuning to adapt CLIP's visual encoder to FSAR data, which not only incurs high computational costs but also overlooks the potential of the visual encoder to engage in temporal modeling and focus on targeted semantics directly. To tackle these issues, we introduce MA-FSAR, a framework that employs the Parameter-Efficient Fine-Tuning (PEFT) technique to enhance the CLIP visual encoder in terms of action-related temporal and semantic representations. Our solution involves a token-level Fine-grained Multimodal Adaptation mechanism: a Global Temporal Adaptation captures motion cues from video sequences, while a Local Multimodal Adaptation integrates text-guided semantics from the support set to emphasize action-critical features. Additionally, we propose a prototype-level text-guided construction module to further enrich the temporal and semantic characteristics of video prototypes. Extensive experiments demonstrate our superior performance in various tasks using minor trainable parameters.

1. Introduction

Few-shot action recognition (FSAR) aims to quickly learn new action categories using limited labeled samples. Unlike conventional action recognition (AR), FSAR is characterized by the extremely limited amount of labeled data available for each task and the wide variety of distinct task types. Therefore, FSAR necessitates the development of models capable of swiftly adapting to different tasks, making this endeavor exceptionally challenging. Previous approaches [1–5] mainly focused on the metric-based meta-learning paradigm and episode training to facilitate the transfer to new classes. However, relying solely on this paradigm still requires the model to spend much time training on different datasets, which somewhat hinders its application in the industry.

In recent years, more and more large-scale foundation vision-language models (VLM) have emerged, like CLIP [6], ALIGN [7], and Florence [8]. As a consequence, researchers have actively delved into methods to effectively adapt these large models to their specific downstream tasks, such as action recognition [9], segmentation [10], and object detection [11]. Undoubtedly, applying the "pre-training, fine-tuning" paradigm leverages the power of robust pre-trained models, thus eliminating the need to train a network from scratch and obtain impressive performance. In few-shot action recognition, existing FSAR methods like CLIP-FSAR [12] and MVP-shot [13] have made preliminary attempts, but they opt for full-parameter fine-tuning of the CLIP visual encoder with high computation costs, as shown in Fig. 1(a)(ii). More critically, these approaches focus merely on domain adaptation while relegating temporal modeling and semantic distillation to post-hoc prototype matching processes. Given these limitations, we propose to fundamentally enhance CLIP's intrinsic capacity for temporal and semantic understanding through Parameter-Efficient Fine-Tuning (PEFT), a paradigm proven effective in standard action recognition (AR) through methods like AIM [14] and ST-Adapter [15]. Its core idea is to keep the large pre-trained foundation model frozen and introduce trainable adapters [14,16,17] or prompts [18,19] for efficient fine-tuning to achieve robust performance among various tasks.

[*] Correspondence to: State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, 310027, China.
E-mail addresses: jiazhengxing@zju.edu.cn (J. Xing), zhaoj90@chinatelecom.cn (J. Zhao), yongliu@iipc.zju.edu.cn (Y. Liu).

[1] The two authors contribute equally to this work.

https://doi.org/10.1016/j.patcog.2025.111902
Received 26 October 2024; Received in revised form 14 April 2025; Accepted 22 May 2025
Available online 9 June 2025

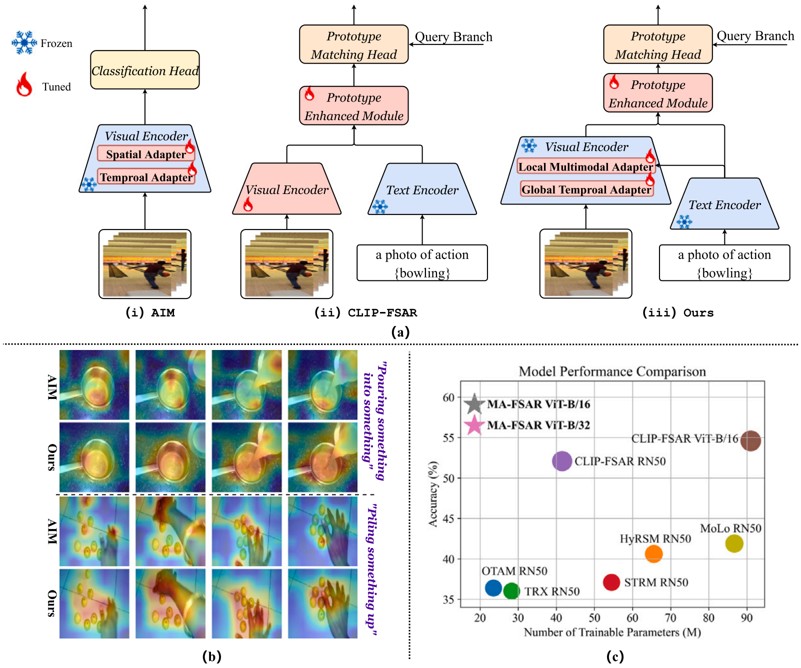


Fig. 1. (a): (i) AIM [14], a method that successfully applied PEFT technology in action recognition; (ii) The support branch of CLIP-FSAR [12], a representative method that fully fine-tunes CLIP for few-shot action recognition; and (iii) the pipeline of our proposed method's support branch. (b): Visualization of the attention map at the visual encoder's last layer for the proposed MA-FSAR and AIM [14]. AIM serves for action recognition as a classification task, whereas few-shot action recognition is a matching task. Therefore, for a fair comparison, both methods use the same few-shot temporal alignment metric, OTAM [1]. For the comparison result, the attention maps from our method are more focused on action-related objects due to the integration of textual tokens and visual tokens in the visual encoder. (c): Performance comparison of different few-shot action recognition methods in the SSv2-Small 5-way 1-shot task, including our **MA-FSAR**, OTAM [1], TRX [4], STRM [20], HyRSM [5], MoLo [21] and CLIP-FSAR [12]. Bubble or star size indicates the recognition accuracy. Our **MA-FSAR** achieves the highest recognition accuracy with the least number of trainable parameters.

PEFT's application in AR can be broadly categorized into two main approaches: Adapter-based and Prompt-tuning-based. For the sake of convenience and performance, we choose to use the Adapter-based technique in FSAR. However, directly borrowing the AR-oriented Adapter fails to address FSAR's core task requirement. Since FSAR is a matching task rather than the classification task as AR, more discriminative features are required to establish class prototype centers for each task. Concretely, as evidenced in Fig. 1(a)(i), the Adapter-based AR method AIM [14] employs a temporal Adapter to integrate temporal information into the CLIP encoder, but the visual tokens encompass numerous cues unrelated to the action, which can diminish the discriminativeness of temporal features. More critically, the Adapter-based AR methods do not need to consider FSAR's unique textual resources. The support set in FSAR contains distinctive labeled text information that can be used as textual features to bolster the semantic discriminativeness of dynamic features.

To address these issues, we propose a novel method, dubbed **MA-FSAR**, a short for **M**ultimodal **A**daptation of CLIP for **F**ew-**S**hot **A**ction **R**ecognition. Our solution is built upon Adapter-based technique of PEFT and incorporates a Fine-grained Multimodal Adaptation (FgMA) tailored for FSAR without altering the original CLIP weights, thus enabling action-related temporal awareness and semantic mining capabilities to be seamlessly integrated into the CLIP visual encoder in an effective and efficient manner. Specifically, we first introduce a Global Temporal Adaptation that processes only the class token, extracted from the visual tokens, which contains the precise semantics of each frame. This approach efficiently captures global dynamic cues while minimizing computational costs. These outputs provide an accurate motion prior to the subsequent Local Multimodal Adaptation and are integrated with the current video tokens to form new visual tokens. Additionally, the module is capable of integrating text features specific to the FSAR support set as textual tokens. In this module, the query branch utilizes the updated visual tokens to focus on learning spatiotemporal details, while the support branch combines the visual and textual tokens to achieve multimodal modeling, highlighting the fine-grained semantics associated with actions. As shown in Fig. 1(b), compared to the adapter used in AIM [14], which does not incorporate textual tokens, our method's attention maps are more focused on action-related objects. After completing the token-level designs of CLIP, we propose a Text-guided Prototype-level Construction Module (TPCM) to further enrich the temporal and semantic characteristics of video prototypes, which aids in optimizing both intra-class and inter-class correlations of video features. In summary, our proposed plug-and-play MA-FSAR can flexibly integrate with any common FSAR matching metric, ensuring the efficient and effective application of CLIP to FSAR. Extensive experiments unequivocally demonstrate that our method attains exceptional performance while employing the fewest tunable parameters, as shown in Fig. 1(c). We make the following contributions:

- We propose a novel method, **MA-FSAR**, to refine CLIP's visual encoder at the token level for FSAR by introducing the Fine-grained Multimodal Adaptation with the enhancement of the action-related temporal and semantic representations, which is fast, efficient, and cost-effective in training.
- At the prototype level, we propose a Text-guided Prototype Construction Module to further enhance the temporal and semantic representation of video prototypes.
- Experiments demonstrate that our method performs excellently in various task settings on five widely used datasets with minimal trainable parameters.

2. Related works

2.1. Few-shot learning

Research on few-shot learning can be mainly classified into adaptation-based and metric-based methods. The former aims to find a network initialization that can be fine-tuned for unknown tasks using limited labeled data, called *gradient by gradient*. Classical adaptation-based approaches include MAML [22] and Reptile [23], with further in-depth research found in [24,25]. The latter aims to acquire knowledge of feature space and compare task features using various matching strategies, referred to as *learning to compare*. Representative methods include Prototypical Networks [26] and Matching Networks [27], with many approaches [28,29] aiming to make improvements based on these models.

2.2. Few-shot action recognition

The core concept of few-shot action recognition (FSAR) is similar to few-shot learning (FSL), but the inclusion of the temporal dimension increases the complexity of the problem. Adaptation-based methods such as MetaUVFS [30] have received limited attention in FSAR due to their high computational demands and extensive experimental time. Therefore, existing research predominantly emphasizes metric-based learning approaches with varying focuses. On the one hand, some methods focus on class prototype matching metric strategies. OTAM [1] introduces a temporal alignment metric to calculate the distance value between query and support set videos. TRX [4] matches each query sub-sequence with all sub-sequences in the support set, facilitating correspondences between different videos. And, HyRSM [5] proposes a bidirectional Mean Hausdorff Metric that exhibits robustness to complex actions. On the other hand, certain approaches aim to enhance feature or class prototype representations. STRM [20] adopts local and global enrichment modules for features' spatiotemporal modeling. And, HyRSM [5] utilizes hybrid relation modeling to learn task-specific embeddings. Recently, with the development of large foundation vision-language models, their application in FSAR is receiving increasing attention. The most representative work, CLIP-FSAR [12], refines CLIP for FSAR by fully fine-tuning CLIP's visual encoder and designing a prototype modulation module to enhance multimodal representations at the prototype level. However, despite its significant computational overhead, it falls short in temporal modeling and multimodal feature fusion within the CLIP visual encoder. Meanwhile, other multimodal-based methods have explored different aspects. For instance, CLIP-MDMF [31] explores multi-modal information from different views and subsequently performs multi-view fusion. Similarly, MVP-shot [13] progressively learns and aligns semantic-related action features at multi-velocity levels. In contrast, CLIP-CPM^{2}C [32] develops a consistency prototype and motion compensation network based on CLIP to efficiently leverage both text labels and motion features from videos.

2.3. Parameter-efficient fine-tuning (PEFT) for vision models

Parameter-efficient Fine-tuning (PEFT) technique, initially employed in Natural Language Processing [33,34], has exhibited impressive advancements in Computer Vision in recent times. Its application in video understanding can be broadly categorized into two main approaches: Adapter-based and Prompt-tuning-based. The design of the Adapter originates from [33]. It adds two fully connected layers (FC) with residual structures in each transformer layer to fine-tune the model, where the original transformer is frozen and only the adapter layer is trained during the process. Inspired by this, AIM [14] applies the Adapter technique in action recognition. In each Vision Transformer (ViT) [35] block, AIM designs three adapters for spatial, temporal, and joint adaptation, achieving excellent results. Besides that, ST-Adapter [15] introduced a parameter-efficient spatiotemporal adapter, effectively leveraging the capabilities of CLIP's image models for video understanding. As for Prompt-tuning, it refers to the flexible adjustment of prompts, which can significantly impact the final performance of the model. The pioneering use of Prompt-tuning in the visual domain is by VPT [36]. It introduces learnable prompts within ViT while freezing the other training parameters in the network and achieves impressive results in image-related downstream tasks. Inspired by this, Vita-CLIP [18] designs the Prompt-tuning method specifically for videos, which proposes the learnable video summary tokens, frame-level prompts, and video-level prompts, achieving impressive results. In this work, we incorporate the Adapter-based technique of PEFT into the FSAR task to enhance the performance of the CLIP visual encoder without demanding excessive computational resources.

3. Method

3.1. Problem formulation

In few-shot action recognition, the objective is to classify an unlabeled query video into one of the M action categories within the support set, with only K limited samples per action class. This is considered an M-way K-shot task. Similar to previous researches [1–5], we follow the episode training framework, where episodes are randomly selected from a vast pool of collected data. In each episode, we assume that the set $\mathcal{S}$ comprises $M \times K$ samples originating from M different action classes. Specifically, $S_k^m = \left\{s_{k1}^m, s_{k2}^m, \dots, s_{kT}^m\right\}$ denotes the kth video in class $m \in \{1, \dots, M\}$, randomly sampled with T frames. And, the query video is represented as $Q = \left\{q_1, q_2, \dots, q_T\right\}$, also sampled with T frames.

3.2. Architecture overview

In this work, we choose CLIP [6] as the pre-trained foundation vision-language model, which features a dual-encoder structure consisting of visual and text encoders. It can perform cross-modal reasoning and achieve mutual conversion between images and texts. For the vision branch, we select the ViT (Vision Transformer) [35] architecture from CLIP as our visual encoder due to its powerful feature encoding capabilities and its flexible token-based learning network structure, which facilitates the application of the Adapter-based PEFT technique. For the text branch, labeled input texts are typically combined with prompt templates before passing through the encoder, and the method for selecting these templates is detailed in (Section 4.1.2).

Our overall architecture is illustrated in Fig. 2. For the frame-selecting strategy, we use the approach from TSN [37], which divides the input video sequence into T segments and extracts snippets from each segment. For simplicity and convenience, we will focus on a specific scenario: the 5-way 1-shot problem with a query set $\mathcal{Q}$ containing a single video. In this pipeline, the query video $Q = \left\{q_1, q_2, \dots, q_T\right\}$ and the class support set videos $S^m = \left\{s_1^m, s_2^m, \dots, s_T^m\right\}$ $\left(S^m \in \mathcal{S} = \left\{S^1, S^2, \dots, S^5\right\}\right)$ pass through the visual encoder with the Fine-grained Multimodal Adaptation (FgMA) to obtain the query feature $\mathbf{F}_{\mathcal{Q}}$ and the support features $\mathbf{F}_{\mathcal{S}}^m$ $\left(\mathbf{F}_{\mathcal{S}}^m \in \mathbf{F}_{\mathcal{S}}\right)$ in each episode. And, the text label descriptions combined with the prompt template C^m $\left(C^m \in \mathcal{C} = \left\{C^1, C^2, \dots, C^5\right\}\right)$ pass through the text encoder to obtain text features $\mathbf{F}_{\mathcal{T}}^m$ $\left(\mathbf{F}_{\mathcal{T}}^m \in \mathbf{F}_{\mathcal{T}}\right)$. Then we apply global average pooling operation to $\mathbf{F}_{\mathcal{S}}$ and $\mathbf{F}_{\mathcal{Q}}$ to obtain $\mathbf{F}_{\mathcal{S}}^{avg}$ and $\mathbf{F}_{\mathcal{Q}}^{avg}$. The Kullback–Leibler divergence losses $\mathcal{L}_{\mathcal{S}2\mathcal{T}}$ and $\mathcal{L}_{\mathcal{Q}2\mathcal{T}}$ are calculated by the cosine similarity metric between $\mathbf{F}_{\mathcal{S}}^{avg}$, $\mathbf{F}_{\mathcal{Q}}^{avg}$, and $\mathbf{F}_{\mathcal{T}}$, facilitating the adaptation of CLIP to FSAR domain. And, the probability distribution $\mathbf{p}_{\mathcal{Q}2\mathcal{T}}$ is derived using the cosine similarity metric. Then, $\mathbf{F}_{\mathcal{S}}$ and $\mathbf{F}_{\mathcal{Q}}$ are passed through the Text-guided Prototype Construction Module (TPCM) with weight sharing to obtain the final features before the prototype matching, denoted as $\widetilde{\mathbf{F}_{\mathcal{S}}}$ and $\widetilde{\mathbf{F}_{\mathcal{Q}}}$. Finally, the enhanced features are fed into the prototype matching metric to obtain the probability distribution $\mathbf{p}_{\mathcal{Q}2\mathcal{S}}$ and loss $\mathcal{L}_{\mathcal{Q}2\mathcal{S}}$.

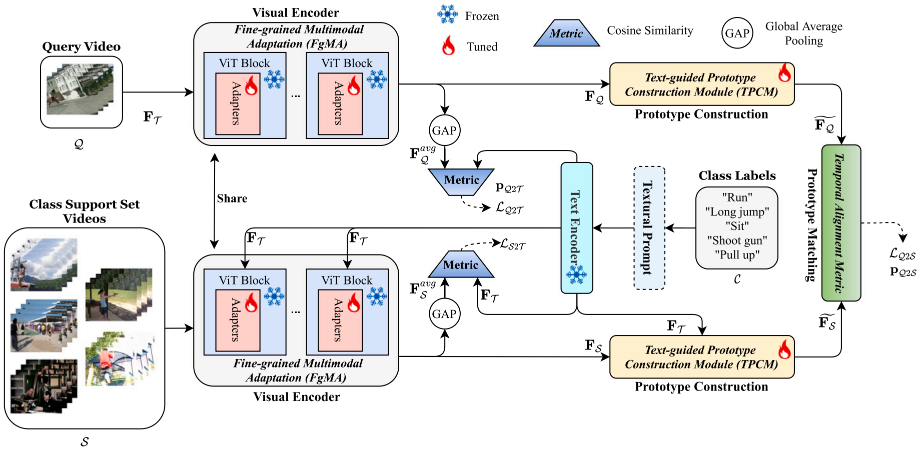

Frozen
Tuned
Metric
Cosine Similarity
GAP
Global Average Pooling
Query Video
$\mathcal{Q}$
$\mathbf{F}_\mathcal{T}$
Visual Encoder
Fine-grained Multimodal Adaptation (FgMA)
ViT Block
Adapters
...
$\mathbf{F}_\mathcal{Q}$
Text-guided Prototype Construction Module (TPCM)
Prototype Construction
$\widetilde{\mathbf{F}_\mathcal{Q}}$
$\mathbf{F}_\mathcal{Q}^{avg}$
$\mathbf{p}_{\mathcal{Q}2\mathcal{T}}$
$\mathcal{L}_{\mathcal{Q}2\mathcal{T}}$
Share
Text Encoder
Textural Prompt
Class Labels
"Run"
"Long jump"
"Sit"
"Shoot gun"
"Pull up"
$\mathcal{C}$
Prototype Matching
Temporal Alignment Metric
$\mathcal{L}_{\mathcal{Q}2\mathcal{S}}$
$\mathbf{p}_{\mathcal{Q}2\mathcal{S}}$
Class Support Set Videos
$\mathcal{S}$
$\mathcal{L}_{\mathcal{S}2\mathcal{T}}$
$\mathbf{F}_\mathcal{S}^{avg}$
$\mathbf{F}_\mathcal{S}$
$\widetilde{\mathbf{F}_\mathcal{S}}$

Fig. 2. Overview of **MA-FSAR**. For simplicity and convenience, we focus on a specific scenario: the 5-way 1-shot task with a query set $\mathcal{Q}$ containing a single video. The support set video features $\mathbf{F}_\mathcal{S}$ and query video feature $\mathbf{F}_\mathcal{Q}$ are obtained by the visual encoder with the Fine-grained Multimodal Adaptation (FgMA). Text features $\mathbf{F}_\mathcal{T}$ are obtained through a text encoder. The Text-guided Prototype Construction Module (TPCM) generates the final features before the prototype matching, denoted as $\widetilde{\mathbf{F}_\mathcal{S}}$ and $\widetilde{\mathbf{F}_\mathcal{Q}}$. The probability distribution $\mathbf{p}_{\mathcal{Q}2\mathcal{T}}$ is obtained using cosine similarity metric, and $\mathbf{p}_{\mathcal{Q}2\mathcal{S}}$ is calculated using prototype matching metric. The loss $\mathcal{L}_{\mathcal{Q}2\mathcal{S}}$ is the standard Cross-Entropy loss, while $\mathcal{L}_{\mathcal{S}2\mathcal{T}}$ and $\mathcal{L}_{\mathcal{Q}2\mathcal{T}}$ are Kullback–Leibler divergence (KL) losses.

3.3. Fine-grained multimodal adaptation (FgMA)

We employ the Parameter-Efficient Fine-Tuning (PEFT) technique to refine CLIP for the few-shot action recognition (FSAR) domain with minimal trainable parameters. Due to Adapter's [33] simplicity and Adapter-based method's [14] success in action recognition, we propose the Fine-grained Multimodal Adaptation (FgMA) tailored for FSAR, which can enhance temporal modeling and targeted semantics focusing at the token level. This approach freezes the pre-trained image and text encoders during training while introducing new, lightweight, learnable adapters.

In this subsection, since we explore how to design adaptations for the Vision Transformer in FSAR, we first provide an overview of the conventional ViT Block. Given an input video $V \in \mathbb{R}^{T\times H\times W\times 3}$, each frame is split into $N = HW/P^2$ patches. These patches are projected into token embeddings $\mathbf{x}_{t,p} \in \mathbb{R}^{N\times D}$ via a linear projection $\mathbf{E}$, prepended with a class token $\mathbf{x}_{cls}$ to form the initial sequence $\mathbf{x}_t^{(0)} = \left[\mathbf{x}_{cls}; \mathbf{x}_{t,p}\right]$. Spatial positional encoding $\mathbf{e}_{pos}$ is added:

$$\mathbf{z}_t^{(0)} = \mathbf{x}_t^{(0)} + \mathbf{e}_{pos} \tag{1}$$

Each ViT block, as shown in Fig. 3(b), processes input tokens through:

$$\mathbf{z'}_t^{(l)} = \mathbf{z}_t^{(l-1)} + \mathrm{MSA}\left(\mathrm{LN}\left(\mathbf{z}_t^{(l-1)}\right)\right) \tag{2}$$

$$\mathbf{z}_t^{(l)} = \mathbf{z'}_t^{(l)} + \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z'}_t^{(l)}\right)\right) \tag{3}$$

where $\mathbf{z}_t^{(l)}$ denotes the output of the lth layer. The video-level representation aggregates frame tokens across T frames: $\mathbf{z}^{(l)} = \left[\mathbf{z}_0^{(l)} \cdots \mathbf{z}_t^{(l)} \cdots \mathbf{z}_T^{(l)}\right]$.

As for our Fine-grained Multimodal Adaptation (FgMA), it can be divided into three parts: Global Temporal Adaptation, Local Spatiotemporal/Multimodal Adaptation, and Joint Adaptation. Each Adaptation contains a frozen attention layer and a trainable adapter with a straightforward structure that includes two fully connected layers (FC), an activation layer, and a residual connection, as depicted in Fig. 3(a). The key to the aforementioned two Adaptation lies in controlling the attention layer to perform feature modeling with different tendencies across various dimensions at the token level through fine-tuning the Adapter. In FSAR, the label information of the support set is known, while that of the query set is unknown, resulting in differing network structures for each, shown in Fig. 3(c) and (d). In what follows, we introduce three types of Adaptation:

3.3.1. Global temporal adaptation (GTA)

Temporal modeling is essential for FSAR to capture motion evolution across frames, but directly involving all visual tokens in exploring temporal relationships would incur prohibitive quadratic complexity $O\left(T^2(N+1)\right)$. Meanwhile, the visual tokens encompass numerous cues unrelated to the action, and applying all of the tokens might affect the salience of capturing temporal signals. To resolve this, our Global Temporal Adaptation (GTA) selectively processes class tokens, which are compressed semantic summaries of each frame, through lightweight temporal attention. Our design achieves the complexity reduction from $O\left(T^2(N+1)\right)$ to $O\left(T^2\right)$ compared to dense token methods while amplifying motion signal-to-noise ratio, as class tokens inherently filter non-action regions. Specifically, for the lth layer, given the input video [class] token embedding $\mathbf{x}_{cls}^{(l-1)} \in \mathbb{R}^{T\times 1\times D}$, we reshape it into $\mathbf{x}_{TA}^{(l-1)} \in \mathbb{R}^{1\times T\times D}$. Then we feed $\mathbf{x}_{TA}^{(l-1)}$ into Global Temporal Adaptation to capture the global dynamic cues between multiple frames, given by:

$$\mathbf{x}_{TA}^{(l)} = \mathbf{x}_{TA}^{(l-1)} + \mathrm{Adapter}\left(\mathrm{GT\text{-}MSA}\left(\mathrm{LN}\left(\mathbf{x}_{TA}^{(l-1)}\right)\right)\right) \tag{4}$$

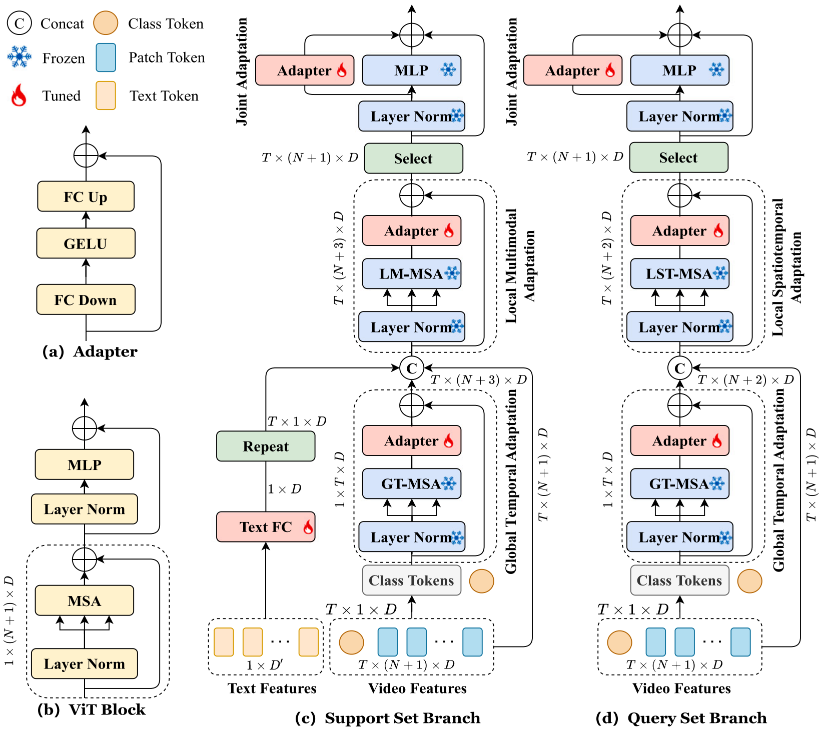
where $\mathbf{x}_{TA}^{(l-1)}$ and $\mathbf{x}_{TA}^{(l)}$ denote the Global Temporal Adaptation input and output of the lth transformer block. Self-attention GT-MSA operates on the temporal dimension T to learn the global temporal relationships between multiple frames. The Adapter structure maintains the same configuration as shown in Fig. 3(a). However, the skip connection is removed to separate the influence of the adaptation during the initial training phase.

3.3.2. Local spatiotemporal/multimodal adaptation (LSTA/LMA)

The distilled global temporal priors from GTA establish foundational motion semantics. Building upon this, we devise Local Spatiotemporal Adaptation (LSTA) that hierarchically injects global motion contexts into local patch tokens to guide local visual tokens in performing local spatiotemporal modeling. Specifically, in the query branch, we can concatenate each frame's input visual tokens $\mathbf{z}^{(l-1)}$ and the corresponding global temporal token $\mathbf{x}_{TA}^{(l)}$ along the spatial dimension to obtain $\mathbf{z}_{STA\text{-}Q}^{(l-1)} = \left[\mathbf{z}^{(l-1)}; \mathbf{x}_{TA}^{(l)}\right] \in \mathbb{R}^{T\times(N+2)\times D}$. Then we feed $\mathbf{z}_{MA-Q}^{(l-1)}$ into the Local Spatiotemporal Adaptation as shown in Fig. 3(d), written by:

$$\mathbf{z}_{STA\text{-}Q}^{(l)} = \mathbf{z}_{STA\text{-}Q}^{(l-1)} + \mathrm{Adapter}\left(\mathrm{LST\text{-}MSA}\left(\mathrm{LN}\left(\mathbf{z}_{STA\text{-}Q}^{(l-1)}\right)\right)\right) \tag{5}$$

where $\mathbf{z}_{STA\text{-}Q}^{(l-1)}$ and $\mathbf{z}_{STA\text{-}Q}^{(l)}$ denote the Local Spatiotemporal Adaptation input and output of the lth transformer block. Meanwhile, self-attention ST-MSA operates on the spatial dimension that has merged



(a) Adapter

(b) ViT Block

(c) Support Set Branch

(d) Query Set Branch

Fig. 3. (a) shows the structure of the Adapter [33], and (b) shows the structure of a standard ViT [35] block. (c) and (d) illustrate the fine-grained multimodal adaptation of each ViT block for the support and query set branch. Note that GT-MSA, LM-MSA, and LST-MSA share weights but are applied to different inputs with different motivations for global temporal, local multimodal, and local spatiotemporal modeling.

spatiotemporal tokens along it to explore the local spatiotemporal relationships.

It is important to note that, unlike the query branch, the support branch is equipped with class text labels corresponding to support videos. This semantic alignment enables the Local Spatiotemporal Adaptation (LSTA) to transition into Local Multimodal Adaptation (LMA) by architecturally integrating textual tokens into each frame's visual token sequence. By introducing text priors, the module enhances its focus on action semantics during spatiotemporal modeling, thus achieving multimodal modeling. Specifically, we input the labeled text description corresponding to each support set video $C^m \in \mathcal{C}$ into the CLIP text encoder to get textual tokens $\mathbf{F}_{\mathcal{T}}^{m}$ $\left(\mathbf{F}_{\mathcal{T}}^{m} \in \mathbf{F}_{\mathcal{T}}\right)$. The text encoder is frozen to avoid the extra computation cost and catastrophic forgetting phenomenon. To facilitate the fusion of multimodal data, we process the text features $\mathbf{F}_{\mathcal{T}}^{m} \in \mathbb{R}^{1\times D'}$ as follows:

$$\mathbf{F}_{\mathcal{T}}^{MA} = \text{Repeat}\left(\text{FC}_{text}\left(\mathbf{F}_{\mathcal{T}}^{m}\right)\right) \tag{6}$$

where $\text{FC}_{text} \in \mathbb{R}^{D'\times D}$ aims to align textual tokens with video tokens in the feature dimension, and the FC_{text} weights are shared across all layers of the visual transformer. The Repeat operation duplicates text features T times to obtain $\mathbf{F}_{\mathcal{T}}^{MA} \in \mathbb{R}^{T\times 1\times D}$. For the support set branch, given the global temporal token $\mathbf{x}_{TA}^{(l)} \in \mathbb{R}^{T\times 1\times D}$, the input visual tokens $\mathbf{z}^{(l-1)} \in \mathbb{R}^{T\times (N+1)\times D}$ and the text semantic token $\mathbf{F}_{\mathcal{T}}^{MA} \in \mathbb{R}^{T\times 1\times D}$, we concatenate these tokens together along the spatial dimension to obtain $\mathbf{z}_{MA\text{-}S}^{(l-1)} = \left[\mathbf{z}^{(l-1)}; \mathbf{x}_{TA}^{(l)}; \mathbf{F}_{\mathcal{T}}^{MA}\right] \in \mathbb{R}^{T\times (N+3)\times D}$, where N denotes the total number of patches. However, the corresponding text labels for the videos are unknown for the query set branch, so we can only concatenate the input video features $\mathbf{z}^{(l-1)}$ and temporal adapted features $\mathbf{x}_{TA}^{(l)}$ to obtain $\mathbf{z}_{STA\text{-}Q}^{(l-1)} = \left[\mathbf{z}^{(l-1)}; \mathbf{x}_{TA}^{(l)}\right] \in \mathbb{R}^{T\times (N+2)\times D}$. Then, we feed $\mathbf{z}_{MA\text{-}S}^{(l-1)}$ into Local Multimodal Adaptation to fuse spatiotemporal information with text semantic information as illustrated in Fig. 3(c), given by:

$$\mathbf{z}_{MA\text{-}S}^{(l)} = \mathbf{z}_{MA\text{-}S}^{(l-1)} + \text{Adapter}\left(\text{LM-MSA}\left(\text{LN}\left(\mathbf{z}_{MA\text{-}S}^{(l-1)}\right)\right)\right) \tag{7}$$

where $\mathbf{z}_{MA\text{-}S}^{(l-1)}$ and $\mathbf{z}_{MA\text{-}S}^{(l)}$ denote the Local Multimodal Adaptation input and output of the lth ViT block. Self-attention LM-MSA operates on the spatial dimension, where multimodal tokens have been merged, to perform local spatiotemporal modeling and enhance attention to action-related semantics. The Local Spatiotemporal and Multimodal Adaptation share weight parameters, allowing query and support samples to be in the same feature space.

3.3.3. Joint adaptation (JA)

Lastly, we introduce Joint Adaptation, in which an Adapter is parallel to the MLP layer to tune the final representations jointly. Specifically, to ensure the consistency of each layer of the transformer block in the spatial dimension, we perform the Select operation on $\mathbf{z}_{MA\text{-}S}^{(l)}$ and $\mathbf{z}_{STA\text{-}Q}^{(l)}$, taking the leading $N+1$ tokens (class token + critical spatial patch tokens) in the spatial dimension of them. Joint adaptation can be computed as follows:

$$\mathbf{z}^{(l)} = \begin{cases} \mathbf{z}_{MA\text{-}S}^{(l)} + \text{MLP}\left(\text{LN}\left(\mathbf{z}_{MA\text{-}S}^{(l)}\right)\right) + \\ \qquad r \cdot \text{Adapter}\left(\text{LN}\left(\mathbf{z}_{MA\text{-}S}^{(l)}\right)\right) & if\ \ i = 0 \\ \mathbf{z}_{STA\text{-}Q}^{(l)} + \text{MLP}\left(\text{LN}\left(\mathbf{z}_{STA\text{-}Q}^{(l)}\right)\right) + \\ \qquad r \cdot \text{Adapter}\left(\text{LN}\left(\mathbf{z}_{STA\text{-}Q}^{(l)}\right)\right) & if\ \ i = 1 \end{cases} \tag{8}$$

where $i = 0$ and $i = 1$ refer to the support and query set branches, respectively. In this context, r is a scaling factor that regulates the influence of the Adapter's output weight.

3.4. Text-guided prototype construction module (TPCM)

In few-shot action recognition (FSAR), the quality of class prototype construction directly affects the performance of class prototype matching. Previous methods [1–4,20,38] focus on using limited video features to construct class prototypes, which can easily lead to confusion among prototypes of similar categories. Therefore, we design a Text-guided

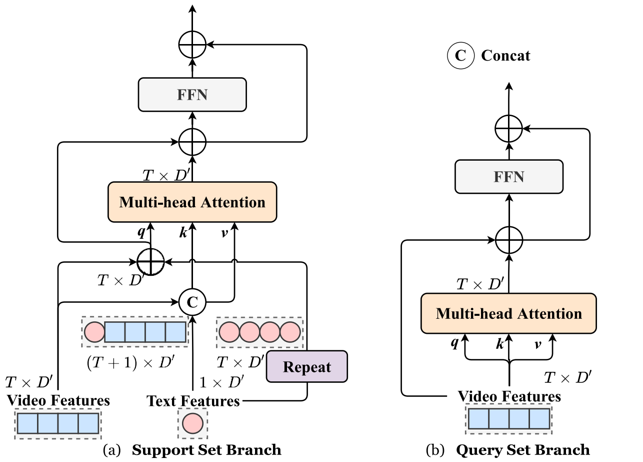


Fig. 4. (a) and (b) respectively show the structure of the TPCM module for the support set and query set branch. ⊕ denotes element-wise summation.

Prototype Construction Module (TPCM) to enrich temporal and semantic representations at the prototype level by fully leveraging the powerful multimodal capabilities of CLIP. Specifically, for the support set branch, given the adapted features from FgMA $\mathbf{F}^m_{\mathcal{S}} \in \mathbf{F}_{\mathcal{S}}$ and the corresponding text features $\mathbf{F}^m_{\mathcal{T}} \in \mathbf{F}_{\mathcal{T}}$, we apply the cross-attention between them to utilize text features for guiding the construction of support class prototypes, resulting in $\widetilde{\mathbf{F}^m_{\mathcal{S}}} \in \widetilde{\mathbf{F}_{\mathcal{S}}}$. The process of obtaining the query-key–value triplets $\mathbf{q}^m_{\mathcal{S}}$, $\mathbf{k}^m_{\mathcal{S}}$, $\mathbf{v}^m_{\mathcal{S}}$ can be described as follows:

$$\mathbf{q}^m_{\mathcal{S}} = \mathbf{F}^m_{\mathcal{S}} + \mathrm{Repeat}\left(\mathbf{F}^m_{\mathcal{T}}\right) \tag{9}$$

$$\mathbf{k}^m_{\mathcal{S}} = \mathbf{v}^m_{\mathcal{S}} = \mathrm{Concat}\left(\left[\mathbf{F}^m_{\mathcal{S}}; \mathbf{F}^m_{\mathcal{T}}\right]\right) \tag{10}$$

where $\mathbf{F}^m_{\mathcal{S}} \in \mathbb{R}^{T \times D'}$, $\mathbf{F}^m_{\mathcal{T}} \in \mathbb{R}^{1 \times D'}$, $\mathbf{q}^m_{\mathcal{S}} \in \mathbb{R}^{T \times D'}$, $\mathbf{k}^m_{\mathcal{S}} = \mathbf{v}^m_{\mathcal{S}} \in \mathbb{R}^{(T+1) \times D'}$, and Repeat aims to copy $\mathbf{F}^m_{\mathcal{T}}$ T times. Then, we apply the multi-head attention (MHA) and a feed-forward network (FFN) to obtain the enhanced support class prototype $\widetilde{\mathbf{F}^m_{\mathcal{S}}} \in \mathbb{R}^{T \times D'}$ as shown in Fig. 4(a), given by:

$$\bar{\mathbf{F}}^m_{\mathcal{S}} = \mathbf{q}^m_{\mathcal{S}} + \mathrm{MHA}\left(\mathbf{q}^m_{\mathcal{S}}, \mathbf{k}^m_{\mathcal{S}}, \mathbf{v}^m_{\mathcal{S}}\right) \tag{11}$$

$$\widetilde{\mathbf{F}^m_{\mathcal{S}}} = \bar{\mathbf{F}}^m_{\mathcal{S}} + \mathrm{FFN}\left(\bar{\mathbf{F}}^m_{\mathcal{S}}\right) \tag{12}$$

Similarly, we perform the same operation on the query set videos to achieve the temporal enhancement at the prototype level, as shown in Fig. 4(b). However, the difference is that $\mathbf{q}^m_{\mathcal{Q}} = \mathbf{k}^m_{\mathcal{Q}} = \mathbf{v}^m_{\mathcal{Q}} = \mathbf{F}^m_{\mathcal{Q}} \in \mathbb{R}^{T \times D'}$ since it does not have corresponding textual features. Note that the support and query set branches share the parameter weights of all modules to reduce computation costs while ensuring that query and support samples are in the same feature space. By incorporating the guidance of powerful multimodal information in constructing class prototypes, we can optimize intra-class and inter-class correlations of video features.

3.5. Metric loss and predictions

Existing few-shot action recognition works [1–4,20,39], relying solely on visual information, classify a query video by comparing the temporally-aligned distances between the query video and the support set prototypes. The advent of the visual-language pre-training model CLIP enables query videos to be classified by matching not only with the prototypes of the support set (visual branch) but also with the corresponding text features of the support set (text branch), as shown in Fig. 2. For the visual branch, given the enhanced class support prototype $\widetilde{\mathbf{F}^m_{\mathcal{S}}} \in \widetilde{\mathbf{F}_{\mathcal{S}}}$ and the query enhanced feature $\widetilde{\mathbf{F}_q} \in \widetilde{\mathbf{F}_{\mathcal{Q}}}$, the distance $D_{q,\mathcal{S}^m}$ can be calculated as:

$$D_{q,\mathcal{S}^m} = \mathcal{M}\left(\widetilde{\mathbf{F}_q}, \widetilde{\mathbf{F}^m_{\mathcal{S}}}\right) \tag{13}$$

where $\mathcal{M}$ denotes the temporal alignment metric, and $D_{q,\mathcal{S}^m} \in D_{q,\mathcal{S}}$. Based on the distances $D_{q,\mathcal{S}}$, we can obtain the probability distribution over support classes $\mathbf{p}_{\mathcal{Q}2\mathcal{S}}$ and use a standard cross-entropy loss $\mathcal{L}_{\mathcal{Q}2\mathcal{S}}$ to optimize the model parameters. For the text branch, given the adapted support set prototype feature $\mathbf{F}^m_{\mathcal{S}} \in \mathbf{F}_{\mathcal{S}}$, adapted query feature $\mathbf{F}_q \in \mathbf{F}_{\mathcal{Q}}$, and corresponding text feature $\mathbf{F}^m_{\mathcal{T}} \in \mathbf{F}_{\mathcal{T}}$, we apply global average pooling on temporal dimension to the features $\mathbf{F}^m_{\mathcal{S}}$ and $\mathbf{F}_q$ to obtain $\mathbf{F}^{m\text{-}avg}_{\mathcal{S}}$ and $\mathbf{F}^{avg}_q$. To bring the pairwise representations of videos and labels closer to each other, we define symmetric similarities between the two modalities using cosine distances in the similarity calculation module, given by:

$$s\left(\mathbf{F}^{m\text{-}avg}_{\mathcal{S}}, \mathbf{F}^m_{\mathcal{T}}\right) = \frac{\left\langle \mathbf{F}^{m\text{-}avg}_{\mathcal{S}}, \mathbf{F}^m_{\mathcal{T}} \right\rangle}{\left\|\mathbf{F}^{m\text{-}avg}_{\mathcal{S}}\right\| \left\|\mathbf{F}^m_{\mathcal{T}}\right\|} \tag{14}$$

$$s\left(\mathbf{F}^{avg}_q, \mathbf{F}^m_{\mathcal{T}}\right) = \frac{\left\langle \mathbf{F}^{avg}_q, \mathbf{F}^m_{\mathcal{T}} \right\rangle}{\left\|\mathbf{F}^{avg}_q\right\| \left\|\mathbf{F}^m_{\mathcal{T}}\right\|} \tag{15}$$

where $s\left(\mathbf{F}^{m\text{-}avg}_{\mathcal{S}}, \mathbf{F}^m_{\mathcal{T}}\right) \in s\left(\mathbf{F}^{avg}_{\mathcal{S}}, \mathbf{F}_{\mathcal{T}}\right)$ and $s\left(\mathbf{F}^{avg}_q, \mathbf{F}^m_{\mathcal{T}}\right) \in s\left(\mathbf{F}^{avg}_{\mathcal{Q}}, \mathbf{F}_{\mathcal{T}}\right)$. Based on the cosine similarities $s\left(\mathbf{F}^{avg}_{\mathcal{S}}, \mathbf{F}_{\mathcal{T}}\right)$ and $s\left(\mathbf{F}^{avg}_{\mathcal{Q}}, \mathbf{F}_{\mathcal{T}}\right)$, we can obtain the softmax-normalized video-to-text similarity scores $\mathbf{p}_{\mathcal{S}2\mathcal{T}}$ and $\mathbf{p}_{\mathcal{Q}2\mathcal{T}}$. Inspired by ActionCLIP [9], we define the Kullback–Leibler (KL) divergence as the video–text contrastive loss $\mathcal{L}_{\mathcal{S}2\mathcal{T}}$ and $\mathcal{L}_{\mathcal{Q}2\mathcal{T}}$. By optimizing contrastive loss, the CLIP model can be adapted to our FSAR task. Finally, we combine the losses from both the visual and textual branches, and integrate the query set video prediction distributions from both branches, as given by:

$$\mathcal{L} = \alpha \cdot \frac{1}{2}\left(\mathcal{L}_{\mathcal{S}2\mathcal{T}} + \mathcal{L}_{\mathcal{Q}2\mathcal{T}}\right) + (1-\alpha) \cdot \mathcal{L}_{\mathcal{Q}2\mathcal{S}} \tag{16}$$

$$\mathbf{p} = \alpha \cdot \mathbf{p}_{\mathcal{Q}2\mathcal{T}} + (1-\alpha) \cdot \mathbf{p}_{\mathcal{Q}2\mathcal{S}} \tag{17}$$

where $\alpha \in [0,1]$ is an adjustable hyperparameter.

4. Experiments

4.1. Experimental setup

4.1.1. Datasets

Our method's performance is assessed on five datasets that can be classified into two categories: (1) spatial-related datasets, including Kinetics [40], HMDB51 [41], and UCF101 [42]. (2) temporal-related datasets, including SSv2-Full [43] and SSv2-Small [43]. For the former, action recognition primarily relies on background information, with temporal information playing a minor role. In contrast, for the latter, the key to action recognition lies in effective temporal modeling. Referring to the previous setups [1,2,44] on Kinetics, SSv2-Full, SSv2-Small, we select 100 classes and divide them into 64/12/24 action classes as training/validation/testing classes. For UCF101 and HMDB51, we evaluate our method on the splits provided by [3].

4.1.2. Network architectures

We choose CLIP [6] as our backbone for efficient fine-tuning, where the visual encoder is ViT-B/32 or ViT-B/16, and the text encoder is a 12-layer, 512-wide transformer with eight attention heads. However, due to the previous works [1–4,20] using ResNet-50 [45] pre-trained on ImageNet [46] as the backbone, we provide a version of utilizing pre-trained CLIP ResNet50 without the FgMA module as our visual encoder. Meanwhile, we set the bottleneck ratio of Adapters to 0.25 in the FgMA module (Section 3.3), the same as AIM [14]. And, we set the scaling factor r to 0.5 on Joint Adaptation (Section 3.3.3). As for the

Table 1
Comparison under 5-way k-shot settings on spatial-related benchmarks including HMDB51, UCF101, and Kinetics. "Average" represents the mean accuracy scores across three spatial-related datasets. The **boldfacen** and underline font indicate the highest and the second highest results.

Method	Reference	Pre-training	Fine-tuning	HMDB51		UCF101		Kinetics		Average	
				1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CMN++ [2]	ECCV(18)	INet-RN50	Full	–	–	–	–	57.3	76.0	-	-
OTAM [1]	CVPR(20)	INet-RN50	Full	54.5	68.0	79.9	88.9	73.0	85.8	69.1	80.9
TRX [4]	CVPR(21)	INet-RN50	Full	54.9[a]	75.6	81.0[a]	96.1	65.1[a]	85.9	67.0	85.8
STRM [20]	CVPR(22)	INet-RN50	Full	57.6[a]	77.3	82.7[a]	96.9	65.1[a]	86.7	68.5	87.0
HyRSM [5]	CVPR(22)	INet-RN50	Full	60.3	76.0	83.9	94.7	73.7	86.1	72.6	85.6
HCL [48]	ECCV(22)	INet-RN50	Full	59.1	76.3	82.5	93.9	73.7	85.8	71.8	85.3
Huang *etal*. [38]	ECCV(22)	INet-RN50	Full	60.1	77.0	71.4	91.0	73.3	86.4	68.2	84.8
Nguyen *etal*. [49]	ECCV(22)	INet-RN50	Full	59.6	76.9	84.9	95.9	74.3	87.4	72.9	86.7
SloshNet [39]	AAAI(23)	INet-RN50	Full	59.4	77.5	86.0	97.1	70.4	87.0	71.9	87.2
MoLo (OTAM) [21]	CVPR(23)	INet-RN50	Full	59.8	76.1	85.4	95.1	73.8	85.1	73.0	85.4
GgHM [50]	ICCV(23)	INet-RN50	Full	61.2	76.9	85.2	96.3	74.9	87.4	73.8	86.9
CLIP-FSAR [12]	IJCV(23)	CLIP-RN50	Full	69.4	80.7	92.4	97.0	90.1	92.0	84.0	89.9
CLIP-FSAR [12]	IJCV(23)	CLIP-ViT-B/16	Full	77.1	87.7	97.0	99.1	94.8	95.4	89.6	94.1
AIM (OTAM) [14]	ICLR(23)	CLIP-ViT-B/32	PEFT	74.7[a]	83.1[a]	92.3[a]	96.8[a]	89.0[a]	92.8[a]	85.3	90.9
AIM (OTAM) [14]	ICLR(23)	CLIP-ViT-B/16	PEFT	76.2[a]	86.9[a]	95.4[a]	98.2[a]	90.5[a]	95.3[a]	87.4	93.5
MVP-shot [13]	Arxiv(24)	CLIP-RN50	Full	72.5	82.5	92.2	97.6	90.0	93.2	84.9	91.1
MVP-shot [13]	Arxiv(24)	CLIP-ViT-B/16	Full	77.0	**88.1**	96.8	99.0	91.0	95.1	88.3	94.1
CLIP-CPM^{2}C [32]	Neurocomputing(25)	CLIP-RN50	Full	66.3	81.2	91.1	97.4	88.6	93.1	82.0	90.6
CLIP-CPM^{2}C [32]	Neurocomputing(25)	CLIP-ViT-B/16	Full	75.9	88.0	95.0	98.6	91.0	95.5	87.3	94.0
MA-FSAR	–	CLIP-RN50	Full	73.3	82.1	92.8	97.2	92.8	93.0	86.3	90.8
MA-FSAR	–	CLIP-ViT-B/32	PEFT	77.3	83.9	95.0	98.7	93.5	94.3	88.6	92.3
MA-FSAR	–	CLIP-ViT-B/16	PEFT	**83.4**	87.9	**97.2**	**99.2**	**95.7**	**96.0**	**92.1**	**94.4**

[a] means our implementation.

Note: For Fine-tuning, "Full" indicates the full-parameter fine-tuning of the visual encoder, and "PEFT" indicates the parameter-efficient fine-tuning of the visual encoder. Methods with the CLIP pre-training all use textual input from the support set videos except AIM [14].

adjustable hyperparameter α that controls metric loss and predictions (Section 3.5), we set α to 0.75 for spatial-related datasets and 0.25 for temporal-related datasets. For the prompt templates of the text encoder, we discussed in Section 4.3.8. In training, a prompt template is randomly selected from 16 candidate templates for each video. The vector is obtained during inference by utilizing all 16 prompt templates as inputs and taking their average. For the temporal alignment metric $\mathcal{M}$, we choose OTAM [1] as our matching metric.

4.1.3. Training and inference

Following TSN [37], we uniformly select 8 frames (T=8) of a video as the input augmented with some fundamental techniques, such as random horizontal flipping, cropping, and color jitter in training, while only center crop in inference. For training, SSv2-Full and SSv2-Small randomly sample 100,000 training episodes, and the other datasets randomly sample 10,000 training episodes. Meanwhile, we freeze the pre-trained CLIP and only fine-tune lightweight adapters during the training process when the visual encoder is ViT. If the visual encoder is ResNet-50, we only freeze the text encoder and fully fine-tune the visual encoder. Moreover, our framework uses the Adam [47] optimizer with the multi-step scheduler. As for inference, the average results of 10,000 tasks randomly sampled from the test sets in all datasets are reported in our experiments.

4.2. Results

In this work, we chose OTAM [1] as our temporal alignment metric and maintained this default setting in all subsequent experiments. Methods with the CLIP pre-training except for AIM [14] all use textual input from the support set videos. Our approach reports results using three different visual encoders. The CLIP-RN50 model has a fully fine-tuned visual encoder since it does not have an Adapter structure. In contrast, the two ViT-B models only fine-tune lightweight adapters during the training process. Furthermore, AIM [14] serves as the video classification framework, where we replace its classification head with our same matching head (OTAM) for a fair comparison.

4.2.1. Results on spatial-related datasets

For spatial-related datasets, action recognition primarily depends on background information, with temporal modeling playing a supplementary role. CLIP is the large foundation image pre-trained model that mainly relies on background information to recognize images. Therefore, fine-tuning CLIP on spatial-related datasets will result in a significant improvement in few-shot action recognition. As shown in Table 1, even our CLIP-RN50 model significantly improves accuracy in any task setting compared to excellent methods (such as TRX [4], STRM [20], HyRSM [5], SloshNet [39], MoLo [21], et al.) that use ImageNet pre-training. Compared to CLIP-FSAR [12], MVP-shot [13], CLIP-CPM^{2}C [32], and AIM [14], which uses the same CLIP pre-training, our MA-FASR achieves better results across almost all datasets and task settings, except in the 5-way 5-shot setting on the HMDB51 dataset. This is due to the smaller scale of the HMDB51 dataset, where the 5-shot task provides more data than the 1-shot task, allowing full-parameter fine-tuning methods like CLIP-CPM^{2}C and MVP-shot to potentially overfit and perform slightly better. However, in more challenging settings, such as the HMDB51 1-shot task and larger datasets, our MA-FSAR shows significant improvements in performance benefits in integrating text features at the token level in our FgMA. Additionally, we also report the average accuracy scores for spatial-related datasets, which better reflect the robustness and superiority of our MA-FSAR.

4.2.2. Results on temporal-related datasets

For temporal-related datasets, the key to action recognition is temporal relationship understanding. The performance improvement from CLIP's strong pre-training is less significant than those for spatial-related datasets. However, our model continues to demonstrate excellent results, attributed to our comprehensive global and local temporal modeling within FgMA at the token level and our temporal relationship enhancement at the prototype level. We report three model results using different visual encoders as shown in Table 2. Compared to the baseline OTAM [1], our MA-FSAR using CLIP-RN50 as the visual encoder can bring 16.1%, 16.0% performance improvements in the 1-shot task, and 9.8%, 11.3% accuracy gains in the 5-shot task of SSv2-Small and SSv2-Full, respectively. Meanwhile, our CLIP-RN50 model outperforms all other methods using ResNet-50 as the visual encoder

Table 2
Comparison under 5-way k-shot settings on temporal-related benchmarks including SSv2-Small, and SSv2-Full. "Average" represents the mean accuracy scores across two temporal-related datasets. The **boldfacen** and underline font indicate the highest and the second highest results.

Method	Reference	Pre-training	Fine-tuning	SSv2-Small		SSv2-Full		Average	
				1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
CMN++ [2]	ECCV(18)	INet-RN50	Full	34.4	43.8	36.2	48.8	35.3	46.3
OTAM [1]	CVPR(20)	INet-RN50	Full	36.4	48.0	42.8	52.3	39.6	50.2
TRX [4]	CVPR(21)	INet-RN50	Full	36.0[a]	56.7[a]	42.0[a]	64.6	39.0	60.7
STRM [20]	CVPR(22)	INet-RN50	Full	37.1[a]	55.3[a]	43.1[a]	68.1	40.1	61.7
HyRSM [5]	CVPR(22)	INet-RN50	Full	40.6	56.1	54.3	69.0	47.5	62.6
HCL [48]	ECCV(22)	INet-RN50	Full	38.7	55.4	47.3	64.9	43.0	60.2
Huang *etal.* [38]	ECCV(22)	INet-RN50	Full	38.9	61.6	49.3	66.7	44.1	64.2
Nguyen *etal.* [49]	ECCV(22)	INet-RN50	Full	–	–	43.8	61.1	-	-
SloshNet [39]	AAAI(23)	INet-RN50	Full	–	–	46.5	68.3	-	-
MoLo (OTAM) [21]	CVPR(23)	INet-RN50	Full	41.9	56.2	55.0	69.6	48.5	62.9
GgHM [50]	ICCV(23)	INet-RN50	Full	–	–	54.5	69.2	-	-
CLIP-FSAR [12]	IJCV(23)	CLIP-RN50	Full	52.1	55.8	58.7	62.8	55.4	59.3
CLIP-FSAR [12]	IJCV(23)	CLIP-ViT-B/16	Full	54.6	61.8	62.1	72.1	58.4	67.0
AIM (OTAM) [14]	ICLR(23)	CLIP-ViT-B/32	PEFT	52.0[a]	61.6[a]	61.4[a]	63.9[a]	56.7	62.8
AIM (OTAM) [14]	ICLR(23)	CLIP-ViT-B/16	PEFT	53.9[a]	63.5[a]	62.7[a]	72.1[a]	58.3	67.8
MVP-shot [13]	Arxiv(24)	CLIP-RN50	Full	51.2	57.0	–	–	-	-
MVP-shot [13]	Arxiv(24)	CLIP-ViT-B/16	Full	55.4	62.0	–	–	-	-
CLIP-CPM^{2}C [32]	Neurocomputing(25)	CLIP-RN50	Full	51.5	57.1	58.0	64.0	54.8	60.6
CLIP-CPM^{2}C [32]	Neurocomputing(25)	CLIP-ViT-B/16	Full	52.3	62.6	62.1	**72.8**	57.2	67.7
MA-FSAR	–	CLIP-RN50	Full	52.5	57.8	58.8	63.6	55.7	60.7
MA-FASR	–	CLIP-ViT-B/32	PEFT	56.5	62.3	61.9	64.5	59.2	63.4
MA-FSAR	–	CLIP-ViT-B/16	PEFT	**59.1**	**64.5**	**63.3**	72.3	**61.2**	**68.4**

[a] means our implementation.
Note: For Fine-tuning, "Full" indicates the full-parameter fine-tuning of the visual encoder, and "PEFT" indicates the parameter-efficient fine-tuning of the visual encoder. Methods with the CLIP pre-training all use textual input from the support set videos except AIM [14].

in 1-shot tasks across all temporal-related datasets, highlighting the effectiveness of our TPCM module design. Compared to CLIP-FSAR [12], MVP-shot [13], CLIP-CPM^{2}C [32], and AIM [14], which uses the same CLIP pre-training and temporal alignment metric, our method has a significant performance improvement, especially in 1-shot tasks. For the SSv2-Small dataset, even our ViT-B/32 model can perform better than any CLIP-based ViT-B/16 model, demonstrating the superiority of our FgMA design. Meanwhile, the reported average accuracy scores for temporal-related datasets, our MA-FSAR also achieves the SOTA performance.

4.3. Ablation study

we employ CLIP-ViT-B/32 as our visual encoder, as the default setting for subsequent ablation studies. Ablation experiments 4.3.1, 4.3.2, 4.3.3, 4.3.4, 4.3.5, 4.3.6, and 4.3.7 demonstrate the effectiveness of our module design; 4.3.8 explores the impact of different prompt templates on our MA-FSAR; 4.3.9 presents the zero-shot performance; 4.3.10 discuss the hyperparameters in our model; 4.3.11 provides the cross-validation results of our MA-FSAR; 4.3.12 highlight our MA-FSAR's training efficiency; 4.3.13 provides visualizations of attention maps.

4.3.1. Impact of the proposed components

To assess the impact of each module (*i.e.*, FgMA and TPCM) in our method, we conduct experiments under 5-way 1-shot settings on SSv2-Small and SSv2-Full. As indicated in Table 3, we confirm the effectiveness of each component. The multimodal baseline freezes all the learnable weights without extra modules. Specifically, in comparison to the baseline, the FgMA module results in accuracy improvements of 13.5% and 16.3% on SSv2-Small and SSv2-Full, respectively, while the TPCM module yields gains of 16.9% and 19.6% on the two datasets. The combination of all modules produces the most favorable outcomes, leading to accuracy gains of 27.7% and 31.7% on SSv2-Small and SSv2-Full, respectively, over the baseline.

Table 3
The impact of proposed modules on SSv2-Small and SSv2-Full in the 5-way 1-shot task.

FgMA	TPCM	SSv2-Small	SSv2-Full
✗	✗	28.8	30.2
✗	✓	42.3	46.5
✓	✗	45.7	49.8
✓	✓	56.5	61.9

Table 4
Effectiveness of the adapter components in the SSv2-Small 5-way 1-shot task. LMA/LSTA, GTA, and JA indicate the local multimodal/spatiotemporal adaptation, global temporal adaptation, and joint adaptation.

Method	Params	Tunable Params	Acc
Frozen	154.4M	3.2M	42.3
Fine-tuned visual-only	154.4M	91.0M	53.9
Frozen + LMA/LSTA	159.2M	7.9M	54.0
+ GTA	166.3M	15.1M	55.9
+ JA	169.8M	18.5M	56.5

4.3.2. Effectiveness of the adaptation components

To demonstrate the effectiveness of our proposed multi-type adaptation in FgMA, we compare our method to two baselines. The first baseline represents a frozen space-only model without any adaptation, wherein all the trainable parameters of CLIP encoders are frozen, but the TPCM module is not included. The second baseline involves fully fine-tuning the visual encoder without any adaptation. As presented in Table 4, the fine-tuned visual-only model demonstrates a 12.4% performance improvement over the first baseline but with an increase in tunable parameters from 3.15M to 90.99M. Our approach seeks to introduce lightweight adapters into a fully frozen visual model, maintaining the integrity of the pre-trained weights, to achieve superior performance compared to a fully fine-tuned model. In Table 4, after the Local Multimodal/Spatiotemporal Adaptation, the frozen model achieves a comparable performance with the full fine-tuned visual-only model (54.0 *vs.* 53.9), while utilizing less than one-tenth of the parameter count of the latter (7.94M *vs.* 90.99M). Note that since there

Table 5
Effectiveness comparison between local multimodal and spatiotemporal adaptation (LMA/LSTA) in 5-way 1-shot task.

Method	Dataset	Acc
Double LSTA	SSv2-Small	55.9
LSTA and LMA	SSv2-Small	**56.5**
Double LSTA	SSv2-Full	61.2
LSTA and LMA	SSv2-Full	**61.9**

Table 6
Comparisons of different prototype construction methods.

Method	Visual encoder	Acc
Unimodal Transformer	CLIP-ViT-B/32	47.9
Multimodal Transformer	CLIP-ViT-B/32	54.7
TPCM	CLIP-ViT-B/32	**56.5**

is no injection of the global temporal token, this adaptation cannot perform temporal modeling. Upon adding the Global Temporal and Joint Adaptation, they yield 1.9% and 0.6% performance improvements. Our final model delivers a 2.6% accuracy improvement compared to the fine-tuned visual-only model, yet with only one-fifth of the tunable parameters.

4.3.3. Comparison between local multimodal and spatiotemporal adaptation

To ensure a fair comparison of Local Multimodal and Spatiotemporal Adaptation (LMA/LSTA), we conduct experiments on the 5-way 1-shot task of SSv2-Small and SSv2-Full. As outlined in Section 3.3.2, the distinction between the two adaptations lies in the inclusion or exclusion of text features for self-attention with spatiotemporal features in support videos. The results presented in Table 5 demonstrate that using LMA instead of LSTA leads to performance improvements of 0.6% and 0.7% on SSv2-Small and SSv2-Full, respectively. These findings illustrate the efficacy of enhancing the semantic representation of visual tokens by integrating textual tokens within the Adapter (Section 3.3).

4.3.4. Comparisons of different prototype construction methods

To demonstrate the effectiveness of our proposed module and to compare the performance of various methods for prototype construction, we conduct experiments in the SSv2-Small 5-way 1-shot task. The first baseline unimodal transformer indicates the features $\mathbf{F}_{\mathcal{S}}$ and $\mathbf{F}_{\mathcal{Q}}$ performing self-attention on the temporal dimension. Contrasting this, the second baseline (CLIP-FSAR [12]) differs in that the text features $\mathbf{F}_{\mathcal{T}}$ are stacked along the temporal dimension before self-attention is conducted on the support features $\mathbf{F}_{\mathcal{S}}$. We set all the layers of the transformer to be one. As shown in Table 6, our TPCM module brings 8.6% and 1.8% performance improvements compared to the unimodal transformer and multimodal transformer on SSv2-Small, respectively. These experimental results underscore the TPCM module's considerable efficacy in leveraging textual information as guidance to integrate visual and textual features at the prototype level effectively, resulting in the creation of more robust class prototype representations.

4.3.5. MA-FSAR effectiveness on different temporal alignment metrics

We conduct the experiments using different temporal alignment metrics on the 5-way 1-shot task of Kinetics and SSv2-Small to demonstrate that our model is plug-and-play, capable of being integrated with any common matching metric. We adopt three different temporal alignment metrics, including OTAM [1], TRX [4], and Bi-MHM [5]. As displayed in Table 7, our method showcases its ability to adapt to any temporal alignment metric, with the final accuracies closely correlated to the performance of the chosen metric. Moreover, regardless of the temporal alignment metric employed, our MA-FSAR consistently achieves the most outstanding performance compared to the baselines, providing compelling evidence for the superiority of our model.

4.3.6. Unimodal model vs. Multimodal model

To compare the performance of the unimodal and multimodal models, we experiment with various pre-training and model modalities in the 5-way 1-shot task on Kinetics and SSv2-Small, evaluating them on multiple matching metrics. We provide two baselines for each metric: an ImageNet [46] pre-trained unimodal model and a CLIP pre-trained unimodal model, with all baseline models' visual encoders fully fine-tuned. As depicted in Table 7, using a CLIP pre-trained unimodal model shows some performance improvements compared to the ImageNet pre-trained model, albeit relatively limited. However, when using our proposed MA-FSAR multimodal model, a significant performance improvement is observed in both datasets. Specifically, our MA-FSAR consistently achieves a minimum accuracy improvement of 15% over the ImageNet pre-trained unimodal model and 10% over the CLIP pre-trained unimodal model on two datasets. These results not only highlight the significance of text features for few-shot action recognition but also prove the effectiveness of our method.

4.3.7. Full fine-tuning vs. Adaptation

We conduct experiments in the SSv2-Small 5-way 1-shot task to make a fair comparison between full fine-tuning and adaptation, which demonstrates the effectiveness of the FgMA module we propose. As shown in Table 8, our adaptation method leads to accuracy improvements of 2.6% and 2.8% on SSv2-Small and SSv2-Full, respectively, over the full fine-tuning model. Our adaptation method implements multimodal fusion and temporal modeling, while the full fine-tuning method does not achieve this. However, our method has only one-fifth (18.54M *vs.* 90.99M) of tunable parameters compared to the full fine-tuning method, which requires 1.6G (11.9G *vs.* 13.5G) less memory usage, and takes 0.4 h (3.0H vs. 3.4H) less time to train for 10,000 tasks on a single RTX3090. The experimental results highlight that our MA-FSAR is fast, efficient, and has low training costs.

4.3.8. Comparison of different prompt templates

To explore the impact of different prompt templates on recognition performance, we conduct 5-way 1-shot experiments on Kinetics and SSv2-Small. Here, "`[CLS]`" refers to no prompt template, and "`a photo of action [CLS]`" is a commonly used prompt template. As illustrated in Table 9, different datasets exhibit distinct preferences for prompt templates. For example, SSv2-Small performs better using "`[CLS]`", while Kinetics shows the opposite trend. To address this, we introduce a mixed prompt template, which involves utilizing 16 different prompt templates and then summing and averaging the outputs of 16 textual tokens during inference. The list of these 16 different prompt templates is shown in Table 10. In Table 9, our mixture method balances the preferences of various datasets for prompt templates, achieving optimal performance on each dataset.

4.3.9. Zero-shot performance

To investigate the zero-shot performance of the CLIP model, we conduct 5-way zero-shot experiments on the spatial-related datasets (Kinetics, HMDB51, and UCF101) and the temporal-related dataset (SSv2-Small). Comparative experiments are set up for four mainstream methods, and the results are detailed in Table 11. In our experiments, CLIP-Freeze refers to using only the original CLIP model and its pre-trained models. CLIP-FSAR [12], AIM [14], and our MA-FSAR only use the text-to-image matcher, which fine-tunes the training sets and performs zero-shot recognition on the testing sets. We find that the original frozen CLIP model demonstrates strong recognition capabilities on spatial-related datasets due to its pre-trained model's robust image background discriminative ability. However, its recognition performance is unsatisfactory with near-random accuracy (20%) of **28.8%** on SSv2-Small, reflecting its lack of capability for temporal modeling. In contrast, our method MA-FSAR exhibits a notable accuracy improvement of 18.8% over the Frozen CLIP model on SSv2-Small, highlighting

Table 7
Method effectiveness on different temporal alignment metrics on SSv2-Small and Kinetics in the 5-way 1-shot task. And effectiveness Comparisons between the Unimodal model and the Multimodal model.

Temporal Alignment Metric	Model Modality	Pre-training	Kinetics	SSv2-Small
OTAM [1]	Unimodal	INet-ViT-B/32	75.8	38.2
OTAM [1]	Unimodal	CLIP-ViT-B/32	83.7	44.8
MA-FSAR (OTAM)	Multimodal	CLIP-ViT-B/32	**93.5**	**56.5**
TRX [4]	Unimodal	INet-ViT-B/32	67.2	37.3
TRX [4]	Unimodal	CLIP-ViT-B/32	82.8	42.7
MA-FSAR (TRX)	Multimodal	CLIP-ViT-B/32	**92.8**	**52.4**
Bi-MHM [5]	Unimodal	INet-ViT-B/32	75.2	39.5
Bi-MHM [5]	Unimodal	CLIP-ViT-B/32	83.2	45.5
MA-FSAR (Bi-MHM)	Multimodal	CLIP-ViT-B/32	**93.2**	**56.9**

Table 8
Effectiveness comparison between full fine-tuning and adaptation on SSv2-Small and SSv2-Full in the 5-way 1-shot task. “Memory (G)” refers to the amount of video memory usage, and “Time (Hours)” indicates the time required to train 10,000 tasks, measured in hours on a single RTX3090.

Method	Dataset	Tunable Param (M)	Memory (G)	Time (Hours)	Acc
Full fine-tuning	SSv2-Small	90.99	13.5	3.4	53.9
Adaptation	SSv2-Small	**18.54**	**11.9**	**3.0**	**56.5**
Full fine-tuning	SSv2-Full	90.99	13.5	3.4	59.1
Adaptation	SSv2-Full	**18.54**	**11.9**	**3.0**	**61.9**

Table 9
Comparison experiments of different prompt templates on SSv2-Small, Kinetics in the 5-way 1-shot task. “Mixture” refers to using 16 different prompt templates, then summing and averaging the outputs of 16 textual tokens during inference.

Prompt template	SSv2-Small	Kinetics
“[CLS]”	56.1	92.0
“a photo of action [CLS]”	55.3	92.7
Mixture	**56.5**	**93.5**

Table 10
The list of 16 different prompt templates.

Prompt template
“[CLS]”
“a photo of action [CLS]”
“a picture of action [CLS]”
“Human action of [CLS]”
“[CLS], an action”
“[CLS] this is an action”
“[CLS], a video of action”
“Playing action of [CLS]”
“Playing a kind of action, [CLS]”
“Doing a kind of action, [CLS]”
“Look, the human is [CLS]”
“Can you recognize the action of [CLS]”
“Video classification of [CLS]”
“A video of [CLS]”
“The man is [CLS]”
“The woman is [CLS]”

Table 11
Comparison of 5-way zero-shot performance with CLIP on spatial-related datasets (Kinetics, HMDB51, and UCF101) and temporal-related dataset (SSv2-Small).

Method	SSv2-Small	Kinetics	HMDB51	UCF101
CLIP-Frezze	28.8	92.8	63.3	87.1
CLIP-FSAR [12]	44.5	92.9	72.1	90.1
AIM [14]	46.7	**93.0**	72.6	88.7
MA-FSAR	**47.1**	**93.0**	**73.2**	**89.9**

our approach’s powerful temporal relation modeling capability. Additionally, our method also shows some performance improvement on spatial-related datasets. Meanwhile, compared to AIM [14] and CLIP-FSAR [12], our MA-FSAR also has the best zero-shot performance on various datasets.

Table 12
Effect of the scaling factor r.

scaling factor r	0	0.25	0.5	0.75	1
Kinetics	93.1	93.3	**93.5**	93.4	93.4
SSv2-Small	55.8	56.2	**56.5**	56.3	56.2

Table 13
Effect of the loss factor α.

loss factor α	0	0.25	0.5	0.75	1
Kinetics	91.7	92.5	92.9	**93.5**	93.0
SSv2-Small	56.2	**56.5**	54.9	52.7	47.1

Table 14
Cross-validation with MA-FSAR (CLIP-ViT-B/32) over 5 rounds under 5-way k-shot settings on Kinetics, and SSv2-Small. “Average" indicates the mean performance across all 5 rounds. “Default” represents the accuracy results with the default division set..

Round	Kinetics		SSv2-Small	
	1-shot	5-shot	1-shot	5-shot
Round 1	93.4	94.2	56.6	62.5
Round 2	93.6	94.3	56.2	62.2
Round 3	93.8	94.4	56.7	62.4
Round 4	93.9	94.5	56.4	62.2
Round 5	93.3	94.2	56.8	62.6
Average	93.6	94.3	56.5	62.4
Default	93.5	94.3	56.5	62.3

4.3.10. Analysis of hyperparameters

We explore the impact of hyperparameters, including scaling factor r (Section 3.3.3) and the loss factor α (Section 3.5). As shown in Table 12, our MA-FSAR delivers optimal performance across various datasets when the scaling factor r in Joint Adaptation is set to 0.5. As illustrated in Table 13, we conduct experiments to analyze the impact of the adjustable hyperparameter α, which controls metric loss and predictions (Sec.řefprediction) on representative spatial-related (Kinetics) and temporal-related (SSv2-Small) datasets. The results reveal that the model achieves the highest accuracy when α is set to 0.75 on Kinetics and 0.25 on SSv2-Small.

4.3.11. Cross-validation of MA-FSAR

We conduct the cross-validation experiments using MA-FSAR (CLIP-ViT-B/32) on the representative spatial-related dataset Kinetics and the temporal-related dataset SSv2-Small, demonstrating the robustness

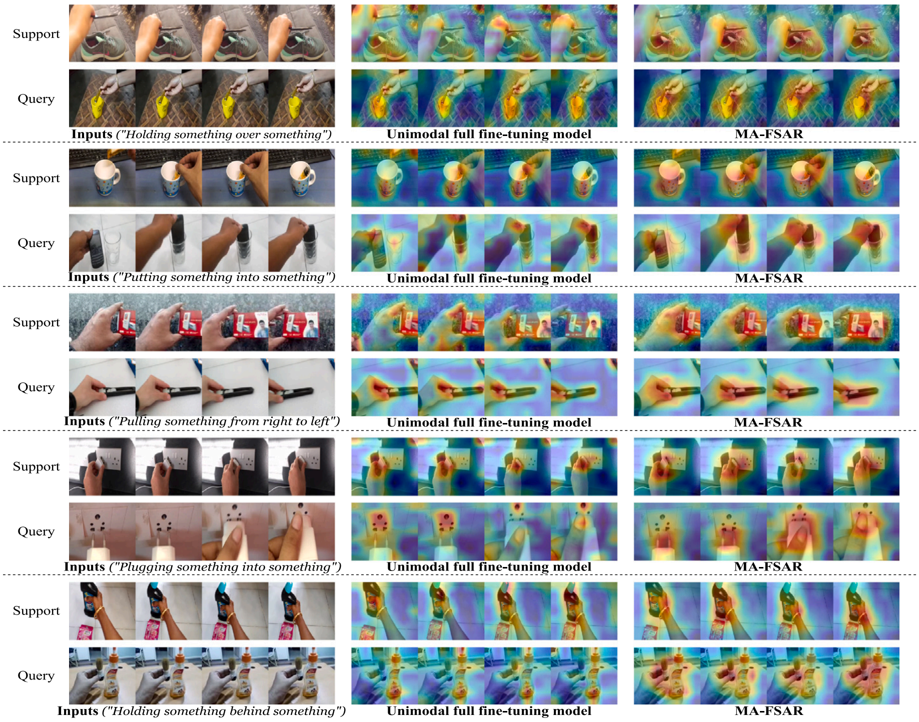

Support
Query
Inputs ("Holding something over something")
Unimodal full fine-tuning model
MA-FSAR
Support
Query
Inputs ("Putting something into something")
Unimodal full fine-tuning model
MA-FSAR
Support
Query
Inputs ("Pulling something from right to left")
Unimodal full fine-tuning model
MA-FSAR
Support
Query
Inputs ("Plugging something into something")
Unimodal full fine-tuning model
MA-FSAR
Support
Query
Inputs ("Holding something behind something")
Unimodal full fine-tuning model
MA-FSAR

Fig. 5. Attention visualization of our MA-FSAR in the SSv2-Small 5-way 1-shot task. Corresponding to the original RGB images(left), the attention maps of the unimodal full fine-tuning model (middle) are compared to the attention maps with our MA-FSAR (right). The temporal alignment metric is OTAM [1].

of our MA-FSAR. Specifically, both Kinetics and SSv2-Small follow the same data structure, with 100 categories divided into 64 action categories for training, 12 action categories for validation, and 24 action categories for testing. Due to the requirement of dividing the 100 categories according to the 64/12/24 rule, it is difficult to implement k-fold cross-validation. Therefore, we randomly split the 100 categories into 64/12/24 for training/validation/testing five times as the cross-validation. This approach, a variant of traditional cross-validation through repeated random splitting, offers the advantage of assessing the model's stability across different category combinations. The experiment results are shown in Table 14. From the table, it can be seen that the performance output of our model is quite stable, with a small difference between the average results of the five rounds of random divisions and the output results of the default division set.

4.3.12. Comparison of model efficiency

The model efficiency results on an RTX3090 GPU for the 5-way 1-shot task are shown in Table 15. Compared to traditional unimodal methods such as OTAM [1] and HyRSM [5], our model, despite having fewer trainable parameters, uses a more complex CLIP-ViT architecture instead of ResNet50, leading to higher GPU memory usage and computational time. However, this increased complexity leads to a significant improvement in accuracy performance. Compared to the multimodal method CLIP-FSAR [12], our MA-FSAR, benefiting from the PEFT training framework, achieves higher recognition accuracy while requiring fewer trainable parameters, less GPU memory, and shorter training time, thereby demonstrating the efficiency of our model.

4.3.13. Attention visualization of MA-FSAR

Fig. 5 shows the attention visualization of our MA-FSAR in the SSv2-Small 5-way 1-shot task. Corresponding to the original RGB images (left), the attention maps of the unimodal full fine-tuning model using CLIP pre-trained weights (middle), which we have mentioned in Section 4.3.6 are compared to the attention maps with our MA-FSAR (right). As illustrated in Fig. 5, the attention maps generated by MA-FSAR prioritize action-related objects and reduce attention to the background and unrelated objects. These observations substantiate the empirical evidence of the efficacy of our MA-FSAR in enhancing semantic and spatiotemporal representation.

Table 15
Comparison of model efficiency for 5-way 1-shot SSv2-Small evaluation. "Train./Full Param." indicates the Trainable/Full Parameters. "Memory" denotes the GPU memory usage. "Time" represents the training time cost of a 5-way 1-shot task. "Acc" means the 5-way 1-shot accuracy on SSv2-Small. All experiments are carried out on one Nvidia RTX3090 GPU..

	Pre-Training	Train./Full Param.	Memory	Time	Acc
OTAM [1]	INet-RN50	23.5 M/23.5 M	7.9G	0.85 s/it	36.4
HyRSM [5]	INet-RN50	65.6 M/65.6 M	8.4G	0.91 s/it	40.6
CLIP-FSAR [12]	CLIP-ViT-B/32	90.2 M/90.2 M	13.2G	1.34 s/it	53.1
MA-FSAR	CLIP-ViT-B/32	18.5 M/91.0 M	11.9G	1.11 s/it	56.5

5. Conclusion

In this work, we propose a novel method, MA-FSAR, to refine CLIP for few-shot action recognition. Our solution, based on the Adapter-based technique of PEFT, incorporates a Fine-grained Multimodal Adaptation (FgMA) tailored for FSAR with the enhancement of the action-related temporal and semantic representations, which is fast, efficient, and cost-effective in training. Specifically, we first introduce a Global Temporal Adaptation that processes only the class token to efficiently capture global motion cues. These outputs are fed into the subsequent Local Multimodal Adaptation to guide the local visual tokens in learning spatiotemporal details. Notably, this module can integrate text features specific to the FSAR support set, emphasizing fine-grained semantics associated with actions. At the prototype level, we propose a Text-guided Prototype Construction Module (TPCM) to further enrich the temporal and semantic characteristics of video prototypes. Extensive experiments across various task settings and five widely used datasets consistently demonstrate our method's excellent performance in any temporal alignment metric, achieved with minimal trainable parameters.

Limitations. While our method demonstrates exceptional performance across all datasets, it remains data-driven, relying on fine-tuning a training set to achieve outstanding results on the testing set. Looking ahead, our goal is to explore improved utilization of Large Language Models (LLM) to develop knowledge-driven few-shot action recognition methods, enhancing the model's generalization capabilities.

CRediT authorship contribution statement

Jiazheng Xing: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Jian Zhao:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation. **Chao Xu:** Writing – review & editing, Visualization, Validation, Supervision, Methodology, Investigation, Formal analysis. **Mengmeng Wang:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Formal analysis. **Guang Dai:** Supervision, Software, Resources, Investigation, Funding acquisition, Data curation. **Yong Liu:** Supervision, Software, Resources, Project administration, Funding acquisition, Data curation, Conceptualization. **Jingdong Wang:** Writing – review & editing, Validation, Supervision, Project administration, Investigation, Conceptualization. **Xuelong Li:** Writing – review & editing, Supervision, Software, Resources, Project administration, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the State Key Laboratory of Industrial Control Technology, China, under grant ICT2024A09.

Data availability

Data will be made available on request.

References

[1] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, J.C. Niebles, Few-shot video classification via temporal alignment, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10618–10627.

[2] L. Zhu, Y. Yang, Compound memory networks for few-shot video classification, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 751–766.

[3] H. Zhang, L. Zhang, X. Qi, H. Li, P.H. Torr, P. Koniusz, Few-shot action recognition with permutation-invariant attention, in: European Conference on Computer Vision, Springer, 2020, pp. 525–542.

[4] T. Perrett, A. Masullo, T. Burghardt, M. Mirmehdi, D. Damen, Temporal-relational crosstransformers for few-shot action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 475–484.

[5] X. Wang, S. Zhang, Z. Qing, M. Tang, Z. Zuo, C. Gao, R. Jin, N. Sang, Hybrid relation guided set matching for few-shot action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19948–19957.

[6] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[7] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, T. Duerig, Scaling up visual and vision-language representation learning with noisy text supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 4904–4916.

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

[9] M. Wang, J. Xing, J. Mei, Y. Liu, Y. Jiang, Actionclip: Adapting language-image pretrained models for video action recognition, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[10] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, X. Wang, Groupvit: Semantic segmentation emerges from text supervision, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18134–18144.

[11] S. Zhao, Z. Zhang, S. Schulter, L. Zhao, B. Vijay Kumar, A. Stathopoulos, M. Chandraker, D.N. Metaxas, Exploiting unlabeled data with vision and language models for object detection, in: European Conference on Computer Vision, Springer, 2022, pp. 159–175.

[12] X. Wang, S. Zhang, J. Cen, C. Gao, Y. Zhang, D. Zhao, N. Sang, CLIP-guided prototype modulating for few-shot action recognition, 2023, arXiv preprint arXiv:2303.02982.

[13] H. Qu, R. Yan, X. Shu, H. Gao, P. Huang, G.-S. Xie, MVP-shot: Multi-velocity progressive-alignment framework for few-shot action recognition, 2024, arXiv preprint arXiv:2405.02077.

[14] T. Yang, Y. Zhu, Y. Xie, A. Zhang, C. Chen, M. Li, Aim: Adapting image models for efficient video action recognition, 2023, arXiv preprint arXiv:2302.03024.

[15] J. Pan, Z. Lin, X. Zhu, J. Shao, H. Li, St-adapter: Parameter-efficient image-to-video transfer learning, Adv. Neural Inf. Process. Syst. 35 (2022) 26462–26477.

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

[17] R. Liu, J. Huang, G. Li, J. Feng, X. Wu, T.H. Li, Revisiting temporal modeling for clip-based image-to-video knowledge transferring, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6555–6564.

[18] S.T. Wasim, M. Naseer, S. Khan, F.S. Khan, M. Shah, Vita-CLIP: Video and text adaptive CLIP via multimodal prompting, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 23034–23044.

[19] C. Ju, T. Han, K. Zheng, Y. Zhang, W. Xie, Prompting visual-language models for efficient video understanding, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV, Springer, 2022, pp. 105–124.

[20] A. Thatipelli, S. Narayan, S. Khan, R.M. Anwer, F.S. Khan, B. Ghanem, Spatio-temporal relation modeling for few-shot action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 19958–19967.

[21] X. Wang, S. Zhang, Z. Qing, C. Gao, Y. Zhang, D. Zhao, N. Sang, Molo: Motion-augmented long-short contrastive learning for few-shot action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 18011–18021.

[22] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.

[23] A. Nichol, J. Schulman, Reptile: a scalable metalearning algorithm, 2, (3) 2018, p. 4, arXiv preprint arXiv:1803.02999.

[24] Z. Li, F. Zhou, F. Chen, H. Li, Meta-sgd: Learning to learn quickly for few-shot learning, 2017, arXiv preprint arXiv:1707.09835.

[25] J. Wang, J. Wu, H. Bai, J. Cheng, M-nas: Meta neural architecture search, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, (04) 2020, pp. 6186–6193.

[26] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, Adv. Neural Inf. Process. Syst. 30 (2017).

[27] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016).

[28] S.W. Yoon, J. Seo, J. Moon, Tapnet: Neural network augmented with task-adaptive projection for few-shot learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 7115–7123.

[29] C. Doersch, A. Gupta, A. Zisserman, Crosstransformers: spatially-aware few-shot transfer, Adv. Neural Inf. Process. Syst. 33 (2020) 21981–21993.

[30] J. Patravali, G. Mittal, Y. Yu, F. Li, M. Chen, Unsupervised few-shot action recognition via action-appearance aligned meta-adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8484–8494.

[31] F. Guo, Y. Wang, H. Qi, W. Jin, L. Zhu, J. Sun, Multi-view distillation based on multi-modal fusion for few-shot action recognition (CLIP-mdmf), Knowl.-Based Syst. 304 (2024) 112539.

[32] F. Guo, Y. Wang, H. Qi, L. Zhu, J. Sun, Consistency prototype module and motion compensation for few-shot action recognition (CLIP-CPM2c), Neurocomputing 611 (2025) 128649.

[33] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, S. Gelly, Parameter-efficient transfer learning for NLP, in: International Conference on Machine Learning, PMLR, 2019, pp. 2790–2799.

[34] E.B. Zaken, S. Ravfogel, Y. Goldberg, Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, 2021, arXiv preprint arXiv:2106.10199.

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

[36] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, S.-N. Lim, Visual prompt tuning, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII, Springer, 2022, pp. 709–727.

[37] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L.V. Gool, Temporal segment networks: Towards good practices for deep action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 20–36.

[38] Y. Huang, L. Yang, Y. Sato, Compound prototype matching for few-shot action recognition, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, Springer, 2022, pp. 351–368.

[39] J. Xing, M. Wang, B. Mu, Y. Liu, Revisiting the spatial and temporal modeling for few-shot action recognition, 2023, arXiv preprint arXiv:2301.07944.

[40] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6299–6308.

[41] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre, HMDB: a large video database for human motion recognition, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2556–2563.

[42] K. Soomro, A.R. Zamir, M. Shah, UCF101: A dataset of 101 human actions classes from videos in the wild, 2012, arXiv preprint arXiv:1212.0402.

[43] R. Goyal, S. Ebrahimi Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, et al., The" something something" video database for learning and evaluating visual common sense, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 5842–5850.

[44] Z. Zhu, L. Wang, S. Guo, G. Wu, A closer look at few-shot video classification: a new baseline and benchmark, 2021, arXiv preprint arXiv:2110.12358.

[45] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[46] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.

[47] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[48] S. Zheng, S. Chen, Q. Jin, Few-shot action recognition with hierarchical matching and contrastive learning, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV, Springer, 2022, pp. 297–313.

[49] K.D. Nguyen, Q.-H. Tran, K. Nguyen, B.-S. Hua, R. Nguyen, Inductive and transductive few-shot video classification via appearance and temporal alignments, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XX, Springer, 2022, pp. 471–487.

[50] J. Xing, M. Wang, Y. Ruan, B. Chen, Y. Guo, B. Mu, G. Dai, J. Wang, Y. Liu, Boosting few-shot action recognition with graph-guided hybrid matching, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 1740–1750.

Jiazheng Xing received the B.S. degree from Chongqing University, Chongqing, China in 2021. He is currently working toward the Ph.D. degree with the Laboratory of Advanced Perception on Robotics and Intelligent Learning, College of Control Science and Engineering. His research interests include few-shot action recognition, action recognition, generative AI, computer vision, and deep learning.

Jian Zhao received the bachelor's degree from Beihang University, Beijing, China, in 2012, the master's degree from the National University of Defense Technology, Changsha, China, in 2014, and the Ph.D. degree from the National University of Singapore, Singapore, in 2019. He is currently an Assistant Professor at the Institute of North Electronic Equipment, Beijing. His main research interests include deep learning, pattern recognition, computer vision, and multimedia analysis.

Chao Xu received the B.S. degree in electrical engineering and its automation from Nanchang University, Nanchang, China in 2018. He received the Ph.D. degree in control science and engineering with the School of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2023. His major research interests include computer vision and deep learning.

Mengmeng Wang received the Ph.D. degree in control science and engineering from Zhejiang University in 2024. She is currently working at the College of Computer Science and Technology at Zhejiang University of Technology. Her research interests include image/video understanding, text-to-video generation, computer vision, robotics, and deep learning.

Guang Dai received his B.Eng. degree in Mechanical Engineering from the Dalian University of Technology and M.Phil. degree in Computer Science from the Zhejiang University and the Hong Kong University of Science and Technology. He is currently a senior research scientist at State Grid Corporation of China. His main research interests include Bayesian statistics, deep learning, reinforcement learning, and related applications.

Yong Liu received the B.S. degree in computer science and engineering and the Ph.D. degree in computer science from Zhejiang University, Zhejiang, China, in 2001 and 2007, respectively. He is currently a professor at the Institute of Cyber-Systems and Control at Zhejiang University. His main research interests include robot perception and vision, deep learning, big data analysis, and multi-sensor fusion.

Jingdong Wang is Chief Scientist for computer vision with Baidu. Before joining Baidu, he was a Senior Principal Researcher at Microsoft Research Asia from September 2007 to August 2021. His areas of interest include vision foundation models, self-supervised pretraining, OCR, human pose estimation, semantic segmentation, image classification, object detection, and large-scale indexing.

Xuelong Li is currently a Full Professor with the School of Artificial Intelligence, OPtics and ElectroNics (iOPEN), Northwestern Polytechnical University, Xian, China. He is also with the Key Laboratory of Intelligent Interaction and Applications, Ministry of Industry and Information Technology, Northwestern Polytechnical University.