Spatial-temporal Attributes Enhanced Prompt Weighting and Fusion for Video Recognition

Weilin Qiu[1] **Kaiyan Cao**[1] **Andy Jin-Hua Ma**[1,2,3*]

[1]Sun Yat-sen University, China
[2]Guangdong Provincial Key Laboratory of Information Security Technology, China
[3]Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China
{qiuwlin5, caoky3}@mail2.sysu.edu.cn, majh8@mail.sysu.edu.cn

Abstract

Adapting large-scale vision-language models (VLMs) such as CLIP to video understanding has drawn increasing attention. Prompt tuning offers an efficient and promising way to adapt VLMs to this task. While existing methods propose to enrich textual information over simple textual templates via large language models (LLMs), the generated descriptions are often coarse and may lead to semantic misalignment with the specific video. To address this issue, we propose **S**patial-**T**emporal **A**ttributes enhanced prompt **W**eighting and **F**usion (ST-AWF), a novel parameter-efficient fine-tuning framework for various video recognition tasks. In our method, LLM is leveraged to generate spatial and temporal attributes for each action category, enriching text prompts with informative descriptions. To mitigate attribute irrelevance in LLM outputs, we design a vision-guided prompt weighting mechanism that dynamically evaluates the relevance of each attribute embedding based on its similarity to video features, thereby emphasizing discriminative cues. Furthermore, a dual-attention cross-modal fusion module is proposed to align weighted textual prompts of spatial–temporal attributes with video features for fine-grained cross-modal feature enhancement. Extensive experiments demonstrate that ST-AWF achieves state-of-the-art performance on multiple video recognition benchmarks, consistently improving both discriminability and generalization ability to unseen classes, while maintaining high efficiency with a small number of trainable parameters. Code is available at https://github.com/QiuWilliam/ST-AWF.git.

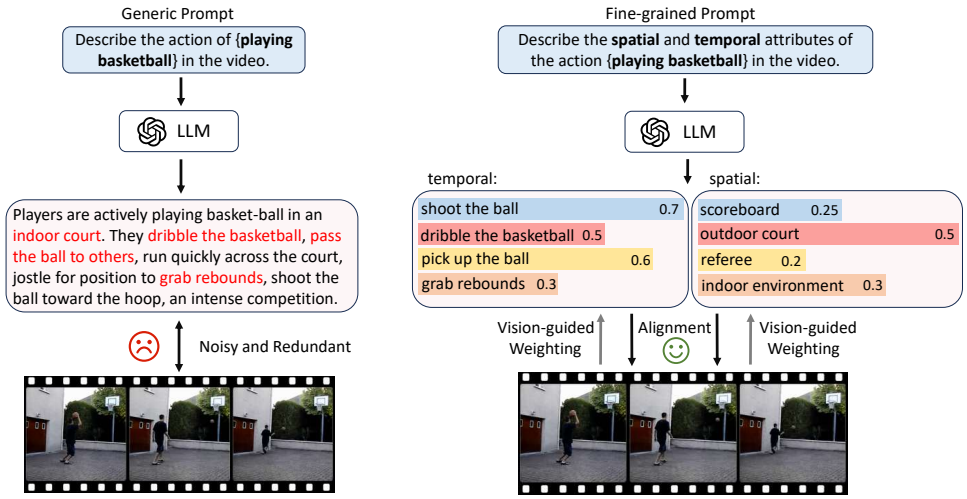
1 Introduction

Pretrained on massive image-text datasets, vision-language models (VLMs), such as CLIP [29], ALIGN [12], and BLIP-2 [20], demonstrate strong generalization ability. They have been successfully deployed in a range of image-based downstream tasks, including image captioning [26], few-shot learning [11, 32, 37], and image retrieval [7, 25]. Building on the success in image-based applications, recent efforts have increasingly focused on transferring the powerful cross-modal representation capabilities to video understanding [10, 30, 36, 39, 40].

To bridge the modality gap between images and videos [21], many existing studies adopt full-parameter fine-tuning for adapting CLIP to video tasks [27, 28, 41]. Temporal modeling through cross-frame interactions [39] or temporal attention [17] has been proposed to capture temporal variations in videos. To preserve the generalization ability aligned with the original CLIP model, researchers have designed regularization strategies such as weight ensembling [6, 17, 33, 39] and

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Existing Method (b) Ours

Figure 1: Motivation of this work. (a) Unprocessed outputs introduce noisy and redundant descriptions, resulting in semantic misalignment. (b) We propose to generate spatial and temporal attributes for each action class. Then, these attribute prompts are adaptively weighted according to visual features, suppressing irrelevant attributes w.r.t. each specific video for more accurate alignment.

knowledge distillation [10]. Nevertheless, full-parameter fine-tuning is computationally expensive. Moreover, it may overwrite the general knowledge acquired during CLIP pre-training, thereby increasing the risk of overfitting to task-specific data.

In contrast to full-parameter fine-tuning, an alternative approach is parameter-efficient fine-tuning (PEFT). Representative methods such as prompt tuning [30, 33, 34] and adapter tuning [1, 35] achieve fast task adaptation while preserving the original representation capacity of CLIP through optimization of only a small set of newly introduced parameters. Among them, prompt tuning utilizes learnable textual prompts to directly leverage pre-trained vision–language knowledge and achieves promising results. Within this paradigm, existing methods [30, 44] inherit the conventional design of vision–language models, relying on simple text templates (e.g., "a video of a {class}") to construct prompts. While videos encapsulate far more complex information than static images, such sparse textual inputs fail to bridge the gap between dense visual content and abstract action categories.

To enrich textual information, large language models (LLMs) are leveraged to generate category descriptions in [6, 10, 13]. Despite the improving performance, the inherent redundancy and noise in these descriptions are often overlooked. Specifically, LLM-generated texts are prompted merely by generic cues and category labels. They tend to be overly general [13], containing irrelevant or absent details for a given video instance, as shown in Figure 1. Notably, the degree of alignment between these descriptions and individual video instances varies significantly even within the same category. Directly using these unprocessed texts may introduce semantic misalignment with the visual content, leading to underutilization and degrading the learning discriminability.

To overcome this limitation, we propose **S**patial-**T**emporal **A**ttributes enhanced prompt **W**eighting and **F**usion (ST-AWF), a novel PEFT framework that leverages prompt learning to bridge the visual-textual semantic gap. In our method, LLM is employed to generate spatial and temporal attributes for each action category. These attributes are embedded into text prompts, replacing static class labels with expressive and context-aware descriptions. To reduce redundancy and noisy matching between raw LLM outputs and each video, we design an adaptive attribute weighting mechanism. By computing the similarity between attribute embeddings and video features, our method suppresses irrelevant or noisy attributes, preserving only the most discriminative key textual prompts. Finally, for fine-grained alignment, we propose a dual-attention cross-modal fusion module that combines spatial and temporal textual attributes with video features respectively. This ensures visual semantics are reinforced by complementary textual prompts, enhancing both discriminative power and generalization ability.

Main contributions of this work are summarized as follows:

- We propose ST-AWF, a novel prompt learning framework that leverages fine-grained spatial and temporal attributes to align with video features via a dual-attention cross-modal fusion mechanism, achieving better generalization ability.
- We design a vision-guided attribute weighting mechanism that adaptively assigns higher weights to prompt embeddings of relevant textual attributes, effectively highlighting key spatial-temporal cues while mitigating noise and redundancy in textual descriptions.
- Our method achieves state-of-the-art performance across a range of video recognition tasks including base-to-novel generalization, few-shot and zero-shot video recognition.

2 Related Work

CLIP-Based Action Recognition. Inspired by the CLIP strong generalization, numerous video action recognition methods have emerged, broadly classified into full-parameter fine-tuning [6, 17, 27] and parameter-efficient fine-tuning [15, 38, 44]. Among them, full-parameter fine-tuning dominates. ViFi-CLIP [30] shows that competitive baseline performance is attainable through direct training. TC-CLIP [17] proposes temporal context mechanism to aggregate global frame features for temporal modeling. However, full-parameter training tends to overfit, limiting performance in open-vocabulary tasks. To address this, Open-VCLIP [39] dynamically interpolates frozen CLIP weights with the current video learner during training, constraining deviation from the original feature space. FROSTER [10] applies knowledge distillation to align with frozen CLIP via a customized residual module. [6, 17, 24] use weight ensembling to retain pre-trained knowledge. PEFT-based methods are generally categorized into prompt-based and adapter-based approaches. MSTA [4] inserts adapters into higher CLIP layers for efficient fine-tuning. Given PEFT advantages in computational cost and open-vocabulary performance, designing PEFT frameworks for video is essential.

Prompt Learning For CLIP. Prompt learning first achieved success in natural language processing [19, 22] and was later adapted for downstream CLIP tasks. CoOp [43] pioneers replacing hand-crafted templates with optimized continuous learnable vectors, yielding strong few-shot image classification results. VPT [14] explores visual prompting by inserting tokens into the vision encoder, while MaPLe [16] proposes synergistic prompt learning across both vision and language branches. In contrast to its rapid progress in image tasks, prompt learning for video action recognition remains limited [30, 33, 38, 44]. Vita-CLIP [38] uses multimodal prompts to adapt CLIP to video tasks, effectively integrating video and text representations; ViLT-CLIP [33] transfers the frozen CLIP model to downstream video tasks via multimodal prompt learning and scene-guided optimization. We utilize explicit fine-grained spatiotemporal attribute text to enrich prompt semantic priors, enabling multimodal joint optimization through a deep cross-modal interaction architecture.

3 Method

The framework of ST-AWF, as illustrated in Figure 2, mainly consists of three core components: Spatial-temporal Attribute for enhanced prompting, Vision-guided Prompt Weighting for noise mitigation, and Dual-attention Cross-modal Fusion for deep alignment. First, we provide a brief overview about the application of CLIP to video action recognition in Section 3.1. Subsequently, we present a comprehensive description of the proposed method in the following subsections.

3.1 Preliminaries

Video Action Recognition with CLIP Leveraging its strong representations from image-text pretraining, CLIP can be directly applied to video action recognition by encoding frame-wise features and aggregating them via temporal averaging. For a T-frame video $V \in \mathbb{R}^{T\times H\times W\times 3}$ and its associated textual description C (formatted as, e.g., "a video of {action}"), the vision encoder f_v and the text encoder f_t of CLIP are employed to derive corresponding video and textual feature representations. The training optimization objective is contrastive learning paradigm, maximizing the similarity between matched video-text pairs while minimizing that between mismatched pairs. This similarity is typically computed using the cosine similarity metric:

$$sim(v, c) = \frac{\langle v, c\rangle}{|v||c|}, v = f_v(V), c = f_t(C). \tag{1}$$

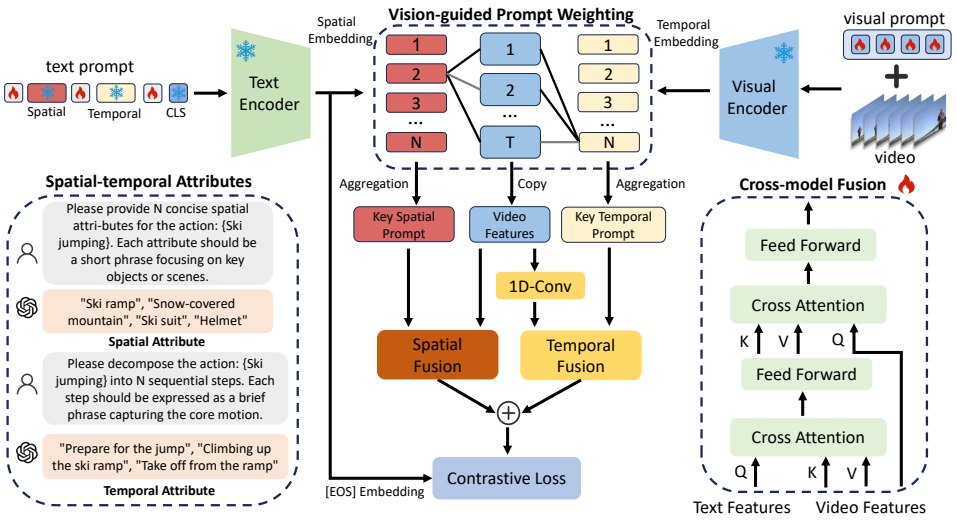


Figure 2: Overall framework of the proposed ST-AWF. Our method first decouples text prompt into Spatial and Temporal Attributes using LLM. These attributes are dynamically selected by a Vision-guided Prompt Weighting mechanism to generate key prompts. Refined text knowledge is then aligned with video via detailed Dual-attention Cross-modal Fusion.

Prompt Learning for CLIP Compared to full-parameter fine-tuning of CLIP, prompt learning is a parameter-efficient paradigm. This method replaces human-crafted text with optimizable embeddings and visual prompts, maintaining a fixed backbone architecture while exclusively fine-tuning the added parameters. Specifically, continuous learnable soft prompt vectors are introduced into the text and vision branches, denoted as $[T_i]_{i=1}^{M}$ and $[V_i]_{i=1}^{N}$, respectively. In the task of video action recognition, the text branch creates inputs by concatenating soft prompts with the discrete label words of the target category labels. Correspondingly, the vision branch injects these prompts at the feature level by prepending them to extracted video patch representations, establishing consistent prompt utilization across modalities. The specific formulation is defined as follows:

$$P_t = [T_1][T_2]\dots[T_M][\text{CLS}], \quad P_v = [V_1][V_2]\dots[V_N][E_v] \tag{2}$$

where CLS denotes the features of the label word for the category, and E_v denotes the initial visual sequence features extracted through the patch embedding of ViT.

3.2 Spatial-temporal Attributes Enhanced Prompting

To enrich the textual information, we incorporate category descriptions into the text prompts. Similar to the ATPrompt [23], we embed corresponding attribute words into the input text prompts. In this work, we leverage GPT-5 to generate attribute descriptors for each action category. However, relying solely on LLM to generate generalized action attribute descriptions fails to fully capture the unique characteristics of video actions, thus limiting the potential of attribute words to complement the text information. To address this, we take into account the specific characteristics of video data and propose a more targeted approach by constructing action category attribute descriptions from both spatial and temporal dimensions:

- Spatial attributes delineate static visual elements present throughout the entire video. These elements primarily consist of objects or scenes, such as interacting entities, tools, or background environments that are highly relevant to the action.
- Temporal attributes execute a sequential decomposition of the video action, separating the action into a series of fine-grained sub-actions.

We employ a structured prompt strategy to guide the LLM in generating attribute words. To avoid information truncation due to the length limitations of the CLIP text encoder, we strictly control the output length in the prompt engineering process. This ensures that all generated attribute words are in

phrase form, striking a balance between information completeness and model compatibility. For each action category {cls}, we prompt LLM to obtain N spatial attribute words and N temporal attribute words, respectively.

Following the generation of the corresponding spatial and temporal attribute word sets for each action category via the aforementioned two queries, we construct the complete text prompt. Each category corresponds to a unique text prompt, formulated as follows:

$$P_T = [T_a][A_1]\dots[A_N][T_b][B_1]\dots[B_N][T_c][\text{CLS}]. \tag{3}$$

Here, $\{[A_i]\}_{i=1}^{N}$ and $\{[B_i]\}_{i=1}^{N}$ represent the spatial and temporal text attribute words corresponding to the action category, respectively. $[T_a]$, $[T_b]$, and $[T_c]$ denote the learnable prompts for the spatial attributes, temporal attributes, and the class category with each consisting of the same number of soft tokens. This prompt design decouples spatial attributes, temporal attributes, and category and prevents different types of information from interfering with each other. In parallel to the text branch, we also introduce visual prompts by adding a set of learnable tokens to the input video frames, maintaining consistency with the prompt-based optimization paradigm.

By embedding spatiotemporal attribute words into the text prompts, we achieve the transfer of fine-grained spatiotemporal knowledge into the text representation space. This design enables the learnable soft tokens to continuously receive semantic guidance from the attribute tokens, guiding the model to understand the inherent patterns of actions and thereby improving its generalization ability.

3.3 Vision-guided Prompt Weighting

To enhance the alignment between spatiotemporal attribute prompts and the visual semantics, we introduce a visual-guided prompt weighting mechanism. The core idea of this module is to dynamically adjust the attribute weights based on the similarity between the visual feature and spatiotemporal attributes. This adaptive weighting approach effectively highlights key attributes that are highly relevant to the current video while suppressing irrelevant or redundant text features, thus achieving dual optimization in noise filtering and semantic focusing.

Specifically, after feeding the constructed text prompt P_T into the text encoder, we obtain a sequence of token representations $\widetilde{\mathbf{W}} = [\widetilde{\mathbf{w}}_1, \dots, \widetilde{\mathbf{w}}_M] \in \mathbb{R}^{M\times d}$, where M denotes the total number of tokens in the sequence and d indicates the feature dimension. Subsequently, we extract the features of each attribute word from the sequence. Since attribute phrase may be tokenized into several sub-word tokens, we identify the corresponding token positions and aggregate their representations via average pooling to form a single representation for each attribute. This yields N spatial attribute embeddings $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_N] \in \mathbb{R}^{N\times d}$ and N temporal attribute embeddings $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_N] \in \mathbb{R}^{N\times d}$, which is the same as the number of attribute words generated by the LLM. Concurrently, the video is fed into the vision encoder to extract the frame-level visual feature representations $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_T] \in \mathbb{R}^{T\times d}$, where T represents the number of video frames.

We first compute the cosine similarity matrix between the frame features of the video and the attribute text features of all categories. Using the spatial attributes of a single category as an example, the similarity between the t-th frame and the i-th spatial attribute $\mathbf{a}_i$ is calculated as follows:

$$s_{t,i}^{spa} = \frac{\mathbf{v}_t^\top \mathbf{a}_i}{|\mathbf{v}_t||\mathbf{a}_i|} \tag{4}$$

Through the above computations, we obtain the similarity matrix between each frame and each spatial attribute. To calculate the matching score between the entire video and the attribute, we design a Top-K frame filtering strategy. Using the frame-level similarity, we compute the matching scores for spatial attribute $\mathbf{a}_i$ by selecting the K frames that exhibit the highest similarities and averaging them:

$$\Omega_i^{spa} = \text{TopK}(\{s_{t,i}^{spa}\}_{t=1}^{T}, K) \tag{5}$$

$$w_i^{spa} = \frac{1}{K}\sum_{t\in\Omega_i^{spa}} s_{t,i}^{spa} \tag{6}$$

where Ω_i^{spa} represents the index set of the K frames with the highest similarity scores for the i-th spatial attribute. In this way, only the frames correlated with the attributes participate in the score

calculation, which effectively suppresses the interference of uninformative frames. Finally, the matching scores of all attributes are normalized to compute the weight of each attribute. These weights are subsequently applied to the attribute texts to formulate the key spatial prompt $\mathbf{P}_{\text{spa}}$:

$$\tilde{w}_i^{\text{spa}} = \frac{\exp(w_i^{\text{spa}})}{\sum_{j=1}^{N} \exp(w_j^{\text{spa}})} \tag{7}$$

$$\mathbf{P}_{\text{spa}} = \sum_{i=1}^{N} \tilde{w}_i^{\text{spa}} \cdot \mathbf{a}_i \tag{8}$$

Similarly, the key temporal prompt $\mathbf{P}_{\text{tem}} = \sum_{i=1}^{M} \tilde{w}_i^{\text{tem}} \cdot \mathbf{b}_i$ is computed for the temporal attributes. These weighted key attribute representations are subsequently fed into the cross-modal fusion module.

3.4 Dual-attention Cross-modal Fusion

To further obtain features with consistent visual-textual semantics, we perform cross-modal fusion between the text and visual modalities. We design a dual-branch spatiotemporal cross-modal fusion module that combines spatiotemporal key attribute text features with video features separately. Moreover, we implement bidirectional cross-attention in both branches to facilitate cross-modal knowledge interaction. The architecture consists of two stages: the first cross-attention layer aligns the text and video modalities, while the second layer enhances the video representation, thereby further improving the performance of model.

Taking spatial fusion as an example, in the first cross-attention layer, we utilize the key spatial prompt $\mathbf{P}_{\text{spa}}$ as the query $\mathbf{Q}$, alongside the video features $\mathbf{V}$ as both the key $\mathbf{K}$ and the value $\mathbf{V}$. We uses text information to refine the video features:

$$\mathbf{Q}_1 = \mathbf{P}_{spa}\mathbf{W}_{Q1}, \quad \mathbf{K}_1 = \mathbf{V}\mathbf{W}_{K1}, \quad \mathbf{V}_1 = \mathbf{V}\mathbf{W}_{V1} \tag{9}$$

$$\mathbf{Z}_1 = \text{Softmax}\left(\frac{\mathbf{Q}_1\mathbf{K}_1^\top}{\sqrt{d_k}}\right)\mathbf{V}_1 \tag{10}$$

$$\tilde{\mathbf{P}}_{spa} = \mathbf{P}_{spa} + \text{FFN}(\mathbf{Z}_1) \tag{11}$$

where $\mathbf{W}_{Q1}$, $\mathbf{W}_{K1}$, and $\mathbf{W}_{V1}$ are learnable linear projection, and d_k is the scaling factor. Through this layer, we obtain the aligned attribute knowledge $\tilde{\mathbf{P}}_{spa}$, which extracts relevant visual context.

In the second cross-attention layer, we inject this aligned attribute knowledge back into the visual space to enhance the visual features with semantically relevant attribute information. We use the original video features $\mathbf{V}$ as the Query, and the output of the first layer, $\tilde{\mathbf{P}}_{spa}$, as the Key and Value:

$$\mathbf{Q}_2 = \mathbf{V}\mathbf{W}_{Q2}, \quad \mathbf{K}_2 = \tilde{\mathbf{P}}_{spa}\mathbf{W}_{K2}, \quad \mathbf{V}_2 = \tilde{\mathbf{P}}_{spa}\mathbf{W}_{V2} \tag{12}$$

$$\mathbf{Z}_2 = \text{Softmax}\left(\frac{\mathbf{Q}_2\mathbf{K}_2^\top}{\sqrt{d_k}}\right)\mathbf{V}_2 \tag{13}$$

$$\tilde{\mathbf{V}}_{spa} = \mathbf{V} + \text{FFN}(\mathbf{Z}_2) \tag{14}$$

After the computation through the two cross-attention layers, we obtain the video features $\tilde{\mathbf{V}}_{spa}$ fused with the spatial text attributes.

For the temporal fusion module, considering the differences between spatial and temporal text attributes, we introduce a temporal modeling module to perform temporal enhancement on the video. Following the ALT [8], we use 1D-Conv for the corresponding operation, which efficiently aggregate local temporal dependencies with minimal computational overhead. Before the video features are fed into the fusion module, temporal modeling is additionally performed on them:

$$\mathbf{V} = \mathbf{V} + \text{Conv1D}(\mathbf{V}) \tag{15}$$

Subsequently, following procedure similar to the spatial fusion module, the video features $\mathbf{V}$ and the key temporal prompt $\mathbf{P}_{\text{tem}}$ are jointly input into the temporal fusion module. This process ultimately yields the video features $\tilde{\mathbf{V}}_{\text{tem}}$ integrated with the temporal attributes. Finally, we apply a residual connection to the spatially enhanced and temporally enhanced video features derived from the two branches. This operation generates the global video representation $\mathbf{V}_{\text{final}}$, which is utilized for the final classification computation:

$$\mathbf{V}_{final} = \tilde{\mathbf{V}}_{spa} + \tilde{\mathbf{V}}_{tem} \tag{16}$$

3.5 Training Loss

After the cross-modal fusion, we apply average pooling to the video feature sequence $\mathbf{V}_{final}$ along the temporal dimension: $\mathbf{z} = \text{AvgPool}(\mathbf{V}_{final})$, to obtain the global video representation. Simultaneously, we extract the [EOS] token feature from the text prompt sequence to serve as the text representation $\mathbf{t}_k$ for the corresponding category. During the training phase, the model is optimized by minimizing the cross-entropy loss:

$$\mathcal{L} = -\log \frac{\exp(\cos(\mathbf{z}, \mathbf{t}_y)/\tau)}{\sum_{k=1}^{K} \exp(\cos(\mathbf{z}, \mathbf{t}_k)/\tau)} \tag{17}$$

where y represents the ground-truth action label corresponding to the video, $\cos(\cdot, \cdot)$ denotes the cosine similarity, and τ is the temperature hyperparameter. Throughout the entire training process, the parameters of the pre-trained CLIP backbone are kept frozen, with optimization and adjustment applied only to the introduced learnable soft prompt vectors and the multi-modal fusion module.

Table 1: Base to novel generalization. HM denotes harmonic mean, balancing the performance between base and novel classes. The **best** and the second-best results are highlighted. † denotes the results reproduced with our implementation using standard CLIP.

Method	Trainable Param (M)	HMDB-51			UCF-101			SSv2			K-400		
		Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
Vanilla CLIP[29]	-	53.3	46.8	49.8	78.5	63.6	70.3	4.9	5.3	5.1	62.3	53.4	57.5
ActionCLIP[36]	168.5	69.1	37.3	48.4	90.1	58.1	70.6	13.3	10.1	11.5	61.0	46.2	52.6
X-CLIP[27]	131.5	69.4	45.5	55.0	89.9	58.9	71.2	8.5	6.6	7.4	74.1	56.4	64.0
PromptCLIP[15]	5.0	46.2	16.0	23.8	90.5	40.4	55.9	8.3	5.3	6.5	69.7	37.6	48.8
ViFi-CLIP[30]	124.7	73.8	53.3	61.9	92.9	67.7	78.3	16.2	12.1	13.9	76.4	61.1	67.9
ViLT-CLIP[33]	4.5	76.7	57.5	65.7	95.2	70.5	81.0	17.3	12.8	14.7	77.4	63.0	69.5
OST[6]	149.6	75.1	34.4	47.2	**96.1**	68.9	80.3	15.3	11.5	13.1	75.4	59.6	66.6
MSTA † [4]	8.7	75.7	51.0	60.9	95.4	64.9	77.3	17.4	5.8	8.7	77.4	59.1	67.0
VTD-CLIP [44]	3.5	73.1	47.1	57.3	95.5	73.7	83.2	17.8	13.9	15.4	**78.5**	63.5	70.1
ST-AWF(Ours)	5.3	**76.9**	**58.5**	**66.5**	95.7	**75.5**	**84.5**	**18.5**	**14.1**	**16.0**	77.9	**64.0**	**70.3**

Table 2: Few-shot setting. We conduct experiments with 2, 4, 8, and 16 video samples per category for training. The **best** and the second-best results are highlighted. † denotes our implementation based on standard CLIP.

Method	Trainable Param (M)	HMDB-51				UCF-101				SSv2			
		$K=2$	$K=4$	$K=8$	$K=16$	$K=2$	$K=4$	$K=8$	$K=16$	$K=2$	$K=4$	$K=8$	$K=16$
Vanilla CLIP [29]	-	41.9	41.9	41.9	41.9	63.6	63.6	63.6	63.6	2.7	2.7	2.7	2.7
ActionCLIP [36]	168.5	47.5	57.9	57.3	59.1	70.6	71.5	73.0	91.4	4.1	5.8	8.4	11.1
X-CLIP [27]	131.5	53.0	57.3	62.8	64.0	48.5	75.6	83.7	91.4	3.9	4.5	6.8	10.0
PromptCLIP [15]	5.0	39.7	50.7	56.0	62.4	71.4	79.9	85.7	89.9	4.4	5.1	6.1	9.7
ViFi-CLIP [30]	124.7	57.2	62.7	64.5	66.8	80.7	85.1	90.0	92.7	6.2	7.4	8.5	**12.4**
OST[6]	149.6	59.1	62.9	64.9	68.2	82.5	87.5	91.7	**93.9**	7.0	7.7	8.9	12.2
MSTA † [4]	8.7	56.8	59.8	65.0	66.8	82.1	87.5	90.8	93.6	6.6	7.5	8.8	11.8
VTD-CLIP [44]	3.5	54.0	59.3	64.3	67.7	80.7	86.5	91.3	92.6	5.1	7.8	8.4	12.0
ST-AWF(Ours)	5.3	**60.1**	**63.3**	**65.7**	**68.3**	**83.6**	**88.7**	**92.2**	93.5	**7.1**	**8.0**	**9.2**	**12.4**

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments across five benchmark datasets for video action recognition: Kinetics-400 [3], Kinetics-600[2], UCF-101 [31], HMDB-51 [18], and Something-Something V2 [9]. The evaluation protocol includes base-to-novel generalization, few-shot recognition, and zero-shot recognition. Additional details are provided in the Appendix.

Implementation Details. All experiments utilize the frozen ViT-B/16 network of CLIP. To validate the open-vocabulary capability of the proposed method, we evaluate base-to-novel generalization on HMDB-51, Kinetics-400, UCF-101, and Something-Something V2; few-shot on HMDB-51,

UCF-101, and SSv2; and zero-shot on HMDB-51, Kinetics-600, UCF-101. For each category, we generate $N = 3$ spatial and temporal attribute words, respectively, and set the top K frame to $K = 16$ during the generation of the key prompt. All models are trained across eight NVIDIA RTX 3090 GPUs. Additional details are provided in the Appendix.

4.2 Main Results

To comprehensively and fairly evaluate the effectiveness of the proposed framework, our main experiments focus on comparing ST-AWF with state-of-the-art parameter-efficient fine-tuning methods. And we also include comparisons with several classic video full-parameter fine-tuning methods.

Base-to-novel generalization. In Table 1, we compare our method with other CLIP-based methods under the base-to-novel setting. Our method achieves the best performance in terms of HM across all four datasets, indicating a superior balance between base and novel classes. Meanwhile, compared to several fully fine-tuned methods, our approach achieves better results with fewer trainable parameters. Notably, our approach shows clear advantages on novel classes, achieving improvements of 1.0%, 1.8%, 0.2% and 0.5% on the four datasets respectively, which highlights its strong generalization capability to unseen categories.

Few-shot video recognition. We assess the learning capability and generalization potential of our model when in the difficult all-way few-shot conditions. The comparison results on three datasets are presented in Table 2. Our approach achieved 11 out of 12 best performances across different sample settings for all datasets, with improvements of up to 1.2% over competing methods. In particular, under extremely few-shot settings (e.g., $K = 2, 4$), our method shows superior performance, demonstrating strong data efficiency.

Zero-shot video recognition. In Table 3, we present our zero-shot video recognition results. In this setting, all models are trained on the Kinetics-400 dataset and directly evaluated on downstream datasets to assess its generalization capability to unseen datasets. Our method consistently outperforms existing approaches across all datasets, achieving state-of-the-art performance on HMDB-51, UCF-101, and Kinetics-600. Compared to the second-best method, our model achieves consistent improvements of approximately +0.7% on HMDB-51, +1.3% on UCF-101, and +1.6% on Kinetics-600, highlighting its effectiveness across different datasets. These results strongly indicate that our approach effectively mitigates overfitting to the source domain, thereby preserving robust generalization to unseen data distributions.

Table 3: Zero-shot video recognition. The results are top-1 accuracies with mean and standard deviation on the evaluation across three validation splits each dataset. The **best** and the second-best results are highlighted. † denotes our implementation based on standard CLIP.

Method	HMDB-51	UCF-101	K600
Vanilla CLIP[29]	40.8 ± 0.3	63.2 ± 0.2	59.8 ± 0.3
ActionCLIP[36]	40.8 ± 5.4	58.3 ± 3.4	66.7 ± 1.1
X-CLIP[27]	44.6 ± 5.2	72.0 ± 2.3	65.2 ± 0.4
Vita-CLIP[38]	48.6 ± 0.6	75.0 ± 0.6	67.4 ± 0.5
ViLT-CLIP[33]	45.3 ± 0.9	73.6 ± 1.1	-
MSTA † [4]	45.8 ± 0.6	72.1 ± 1.0	67.8 ± 0.4
ST-AWF(Ours)	**49.3 ± 1.0**	**76.3 ± 0.5**	**69.0 ± 0.3**

4.3 Ablation Studies

We conduct ablation studies under the base-to-novel setting, as shown in Table 4. To facilitate fast ablation, the reported results are averaged over two primary datasets: HMDB-51 and UCF-101.

Component Analysis. To evaluate the contribution of each component, we conduct a progressive ablation study in Table 4a. Starting with prompt tuning the original CLIP [30] without extra modules, we first add LLM-generated spatial-temporal attributes to the text prompts, substantially improving novel-class accuracy. Fusing video features directly with the pooling attributes then yields consistent gains. Finally, introducing the Vision-guided Prompt Weighting mechanism achieves optimal overall performance. Compared to the baseline, our complete framework significantly improves Base, Novel, and HM by 3.6%, 10.2%, and 8.1%, respectively.

Different Attribute Design Analysis. We explore different attribute texts: manually selected class-irrelevant attributes and those directly generated by LLMs. As shown in Table 4b, irrelevant

Table 4: Ablation studies over two datasets under the base-to-novel generalization setting. Results are averaged across UCF-101 and HMDB-51.

Attribute	Fusion	Weighting	Base	Novel	HM
×	×	×	82.7	56.8	67.3
✓	×	×	84.3	64.6	73.2
✓	✓	×	85.1	65.2	73.9
✓	✓	✓	86.3	67.0	75.4

(a) Ablation on different components.

Attribute Setting	Base	Novel	HM
Irrelevant	83.2	57.9	68.3
LLMs-Direct	85.3	61.1	71.2
Spatial Only	85.6	63.9	73.2
Temporal Only	86.4	65.0	74.2
Spatial-Temporal	86.3	67.0	75.4

(b) Ablation on attribute design.

K	Base	Novel	HM
1	85.5	65.9	74.4
8	86.0	65.8	74.6
16	86.3	67.0	75.4
24	86.3	66.9	75.3
32	86.2	66.0	74.8

(c) Ablation on K.

N	Base	Novel	HM
2	85.8	66.1	74.7
3	86.3	67.0	75.4
4	86.2	66.7	75.2

(d) Ablation on attribute number.

Fusion	Base	Novel	HM
MLP	86.6	65.3	74.4
Single Attention	86.1	65.9	74.7
Dual Attention	86.3	67.0	75.4

(e) Ablation on different cross-modal fusion method.

Temporal	Base	Novel	HM
None	85.4	64.9	73.8
Self-Attention	85.8	65.7	74.4
1D-Conv	86.3	67.0	75.4

(f) Ablation on temporal module.

attributes yield the lowest performance, while LLM-generated ones improve but still underperform well-designed attributes. Both spatial-only and temporal-only attributes perform better, with temporal ones slightly superior, reflecting motion cues' importance in videos. Combining spatial and temporal attributes achieves the best results, highlighting their complementarity.

Top-K Analysis. We investigate the impact of filter frame count K on matching performance in Table 4c. K=1 selects the most relevant frame, while K=32 average-pools all frames. Balancing accuracy and efficiency, we adopt K=16 as optimal configuration for our framework.

Attribute Number Analysis. We investigate the influence of attributes numbers in Table 4d. Results show that the model achieves its optimal performance at $N = 3$. Utilizing fewer attributes limits the capacity of textual semantics, while generating excessive attributes slightly degrades generalization.

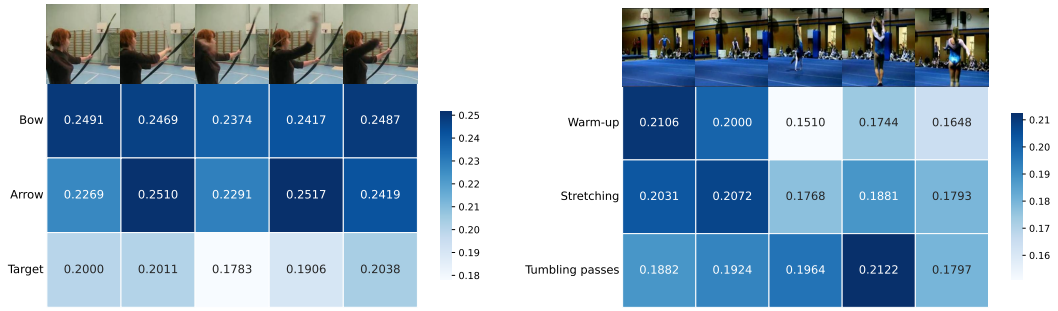
Multi-modal Fusion Mechanisms Analysis. We evaluate the impact of different multi-modal fusion modules, including MLP and one-step cross-attention. As shown in Table 4e, compared to the MLP, attention mechanisms effectively improve generalization. Notably, the Dual Attention achieves the best overall performance, boosting the HM to 75.4%. This indicates that a more comprehensive interaction between modalities leads to better alignment and improved recognition performance.

Temporal Module Analysis. To investigate the impact of different temporal module in the temporal fusion, we compare three variants: (1) no temporal module, the same as spatial fusion, (2) self attention, and (3) 1D convolution. As shown in Table 4f, temporal modeling consistently improves performance, with 1D convolution achieving the best results.

5 Visualization

5.1 Visualizations of Attribute Weighting

To better understand ST-AWF, we visualize the similarities assigned to spatial and temporal attributes in Figure 3. In "Archery", the core interactive objects, "Bow" and "Arrow", consistently receive higher similarities than "Target", while the similarity of "Arrow" fluctuates across frames and peaks at key interaction moments (Figure 3a). In "Floor Gymnastics", the model initially emphasizes "Warm-up" and "Stretching", then adaptively shifts its focus to "Tumbling passes" as the action progresses, with a notable peak at the 4th frame (Figure 3b). These observations demonstrate that ST-AWF dynamically adjusts attribute importance according to evolving video content, effectively capturing fine-grained discriminative cues while reducing the influence of irrelevant information.



(a) Archery (b) Floor Gymnastics

Figure 3: Visualization of dynamic attribute similarities learned by ST-AWF for Archery and Floor Gymnastics. The heatmaps illustrate how attribute similarities adapt across temporal frames to capture evolving action semantics. Darker colors indicate higher similarity scores.

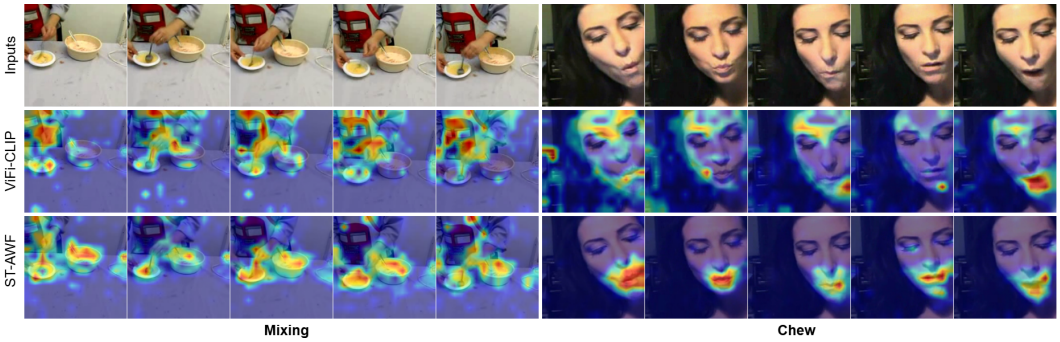


Figure 4: Qualitative comparison of attention maps between the baseline ViFi-CLIP and our proposed ST-AWF. The top row shows the input video frames for the "Mixing" and "Chew" actions.

5.2 Visualizations of Attention Maps

To qualitatively evaluate the representation learning capability of our model, we visualize the attention maps generated by our proposed ST-AWF and compare with ViFi-CLIP [30]. As illustrated in Figure 4, we select two video sequences: "Mixing" and "Chew". The visualizations show that ViFi-CLIP exhibits noisy attention, often focusing on irrelevant backgrounds or static body parts. In contrast, ST-AWF produces more concentrated and semantically meaningful attention, accurately highlighting hand-object interactions in "Mixing" and mouth movements in "Chew". These results demonstrate ST-AWF's ability to suppress spatial-temporal noise and capture fine-grained discriminative cues.

6 Conclusion

In this paper, we propose the ST-AWF framework, which enhances CLIP for video action recognition. We utilize LLM-generated spatial-temporal attributes to enrich textual representations. We further employ vision-guided prompt weighting to identify the key prompt. Finally, dual-attention cross-modal fusion is used to fully align the key prompt with video features. Experiments on multiple benchmarks demonstrate that our method achieves strong generalization and consistently outperforms competitive baselines. Limitations and possible future work are discussed in the Appendix.

7 Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (U22A2095, 62276281) and the Guangdong Basic and Applied Basic Research Foundation, China (2024A1515011882).

References

[1] Ahmad, S., Chanda, S., Rawat, Y.S.: T2l: Efficient zero-shot action recognition with temporal token learning. Transactions on Machine Learning Research (2025)

[2] Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv preprint arXiv:1808.01340 (2018)

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

[4] Chen, H., Huang, Z., Hong, Y., Wang, Y., Lyu, Z., Xu, Z., Lan, J., Gu, Z.: Efficient transfer learning for video-language foundation models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 29129–29138 (2025)

[5] Chen, S., Huang, D.: Elaborative rehearsal for zero-shot action recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13638–13647 (2021)

[6] Chen, T., Yu, H., Yang, Z., Li, Z., Sun, W., Chen, C.: Ost: Refining text knowledge with optimal spatio-temporal descriptor for general video recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18888–18898 (2024)

[7] Chen, Y.C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., Liu, J.: Uniter: Universal image-text representation learning. In: European conference on computer vision. pp. 104–120. Springer (2020)

[8] Chen, Y., Chen, D., Liu, R., Zhou, S., Xue, W., Peng, W.: Align before adapt: Leveraging entity-to-region alignments for generalizable video action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18688–18698 (2024)

[9] 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: Proceedings of the IEEE international conference on computer vision. pp. 5842–5850 (2017)

[10] Huang, X., Zhou, H., Yao, K., Han, K.: Froster: Frozen clip is a strong teacher for open-vocabulary action recognition. In: International Conference on Learning Representations (2024)

[11] Huang, Y., Shakeri, F., Dolz, J., Boudiaf, M., Bahig, H., Ben Ayed, I.: Lp++: A surprisingly strong linear probe for few-shot clip. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23773–23782 (2024)

[12] Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: International conference on machine learning. pp. 4904–4916. PMLR (2021)

[13] Jia, C., Luo, M., Chang, X., Dang, Z., Han, M., Wang, M., Dai, G., Dang, S., Wang, J.: Generating action-conditioned prompts for open-vocabulary video action recognition. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 4640–4649 (2024)

[14] Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European conference on computer vision. pp. 709–727. Springer (2022)

[15] Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: European conference on computer vision. pp. 105–124. Springer (2022)

[16] Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19113–19122 (2023)

[17] Kim, M., Han, D., Kim, T., Han, B.: Leveraging temporal contextualization for video action recognition. In: European Conference on Computer Vision. pp. 74–91. Springer (2024)

[18] Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: 2011 International conference on computer vision. pp. 2556–2563. IEEE (2011)

[19] Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 3045–3059 (2021)

[20] Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: International conference on machine learning. pp. 19730–19742. PMLR (2023)

[21] Li, S., Tang, H.: Multimodal alignment and fusion: A survey. arXiv preprint arXiv:2411.17040 (2024)

[22] Li, X.L., Liang, P.: Prefix-tuning: Optimizing continuous prompts for generation. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 4582–4597 (2021)

[23] Li, Z., Song, Y., Cheng, M.M., Li, X., Yang, J.: Advancing textual prompt learning with anchored attributes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3618–3627 (2025)

[24] Long, F., Li, X., Lv, J., Yang, H., Cheng, X., Li, P.: Bdc-clip: Brownian distance covariance for adapting clip to action recognition. In: Forty-second International Conference on Machine Learning (2025)

[25] Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. Neurocomputing **508**, 293–304 (2022)

[26] Nguyen, T., Gadre, S.Y., Ilharco, G., Oh, S., Schmidt, L.: Improving multimodal datasets with image captioning. Advances in neural information processing systems **36**, 22047–22069 (2023)

[27] Ni, B., Peng, H., Chen, M., Zhang, S., Meng, G., Fu, J., Xiang, S., Ling, H.: Expanding language-image pretrained models for general video recognition. In: European conference on computer vision. pp. 1–18. Springer (2022)

[28] Pan, J., Lin, Z., Zhu, X., Shao, J., Li, H.: St-adapter: Parameter-efficient image-to-video transfer learning. Advances in Neural Information Processing Systems **35**, 26462–26477 (2022)

[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: International conference on machine learning. pp. 8748–8763. PmLR (2021)

[30] Rasheed, H., Khattak, M.U., Maaz, M., Khan, S., Khan, F.S.: Fine-tuned clip models are efficient video learners. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6545–6554 (2023)

[31] 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)

[32] Tang, Y., Lin, Z., Wang, Q., Zhu, P., Hu, Q.: Amu-tuning: Effective logit bias for clip-based few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23323–23333 (2024)

[33] Wang, H., Liu, F., Jiao, L., Wang, J., Hao, Z., Li, S., Li, L., Chen, P., Liu, X.: Vilt-clip: Video and language tuning clip with multimodal prompt learning and scenario-guided optimization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5390–5400 (2024)

[34] Wang, H., Liu, F., Jiao, L., Wang, J., Li, S., Li, L., Chen, P., Liu, X.: Learning to prompt with refining text knowledge for zero-shot video action recognition. IEEE Transactions on Multimedia (2026)

[35] Wang, M., Xing, J., Jiang, B., Chen, J., Mei, J., Zuo, X., Dai, G., Wang, J., Liu, Y.: A multimodal, multi-task adapting framework for video action recognition. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5517–5525 (2024)

[36] Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021)

[37] Wang, X., Zhang, S., Cen, J., Gao, C., Zhang, Y., Zhao, D., Sang, N.: Clip-guided prototype modulating for few-shot action recognition. International Journal of Computer Vision **132**(6), 1899–1912 (2024)

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

[39] Weng, Z., Yang, X., Li, A., Wu, Z., Jiang, Y.G.: Open-vclip: Transforming clip to an open-vocabulary video model via interpolated weight optimization. In: International conference on machine learning. pp. 36978–36989. PMLR (2023)

[40] Xu, H., Ghosh, G., Huang, P.Y., Okhonko, D., Aghajanyan, A., Metze, F., Zettlemoyer, L., Feichtenhofer, C.: Videoclip: Contrastive pre-training for zero-shot video-text understanding. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 6787–6800 (2021)

[41] Yang, T., Zhu, Y., Xie, Y., Zhang, A., Chen, C., Li, M.: Aim: Adapting image models for efficient video understanding. In: International Conference on Learning Representations (2023), https://openreview.net/forum?id=CIoSZ_HKHS7

[42] Zhang, T., Kishore, V., Wu, F., Weinberger, K.Q., Artzi, Y.: Bertscore: Evaluating text generation with bert. arXiv preprint arXiv:1904.09675 (2019)

[43] Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International journal of computer vision **130**(9), 2337–2348 (2022)

[44] Zhu, W., Wang, Y., Li, H., Zhu, P.: Vtd-clip: Video-to-text discretization via prompting clip. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 13979–13987 (2026)

Appendix

In this appendix, we provide supplementary materials to further support the claims, methodology, and experimental details of our method. The contents are structured as follows: Section A details the datasets and implementation settings; Section B provides the demonstration of spatiotemporal attributes generated via LLM; Section C provides more experiments; Section D discusses both potential positive and negative societal impacts and Section E discusses the limitations of this work and potential future works.

A Experimental Details

A.1 Datasets

We conduct experiments on five widely used video action recognition datasets: Kinetics-400 [3], Kinetics-600 [2], UCF-101 [31], HMDB-51 [18], and Something-Something v2 [9].

Kinetics-400(K400). Kinetics-400 [3] is a large-scale dataset for video action recognition, consisting of approximately 240,000 training videos and 20,000 validation videos across 400 human action categories. Each video clip is approximately 10 seconds long and is sourced from YouTube.

Kinetics-600(K600). Kinetics-600 [2] is an expanded version of the K400 dataset, containing around 480,000 video clips and covering 600 action classes. The videos are divided into 390,000 for training, 30,000 for validation, and 60,000 for testing.

UCF-101. UCF-101 [31] is a widely used dataset for action recognition in the wild, consisting of 13,320 video clips from 101 categories It is collected from YouTube. There are three splits of the training and testing data.

HMDB-51. HMDB-51 [18] is a relatively small but challenging dataset containing 6,849 video clips distributed among 51 distinct action categories. The actions are collected from various sources, such as movies and online videos.

Something-Something v2 (SSv2). SSv2 [9] is a motion-centric dataset that focuses on fine-grained human-object interactions. It contains 220,847 videos across 174 categories. It is highly temporal biased compared to other datasets.

A.2 Task Settings and Hyper-parameters

In this section, we provide detailed experimental settings and hyper-parameter configurations for different benchmarks. For these tasks, we mainly follow the protocols of ViFi-CLIP [30].

Zero-shot Recognition. In the zero-shot transfer setting, we train the model on the source dataset (K400) and directly evaluate it on downstream target datasets (HMDB-51, UCF-101, and K600) without any further parameter updates. On UCF-101 and HMDB-51, the evaluations are conducted on three official splits of the test data. For K600, following [5], we use three splits randomly selected from 160 out of the 220 novel categories not present in K400. We report the average accuracy and standard deviation.

We adopt ViT-B/16 as the backbone. Each video is sampled to 32 frames for training and single-view inference is adopted during validation. We use the AdamW optimizer with β_1 = 0.9, β_2 = 0.98 and a weight decay of 1e-3. We only optimize the learnable prompts and the fusion model while keeping the backbone completely frozen. We set the context length to 16 for both visual and textual prompts, and the prompt depths for both modalities to 9. For our proposed attribute prompts, we employ 2 sets learnable prompts, each with a prompt length of 6. The model is trained for 10 epochs with batch size of 256 and learning rate of 8×10^{-6} with a cosine schedule.

Few-shot Recognition. Following the standard few-shot evaluation protocols established in prior works [30], we construct the training set by randomly sampling $K \in \{2, 4, 8, 16\}$ videos per action category across the UCF-101, HMDB-51, and SSv2 datasets. For evaluation, we use the first split of the validation set on UCF-101, HMDB-51, along with the full validation split for SSv2. For this task, we set the context length for both visual and textual prompts to 10, and their prompt depths to 12. The model is trained for 50 epochs with a batch size of 64 and learning rate of 8×10^{-3}. Other hyper-parameters and settings remain identical to those in the zero-shot experiments.

Base-to-Novel Generalization. In this task, the model is trained on base (seen) classes with each category 16 shots and subsequently evaluated on novel (unseen) categories. We conduct experiments on K400, HMDB-51, UCF-101, and SSv2, following the base and novel category splits established by ViFi-CLIP [30]. Three independent training splits are generated for each dataset and models are trained separately on each of these three splits, and we report the averaged results. The evaluations are consistently performed on the full official validation sets across all datasets. For this task, the model is trained for 11 epochs with a batch size of 64 and a base learning rate of 4×10^{-2}, while other hyper-parameters are kept identical to the zero-shot experiments.

B Prompts and Attributes Details

B.1 Prompts for LLMs

For the LLMs-Direct setting in Table 4b, we obtain the attributes by following the approach of AT-Prompt [23], which uses a comprehensive prompt to query LLM to generate dataset-level attributes. Specifically, we query the LLM with the following instruction: "`Please give me several general and independent attributes of the categories in {dataset}.`" This strategy produces a universal set of attributes shared across the entire dataset. Additionally, for the Irrelevant setting, we simply provide the model with the most common and uninformative terms across video datasets, namely "object" and "moving". However, both of these strategies are inherently coarse and fail to capture the fine-grained, class-specific spatial and temporal dynamics, which significantly limits their effectiveness compared to our proposed disentangled Spatial-Temporal design.

To ensure the LLM generates highly relevant and concise attributes, we carefully design our prompt templates. Rather than using generic prompts, we impose strict length constraints and provide explicit semantic context. This strategy is essential to prevent the generated text from exceeding the CLIP 77 token limit and make sure that the generated attributes are of high quality. For a given action category `{cls}`, the prompts used in our implementation are as follows:

- *Spatial Prompt*: "`Please provide` N `concise spatial attributes for the action: {cls}. Each attribute should be a short phrase focusing on key objects or scenes.`"
- *Temporal Prompt:* "`Please decompose the action: {cls} into` N `sequential steps. Each step should be expressed as a brief phrase capturing the core motion.`"

Table 5: Examples of LLM-generated spatial and temporal attributes for representative action categories. Spatial attributes describe key objects, body parts, or scenes, whereas temporal attributes describe the sequential motion process.

Action Category	**Spatial Attributes**	**Temporal Attributes**
Brush hair	hairbrush, hair, hand	picking up the hairbrush, holding the hairbrush, brushing the hair
Cartwheel	body, arms, legs, hands, feet	raising arms, kicking legs, placing hands on the ground, rotating body, landing on feet
Apply eye makeup	eye shadow, eyeliner, mascara, eyeshadow brush, eye	preparing the eye area, applying eye shadow, applying eyeliner, applying mascara
Barbequing	grill, charcoal, meat, tongs, marinade, spatula, vegetables	preparing the grill, lighting the charcoal, marinating the meat, cooking the meat and vegetables
Bartending	bartender, bar, customers, drinks, shaker, glass, ice	greeting customers, taking orders, mixing drinks, shaking cocktails, pouring drinks

B.2 Attributes Examples

Table 5 presents representative examples of the LLM-generated spatial and temporal attributes for several action categories. As demonstrated, the spatial attributes capture salient static visual cues,

while the temporal attributes break each action into fine-grained sequential motion steps. This clear semantic decoupling validates the effectiveness of our prompt design and forms a solid foundation for the subsequent multi-modal fusion process.

Importantly, the attribute generation is performed only once as an offline processing step. For each action category, we query the LLM to generate a fixed set of N spatial attributes and N temporal attributes, and the resulting attributes are then stored for subsequent training and evaluation. Therefore, the LLM is not invoked repeatedly during model training or inference. All video samples belonging to the same action category share the same pre-generated attribute set. In our experiments, we set $N = 3$, meaning that each action category is associated with three fixed spatial attributes and three fixed temporal attributes throughout all runs. Consequently, the attributes used by the model remain unchanged across repeated training or inference runs, avoiding additional randomness introduced by repeated LLM generation.

B.3 Ablation Studies on Different LLMs and Prompts

To further investigate the impact of attribute quality on model performance, we conduct additional analyses by comparing attributes generated by different LLMs and by the same LLM using different prompt templates.

Inspired by BERTScore [42], we introduce an **attribute-set semantic similarity** metric to quantify the semantic consistency between different attribute sets. For each action class, we use a frozen CLIP text encoder to encode the attributes from the two sets into a shared semantic space and compute the pairwise cosine similarity matrix. We then perform bidirectional nearest-neighbor matching between the two attribute sets, where each attribute is matched to its most semantically similar counterpart in the other set. The resulting similarities are aggregated in both directions to obtain a class-level semantic similarity score. Finally, we average the class-level scores across all action classes to derive the dataset-level semantic similarity. The detailed procedure is summarized in Algorithm 1.

Algorithm 1 Attribute-Set Semantic Similarity

Require: Attribute sets $\mathcal{A}_c = \{a_i^c\}_{i=1}^{M_c}$ and $\mathcal{B}_c = \{b_j^c\}_{j=1}^{N_c}$ for each class $c \in \{1, \dots, C\}$; frozen CLIP text encoder f_{text}

Ensure: Dataset-level semantic similarity S_{dataset}

for $c = 1$ to C **do**

$U_c \leftarrow \text{Normalize}(f_{\text{text}}(\mathcal{A}_c)), \quad V_c \leftarrow \text{Normalize}(f_{\text{text}}(\mathcal{B}_c))$

$R_{ij}^c \leftarrow \text{CosSim}(\mathbf{u}_i^c, \mathbf{v}_j^c), \quad \forall \mathbf{u}_i^c \in U_c, \forall \mathbf{v}_j^c \in V_c$

$S_c^{A \to B} \leftarrow \frac{1}{M_c} \sum_{i=1}^{M_c} \max_j R_{ij}^c$

$S_c^{B \to A} \leftarrow \frac{1}{N_c} \sum_{j=1}^{N_c} \max_i R_{ij}^c$

$S_c \leftarrow \frac{S_c^{A \to B} + S_c^{B \to A}}{2}$

end for

$S_{\text{dataset}} \leftarrow \frac{1}{C} \sum_{c=1}^{C} S_c$

return S_{dataset}

Table 6: Analysis of attributes generated by different LLMs. (a) Attribute-set semantic similarity between attributes generated by different LLMs and those generated by GPT-5. (b) Corresponding downstream classification performance averaged across UCF-101 and HMDB-51 datasets under the base-to-novel setting.

(a) Attribute-set semantic similarity

	UCF-101	HMDB-51	SSv2	K-400
GPT-5.6 Sol	0.928	0.914	0.881	0.921
Qwen3.5-9B	0.896	0.882	0.863	0.884
Llama 4	0.919	0.873	0.871	0.897

(b) Downstream performance

	Base	Novel	HM
GPT-5 (ours)	86.3	67.0	75.4
GPT-5.6 Sol	86.4	66.8	75.3
Qwen3.5-9B	85.7	66.9	75.1
Llama 4	86.5	66.5	75.6

Different LLMs. To evaluate the sensitivity of our method with different LLMs, we use GPT-5.6 Sol, Qwen3.5-9B, and Llama 4 Maverick to generate attribute texts. And we use the prompts

templates mentioned in Section B. Taking the GPT-5-generated attribute texts in the paper as the baseline reference, we apply Algorithm1 to compute the dataset-level semantic similarity between the reference attributes and those generated by each alternative LLM. The evaluation is conducted separately on four video datasets. Besides, we evaluate downstream classification performance under the base-to-novel setting. We conduct experiments on UCF-101 and HMDB-51 and report the average results across the two datasets. The results are reported in Table 6.

As shown in Table 6, attributes generated by different LLMs maintain high semantic consistency with those generated by GPT-5 across all four datasets, with similarity scores ranging from 0.873 to 0.928. Despite minor variations in the generated descriptions, the downstream performance remains highly stable: the average HM varies only from 75.1 to 75.6, compared with 75.4 for GPT-5. Similar trends are observed for both Base and Novel accuracy. These results indicate that the generated attributes are semantically consistent across different LLMs and, more importantly, that such variations have negligible influence on the final recognition performance, demonstrating the robustness of our method to the choice of LLM. This robustness may be attributed to the relatively common and semantically straightforward nature of the action categories considered in our benchmarks, for which generating informative spatial and temporal attributes does not require complex reasoning. As a result, even relatively lightweight LLMs can produce sufficiently reliable attribute descriptions for downstream recognition.

Different Prompts. We investigate the impact of different prompts using GPT-5 under the same experimental setting as the original paper. We construct a set of LLM prompts with the same objective: given a specific action category, the LLM is instructed to generate relevant spatial and temporal attribute descriptions. Examples of the LLM prompts are illustrated as follows:

> ***LLM-prompt1:***
> ***Spatial:*** *List N brief spatial attributes for the action* `{cls}`*, focusing on relevant objects and scenes.*
> ***Temporal:*** *Describe the action* `{cls}` *as N ordered sub-actions using brief phrases.*
>
> ***LLM-prompt2:***
> ***Spatial:*** *Identify N concise visual cues associated with the action* `{cls}`*, including key entities, tools, or environments.*
> ***Temporal:*** *Break down the action* `{cls}` *into N sequential motion stages, each expressed as a short phrase.*

We use the prompt in the paper as the baseline and compare it with the other prompts in terms of both dataset-level semantic similarity and base-to-novel classification performance, with the latter averaged over UCF-101 and HMDB-51. The results are reported in Table 7.

Table 7: Analysis of different LLM prompt templates. (a) Attribute-set semantic similarity between attributes generated using different prompt templates and those generated using our original prompt. (b) Corresponding downstream classification performance averaged across UCF-101 and HMDB-51 datasets under the base-to-novel setting.

(a) Attribute-set semantic similarity

	UCF-101	HMDB-51	SSv2	K-400
LLM-prompt1	0.921	0.932	0.902	0.938
LLM-prompt2	0.916	0.904	0.915	0.921

(b) Downstream performance

	Base	Novel	HM
Ours	86.3	67.0	75.4
LLM-prompt1	86.5	66.5	75.2
LLM-prompt2	86.0	67.2	75.5

As shown in Table 7, the attributes generated using different prompt templates remain highly consistent with those produced by the original prompt, with dataset-level semantic similarity scores ranging from 0.902 to 0.938 across the four datasets. This suggests that the generated spatial and temporal attributes are largely insensitive to moderate variations in prompt formulation. Consistent with this observation, the downstream classification performance also remains stable. The average HM varies only slightly from 75.2 to 75.5, compared with 75.4 using the original prompt, while the Base and

Novel accuracies exhibit similarly small fluctuations. These results indicate that our method is robust to prompt formulation and does not rely heavily on a specific prompt template for generating effective attribute guidance.

C Additional Experiments

Scaling to a Larger Backbone. To further examine the scalability of our method, we evaluate it with a larger CLIP ViT-L/14 backbone under the base-to-novel generalization setting on UCF-101. As shown in Table 8, our method consistently outperforms MSTA [4], improving the Base, Novel, and HM performance from 95.6, 76.8, and 85.2 to 96.0, 79.2, and 86.8, respectively. These results demonstrate that the effectiveness of our method is maintained when scaling to a larger visual backbone.

Video-Text Retrieval. We further evaluate the generalizability of our method beyond video recognition by conducting experiments on video-text retrieval. Following the evaluation protocol of ViFi-CLIP [30], we evaluate our method on the MSR-VTT 9K split. As shown in Table 8, our method improves R@1 from 44.8 to 46.2 and R@5 from 72.4 to 76.1 compared with ViFi-CLIP, demonstrating its effectiveness on video-text retrieval tasks.

Table 8: Comparison results.

(a) Base-to-Novel Generalization

	Base	Novel	HM
MSTA	95.6	76.8	85.2
Ours	96.0	79.2	86.8

(b) Video-Text Retrieval

	R@1	R@5
ViFi-CLIP	44.8	72.4
Ours	46.2	76.1

D Societal Impacts

Our work targets video action recognition and may benefit applications such as human–computer interaction, sports analysis, and assistive systems by improving the understanding of fine-grained human actions. At the same time, similar to other video understanding techniques, it may also introduce risks when deployed in real-world scenarios, especially in surveillance-related applications. Recognition errors, dataset bias, or low-quality LLM-generated attributes may lead to unfair performance or inappropriate inferences.

E Limitations and Future Work

Currently, ST-AWF relies on text-only large language models (LLMs) to generate semantic attributes for enhancing textual representations, without leveraging visual information during attribute generation. In future work, we plan to explore vision-language models (VLMs) to incorporate visual cues from video content, enabling the generation of more context-aware and visually grounded attributes for improved video understanding. In addition, the current prompting paradigm is constrained by the limited text length of the CLIP text encoder, which restricts the number of attribute words that can be incorporated. Future work could explore more efficient and flexible prompt paradigm to better utilize richer attribute descriptions under such constraints. Furthermore, although our method is currently designed for video action recognition, extending it to other video understanding tasks, such as video object detection, is a promising direction.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims presented in the abstract and introduction aligns exactly with our method in Section 3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Appendix E, we discuss the limitations and future works of our study.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

Justification: Our method does not include any theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provided detailed information of the experiment in Section 4.1, 4.2 and Appendix B to ensure that our experiment can be reproduced.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided an anonymous link to the code in the abstract, and the relevant information about the data is included in Appendix B and C.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We have described the training and testing details in Section 4.1 and Appendix B to better understand the results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiments do not include error bars, confidence intervals, or statistical significance tests. We focus on the main benchmarks, which we believe is sufficient to support the main claims of the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have reported the compute resources information, including the type of compute workers, memory, and execution time, in Section 4.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that the research conducted in this paper conforms in every respect to the guidelines.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed societal impacts of our work in Appendix D.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All existing assets are properly credited and used in compliance with their licenses as noted in the Appendix B.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Documentation is provided in the README file included with our anonymized code URL.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our research does not involve any study with human participants or crowdsourcing. Therefore, IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The LLM is an integral part of our proposed method. We have provided a detailed description of its usage in Section 3.2.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.