Leveraging Temporal Contextualization for Video Action Recognition

Minji Kim[1†] Dongyoon Han[3] Taekyung Kim[3*] Bohyung Han[1,2*]

[1]ECE & [2]IPAI, Seoul National University [3]NAVER AI Lab

Abstract. We propose a novel framework for video understanding, called Temporally Contextualized CLIP (TC-CLIP), which leverages essential temporal information through global interactions in a spatio-temporal domain within a video. To be specific, we introduce Temporal Contextualization (TC), a layer-wise temporal information infusion mechanism for videos, which 1) extracts core information from each frame, 2) connects relevant information across frames for the summarization into context tokens, and 3) leverages the context tokens for feature encoding. Furthermore, the Video-conditional Prompting (VP) module processes context tokens to generate informative prompts in the text modality. Extensive experiments in zero-shot, few-shot, base-to-novel, and fully-supervised action recognition validate the effectiveness of our model. Ablation studies for TC and VP support our design choices. Our project page with the source code is available at https://github.com/naver-ai/tc-clip.

Keywords: Video Action Recognition · Vision-Language Model

1 Introduction

Pretrained large-scale Vision-Language Models (VLMs) have shown remarkable generalization capability in video understanding and have emerged as promising tools even for zero-shot or open-vocabulary recognition tasks [11, 32, 48]. However, pretraining task-specific models using video-text pairs pose significant challenges, primarily due to substantial computational costs and excessive expense for annotated video-text data [40,43]. Consequently, recent studies in video understanding [4,10,14,29,34,39,41,42] have shifted their focus toward employing image-based VLMs such as CLIP [32] with fine-tuning for aligning video representations with text embeddings derived from category names.

Despite the success of CLIP in video recognition, existing approaches fail to model temporal information in the video feature learning process, as shown in Fig. 1(a)-(b). This limitation stems from the restrictive token interactions in the temporal axis. For instance, cross-frame attention approaches [29, 41], shown in Fig. 2(a), attempt to gather temporal information only through class tokens. Although VCLIP [42] incorporates patch-level details by bringing keys and values

[†]Work done during an internship at NAVER AI Lab.

[*]Corresponding authors.

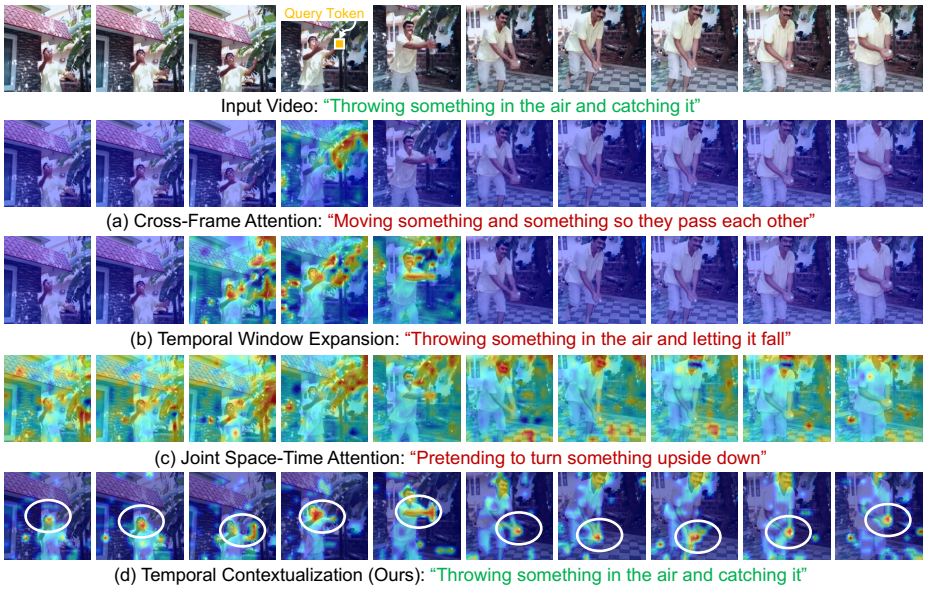

Fig. 1: Comparison of attention maps between various temporal modeling approaches. Both (a) and (b) fail to recognize actions in the latter frames, whereas (c) exhibits weak discriminability due to sparse attention on the background. In contrast, (d) our method successfully focuses on informative regions across all frames, leading to the accurate action recognition result.

from neighborhood frames in its self-attention operation as in Fig. 2(b), its temporal scope is too narrow. Furthermore, ViFi-CLIP [34] simply averages frame-wise representations with no inter-frame information exchanges. Such naïve approaches tend to bias the models towards static information in their representation learning (e.g., objects and backgrounds) and hamper learning temporal dynamics (e.g., motion and temporal variations). To ensure the global interactions of patch tokens in a spatio-temporal domain, one possible option is to consider every patch token from all frames as a reference during the encoding process as illustrated in Fig. 2(c).

Unfortunately, such a straightforward extension for temporally global interactions in VLMs pretrained with short image-text pairs witnesses extrapolation challenges [3, 31]; we have observed that a naïve extension of sequence length along the temporal axis degrades its discriminability substantially, as shown in Fig. 3(a). The joint space-time attention model spreads attention over many patches and fails to focus on informative tokens to recognize actions, resulting in suboptimal performance compared to the frame-wise attention baseline. Moreover, this approach suffers from heavy computational overhead due to numerous redundant and similar tokens, which often correspond to background regions.

This paper presents **Temporally Contextualized CLIP (TC-CLIP)**, a novel paradigm for extending CLIP to videos by encoding holistic video infor-

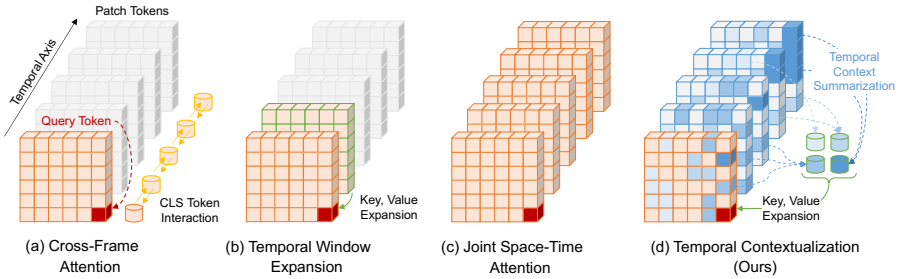

Fig. 2: Temporal information learning methods. Prior works consider temporal cues during the encoding process via (a) cross-frame attention [29,41] with `[CLS]` token interactions or (b) temporal window expansion [42] by adding adjacent frame tokens to key-value pairs. However, the former lacks patch-level interactions, while the latter limits the range of temporal interactions. (c) Joint space-time attention allows full interactions across all tokens, but it is costly and suboptimal in practice (see Fig. 3.) (d) Unlike prior approaches, our method aggregates pivotal tokens from a broader range yet efficiently for enhanced temporal integration into key-value pairs.

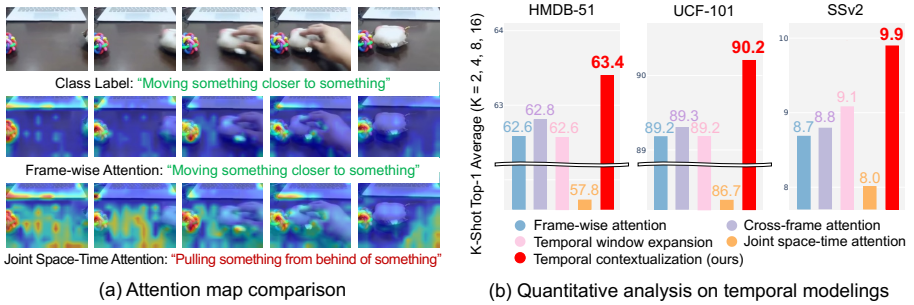

Fig. 3: Pitfall of joint space-time attention. (a) Extending CLIP's temporal sequence length degrades attention quality, presumably because it was not trained on such long sequences. (b) We compare the action recognition performance in the few-shot setting on diverse datasets. All existing methods fall behind our method.

mation through advanced temporal analysis. Specifically, our Temporal Contextualization (TC) pipeline summarizes global action cues into a small set of tokens, called context tokens, for reference during the encoding process. These context tokens act as additional key-value pairs for attention operations, presumably serving as temporal bridges that convey the video-level context. Our preliminary study, shown in Fig. 3(b), implies that existing methods illustrated in Fig. 2 offer minimal gains over the frame-wise attention, highlighting the need for enhanced token interactions.

In addition, a Video-conditional Prompting (VP) module generates instance-level textual prompts based on context tokens from the vision encoder. The VP module comprises cross-attention operations that adopt learnable text prompts as queries and context tokens as keys and values to inject video instance representations into video-conditional textual prompts. This strategy compensates for the lack of textual semantics in action recognition datasets, where textual

descriptions are limited to action class names (e.g., skateboarding, skydiving) without detailed narratives.

To verify the effectiveness and robustness of TC-CLIP, we perform extensive evaluations across diverse video recognition benchmarks. Quantitative comparisons in zero-shot, few-shot, base-to-novel, and fully-supervised settings show that the proposed approach outperforms the state-of-the-art methods by significant margins. We also provide an in-depth analysis of our design choices and the impact of each component in our model.

2 Proposed Method

2.1 Preliminary

We first review how CLIP [32] is used for video action recognition. In particular, we discuss the encoding procedure based on the vision and text encoders of CLIP, denoted by $\{f_{\theta_v}, f_{\theta_c}\}$, to obtain video and text features, $\{\mathbf{v}, \mathbf{c}\}$.

Video encoding. Suppose that there exists a video $V \in \mathbb{R}^{T\times H\times W\times 3}$ of a spatial resolution $H \times W$ with T sampled frames. Following the Vision Transformer (ViT) architecture [7], we first divide each frame into $P \times P$ non-overlapping patches and flatten them as a set of vectors $\{\mathbf{x}_{t,i} \in \mathbb{R}^{3P^2}\}_{i=1}^{N}$, where t is the frame index, i is the patch index, and $N = HW/P^2$ is the number of patches. Then, we derive a frame-level token sequence, $\mathbf{z}_t^0$ as follows:

$$\mathbf{z}_t^0 = [\mathbf{x}_{\text{cls}}, \mathbf{x}_{t,1}\mathbf{W}_{\text{emb}}, \mathbf{x}_{t,2}\mathbf{W}_{\text{emb}}, \ldots, \mathbf{x}_{t,N}\mathbf{W}_{\text{emb}}] + \mathbf{e}_{\text{pos}}, \tag{1}$$

where $\mathbf{x}_{\text{cls}} \in \mathbb{R}^{d_v}$ is a learnable classification embedding named `[CLS]` token, $\mathbf{W}_{\text{emb}} \in \mathbb{R}^{3P^2\times d_v}$ is a linear projection matrix, and $\mathbf{e}_{\text{pos}}$ is the spatial positional embedding. The CLIP vision encoder, $f_{\theta_v}(\cdot)$, sequentially encodes $\mathbf{z}_t^l$ at each layer $l \in \{1, \ldots, L_v\}$, which is given by

$$\mathbf{z}_t^l = f_{\theta_v}^l(\mathbf{z}_t^{l-1}). \tag{2}$$

We project the `[CLS]` token of the t^{th} frame, denoted by $\mathbf{z}_{t,0}^{L_v}$, onto a common vision-language latent space using a matrix $\mathbf{W}_{\text{vis}} \in \mathbb{R}^{d_v\times d_{vl}}$, *i.e.*, $\mathbf{v}_t = \mathbf{z}_{t,0}^{L_v}\mathbf{W}_{\text{vis}}$. Finally, the video representation $\mathbf{v}$ is obtained by averaging the per-frame representations $\mathbf{v}_t$ as $\mathbf{v} = \text{AvgPool}([\mathbf{v}_1, ..., \mathbf{v}_T])$.

Text encoding. Given a text description C—category name in our problem—for a video, the input of the text encoder, $\mathbf{c}^0$, is obtained by tokenizing words in the description and computing their word embedding vectors. In addition to the embeddings from category names, one can augment a sequence of prompt embeddings $\mathbf{p}^0$, which are obtained from either hand-crafted templates such as "`a photo of a`" or learnable prompt vectors. The CLIP text encoder, $f_{\theta_c}(\cdot)$, sequentially processes a sequence of text embeddings including prompt embeddings, denoted by $[\mathbf{p}^0, \mathbf{c}^0]$, and computes an intermediate representation at each layer $l \in \{1, \ldots, L_c\}$ as follows:

$$[\mathbf{p}^l, \mathbf{c}^l] = f_{\theta_c}^l([\mathbf{p}^{l-1}, \mathbf{c}^{l-1}]), \tag{3}$$

Table 1: Motivation: What is the proper format for reference tokens? We compare 16-shot training results using various types of reference tokens during the frame-level representation encoding process. Using context tokens consistently improves the baseline model regardless of the choice of the token aggregation function.

Type of reference tokens	HMDB-51	UCF-101	SSv2
Baseline (No reference tokens)	67.1	93.3	12.0
`[CLS]` tokens from all frames [29, 41]	67.2 (+0.1)	93.2 (−0.1)	12.3 (+0.3)
Patch tokens from adjacent frames [42]	67.8 (+0.7)	93.2 (−0.1)	12.8 (+0.8)
Patch tokens from all frames	63.3 (−3.8)	91.9 (−1.4)	12.0 (+0.0)
Context tokens — K-means [25]	68.0 (+0.9)	93.3 (+0.0)	13.1 (+1.1)
Context tokens — DPC-KNN [13]	67.9 (+0.8)	94.0 (+0.7)	14.3 (+2.3)
Context tokens — Bipartite soft matching [1, 15]	68.0 (+0.9)	93.8 (+0.5)	14.3 (+2.3)
Context tokens — Saliency-aware bipartite matching [5]	67.3 (+0.2)	93.7 (+0.4)	13.6 (+1.6)

where $f^l_{\theta_c}(\cdot)$ denotes the l^{th} layer of the CLIP text encoder. The final text representations $\mathbf{c}$ is obtained by projecting the `[EOS]` token from the last layer to the vision-language latent space using a matrix $\mathbf{W}_{\text{text}} \in \mathbb{R}^{d_l \times d_{vl}}$, *i.e.*, $\mathbf{c} = \mathbf{c}^{L_c}_{\text{eos}}\mathbf{W}_{\text{text}}$.

Video-text alignment. The similarity between video and text embeddings are formulated as $\text{sim}(\mathbf{v}, \mathbf{c}) = \frac{\langle \mathbf{v}, \mathbf{c} \rangle}{\|\mathbf{v}\|\|\mathbf{c}\|}$. During training, the goal is to maximize the similarity if V and C are matched and minimize otherwise. For inference, the category with the highest similarity is chosen as the prediction.

2.2 Motivation

Despite the successful generalization of CLIP for action recognition, its visual feature encoding process in Eq. (2) constrains the model's ability to capture comprehensive spatio-temporal dynamics because it only considers intra-frame token relationships. This limitation has led previous works to additionally incorporate reference tokens, denoted by $\mathbf{s}$, to encode the t^{th} frame tokens $\mathbf{z}_t$ as

$$\mathbf{z}^l_t = f^l_{\theta_v}(\mathbf{z}^{l-1}_t, \mathbf{s}^{l-1}). \tag{4}$$

However, their reference token designs are still limited due to insufficient spatio-temporal interaction range. For instance, cross-frame attention (Fig. 2(a)) [29,41] utilizes learnable global embedding vectors, *e.g.*, `[CLS]` tokens, from all frames to define the reference token as $\mathbf{s} = [\mathbf{z}_{1,0}, ..., \mathbf{z}_{T,0}]$, and temporal window expansion (Fig. 2(b)) [42], on the other hand, integrates neighboring frame patch tokens for $\mathbf{s} = [\mathbf{z}_{t-1}, \mathbf{z}_{t+1}]$. Note that the former lacks patch-level details whereas the latter captures temporal information only within a local range. Although incorporating all patch tokens from a whole video as a reference (Fig. 2(c)), $\mathbf{s} = [\mathbf{z}_1, ..., \mathbf{z}_T]$, might be conceptually reasonable, it is not practical due to the excessive number of tokens. Furthermore, this approach conflicts with the properties of CLIP pretrained on short image-text pairs and significantly degrades attention quality.

To this end, we compute a reference, $\mathbf{s} = \phi([\mathbf{z}_1, ..., \mathbf{z}_T])$, using a small set of context tokens that summarize a whole input video, where $\phi(\cdot)$ is a token

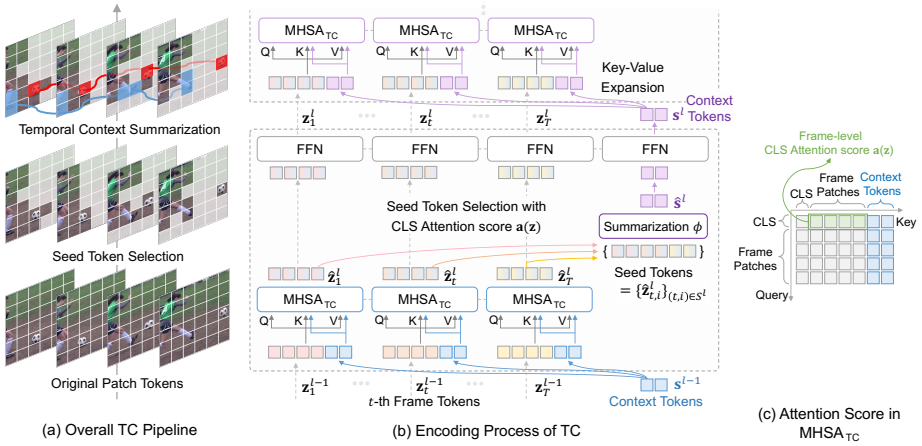

Fig. 4: Overview of Temporal Contextualization (TC). We first collect informative tokens from each frame and then aggregate relevant seed tokens to obtain context tokens. They are used as key-value pairs for the self-attention in the next layer.

aggregation function. This approach delivers the essential temporal information for the feature encoding while maintaining CLIP's effective sequence length. Our preliminary study, presented in Table 1, shows that using the proposed context tokens as a reference consistently outperforms the frame-wise attention baseline, whereas all other types of reference tokens only yield marginal gains or even suffer from performance degradation.

2.3 Temporal Contextualization (TC)

Based on our motivation, we propose Temporal Contextualization (TC), which consists of three steps: 1) informative token selection in each frame, 2) context summarization across spatio-temporal dimensions, and 3) context infusion to all tokens in the subsequent layer. Fig. 4 illustrates an overview of TC.

Informative token selection. Due to the many redundant tokens in a video, using all patches may be suboptimal for extracting desired temporal information. We only select the informative seed tokens based on each frame's attention scores obtained from self-attention operations. Specifically, given patch tokens $\{\mathbf{z}_{t,i}\}_{i=1}^{N}$ in the t^{th} frame, a set of attention scores $\{\mathbf{a}(\mathbf{z}_{t,i})\}_{i=1}^{N}$ is driven from the attentiveness of the `[CLS]` token with respect to other tokens, which is given by

$$\mathbf{a}(\mathbf{z}_t) = \text{Softmax}\Big(\frac{\mathbf{q}_{\text{cls}}\mathbf{K}_{\mathbf{z}_t}^{\mathsf{T}}}{\sqrt{d}}\Big), \tag{5}$$

where both the query $\mathbf{q}_{\text{cls}} = \mathbf{z}_{\text{t},0}\mathbf{W}_q \in \mathbb{R}^d$ and keys $\mathbf{K}_{\mathbf{z}_t} = \mathbf{z}_t\mathbf{W}_k \in \mathbb{R}^{(N+1)\times d}$ are given by linear projections of input $\mathbf{z}_t \in \mathbb{R}^{(N+1)\times d}$. In practice, our model yields multiple `[CLS]` attention scores from multi-head self-attention (MHSA)

operations and computes the average of the attention scores from all heads, *i.e.*, $\bar{\mathbf{a}}_{t,i} = \sum_{h=1}^{H} \mathbf{a}_{t,i}^{h}/H$, where $\mathbf{a}_{t,i}^{h} = \mathbf{a}^{h}(\mathbf{z}_{t,i})$ is the attention score for the i^{th} patch $\mathbf{z}_{t,i}$ in the t^{th} frame and H is the number of heads. Finally, we identify a set of seed token indices for the t^{th} frame, $\mathcal{S}_t$, by selecting n_s elements with the highest attention scores, where n_s is controlled by a hyperparameter $\alpha = n_s/N$.

Temporal context summarization. We describe how to connect the seed tokens derived from individual frames based on their relevance and identify a collection of context tokens. We first collect the seed tokens from all frames as $\{\hat{\mathbf{z}}_{t,i}\}_{(t,i)\in\mathcal{S}}$, where $\mathcal{S} = \{(t,i)|i \in \mathcal{S}_t, t = 1,\dots,T\}$ is a set of seed token indices from all frames and $\hat{\mathbf{z}}_{t,i}$ indicates an interim token encoded from $\mathbf{z}_{t,i}$ via the self-attention operation. Then we perform their spatio-temporal summarization by clustering and merging all the seed tokens as

$$\hat{\mathbf{s}} = \phi\big(\{\hat{\mathbf{z}}_{t,i}\}_{(t,i)\in\mathcal{S}}\big), \tag{6}$$

where $\hat{\mathbf{s}} \in \mathbb{R}^{k\times D}$ denotes a collection of the summarized tokens, which we call context tokens, and ϕ is a token aggregation function. While diverse token aggregation techniques are valid for TC (See Table 8), we adopt bipartite soft matching [1, 15] by default. Subsequently, the context tokens $\hat{\mathbf{s}}$ are fed into a feed-forward network (FFN).

Temporal context infusion. Finally, we infuse the summarized context to all patch tokens by modifying the self-attention function. The keys and values of self-attention in every frame are expanded to include context tokens as follows:

$$\text{Attention}_{\text{TC}}(\mathbf{z}_t, \mathbf{s}) = \text{Softmax}\Big(\frac{\mathbf{Q}_{\mathbf{z}_t}\big[\mathbf{K}_{\mathbf{z}_t}|\mathbf{K}_{\mathbf{s}}\big]^{\mathsf{T}}}{\sqrt{d}} + \mathbf{B}\Big)\big[\mathbf{V}_{\mathbf{z}_t}|\mathbf{V}_{\mathbf{s}}\big], \tag{7}$$

where $\mathbf{K}_{\mathbf{s}} = \mathbf{s}\mathbf{W}_k$ and $\mathbf{V}_{\mathbf{s}} = \mathbf{s}\mathbf{W}_v$ are linear projections of the context tokens $\mathbf{s} \in \mathbb{R}^{k\times d}$. Here, $\mathbf{B} \in \mathbb{R}^{(N+1)\times(N+k+1)}$ is a bias matrix that distinguishes between frame-level local information and video-level global information in the expanded key matrix as follows:

$$\mathbf{B}_{ij} = \begin{cases} b_{\text{local}} & \text{if } j \le N+1 \\ b_{\text{global}} & \text{otherwise,} \end{cases} \tag{8}$$

where b_{local} and b_{global} are learnable parameters and defined for multiple heads at each layer. We build our TC pipeline in a layer-wise manner, and thus the encoding process of each layer is expressed as

$$\hat{\mathbf{z}}_t^l = \begin{cases} \text{MHSA}(\text{LN}(\mathbf{z}_t^{l-1})) + \mathbf{z}_t^{l-1} & \text{if } l = 1 \\ \text{MHSA}_{\text{TC}}(\text{LN}(\mathbf{z}_t^{l-1}), \text{LN}(\mathbf{s}^{l-1})) + \mathbf{z}_t^{l-1} & \text{otherwise,} \end{cases} \tag{9}$$

$$\mathbf{z}_t^l = \text{FFN}(\text{LN}(\hat{\mathbf{z}}_t^l)) + \hat{\mathbf{z}}_t^l, \tag{10}$$

$$\mathbf{s}^l = \text{FFN}(\text{LN}(\hat{\mathbf{s}}^l)) + \hat{\mathbf{s}}^l, \tag{11}$$

where $\text{MHSA}_{\text{TC}}(\cdot,\cdot)$ denotes the MHSA operation based on Eq. (7) and $\text{LN}(\cdot)$ stands for the layer normalization function.

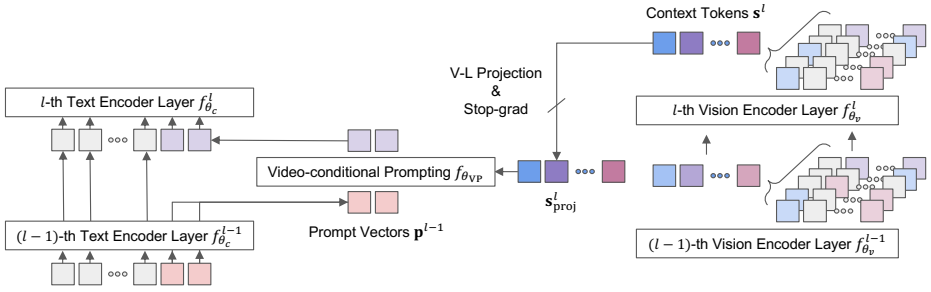

Fig. 5: Video-conditional Prompting (VP) module. Video information from the context tokens is injected into the text prompt vectors using a cross-attention mechanism, generating instance-level prompts that make up for the lack of textual semantics.

2.4 Video-conditional Prompting (VP)

The Video-conditional Prompting (VP) module further leverages the comprehensive video information derived from the visual domain for text encoding. We apply a cross-attention between prompt vectors and context tokens to enrich the information in the prompt vectors as illustrated in Fig. 5. Let $\mathbf{c}^{l-1}$ and $\mathbf{p}^{l-1}$ be class name tokens and learnable prompt vectors from the $(l-1)^{\text{th}}$ layer of the text encoder, respectively. We derive temporally contextualized prompt vectors $\tilde{\mathbf{p}}^{l-1}$ by passing the layer-normalized prompt tokens and context tokens through a cross-attention layer and an FFN layer as follows:

$$\mathbf{s}^l_{\text{proj}} = \text{SG}(\mathbf{s}^l \mathbf{W}_{\text{vis}}), \tag{12}$$

$$\hat{\mathbf{p}}^{l-1} = \text{MHCA}(\text{LN}_p(\mathbf{p}^{l-1}), \text{LN}_s(\mathbf{s}^l_{\text{proj}})) + \mathbf{p}^{l-1}, \tag{13}$$

$$\tilde{\mathbf{p}}^{l-1} = \text{FFN}(\text{LN}(\hat{\mathbf{p}}^{l-1})) + \hat{\mathbf{p}}^{l-1}, \tag{14}$$

where $\text{SG}(\cdot)$ is a stop-gradient function, $\mathbf{W}_{\text{vis}}$ is a weight matrix of CLIP to linearly project vision representations onto a common vision-language latent space, and $\text{MHCA}(\cdot,\cdot)$ is a multi-head cross-attention operation for interactions across modalities, accepting text prompt vectors as queries and vision features as keys and values. The VP module $f_{\theta_{\text{VP}}}(\cdot,\cdot)$ is defined by a composition of Eq. (13) and Eq. (14), and executed before the last layer of text encoder f_{θ_c}. Finally, the new formulation of our encoding process in the text modality is given by

$$[\mathbf{p}^l, \mathbf{c}^l] = \begin{cases} f^l_{\theta_c}([f_{\theta_{\text{VP}}}(\mathbf{p}^{l-1}, \mathbf{s}^l_{\text{proj}}), \mathbf{c}^{l-1}]) & \text{if } l = L_c \\ f^l_{\theta_c}([\mathbf{p}^{l-1}, \mathbf{c}^{l-1}]) & \text{otherwise.} \end{cases} \tag{15}$$

2.5 Training Objective

TC-CLIP learns to maximize the similarity of video representations $\mathbf{v}$ and text representations $\mathbf{c}$ for matching pairs in a mini-batch via the cross-entropy loss as $\mathcal{L} = -\sum_i \log \frac{\exp(\text{sim}(\mathbf{v}_i,\mathbf{c}_i)/\tau)}{\sum_j \exp(\text{sim}(\mathbf{v}_i,\mathbf{c}_j)/\tau)}$, where τ is a learnable temperature parameter. Our model is fully fine-tuned in an end-to-end manner.

Table 2: Comparison with state-of-the-arts on zero-shot action recognition. All the models are trained on Kinetics-400 and directly evaluated on other datasets. WE indicates the weight-space ensemble between the fine-tuned model and CLIP, adopted for all applicable models for fair comparisons. † denotes results reproduced using our implementation. The best results are in **bold-faced** numbers, and the second-best ones are underlined. Our results using the original and LLM-rephrased category names are highlighted in blue and purple, respectively.

Method	WE	HMDB-51	UCF-101	K600 (Top-1)	K600 (Top-5)	All (Top-1)
Vanilla CLIP [32]		40.8 ± 0.3	63.2 ± 0.2	59.8 ± 0.3	83.5 ± 0.2	54.6
ActionCLIP [39]†		49.1 ± 0.4	68.0 ± 0.9	56.1 ± 0.9	83.2 ± 0.2	57.7
A5 [14]		44.3 ± 2.2	69.3 ± 4.2	55.8 ± 0.7	81.4 ± 0.3	56.5
X-CLIP [29]		44.6 ± 5.2	72.0 ± 2.3	65.2 ± 0.4	86.1 ± 0.8	60.6
Vita-CLIP [41]		48.6 ± 0.6	75.0 ± 0.6	67.4 ± 0.5	-	63.7
ViFi-CLIP [34]†		52.3 ± 0.2	78.9 ± 1.1	70.7 ± 0.8	92.1 ± 0.3	67.3
TC-CLIP (Ours)		**53.7** ± 0.7	**80.4** ± 0.9	**72.7** ± 0.5	**93.2** ± 0.2	**68.9**
ActionCLIP [39]†	✓	51.9 ± 0.5	74.2 ± 1.0	67.5 ± 1.2	90.7 ± 0.1	64.5
ViFi-CLIP [34]†	✓	52.2 ± 0.7	81.0 ± 0.9	73.9 ± 0.5	93.3 ± 0.3	69.0
Open-VCLIP [42]	✓	53.9 ± 1.2	**83.4** ± 1.2	73.0 ± 0.8	93.2 ± 0.1	70.1
TC-CLIP (Ours)	✓	**54.2** ± 0.7	82.9 ± 0.6	**75.8** ± 0.5	**94.4** ± 0.2	**71.0**
Using LLM-based text augmentation						
MAXI [24]	✓	52.3 ± 0.7	78.2 ± 0.8	71.5 ± 0.8	92.5 ± 0.4	67.3
OST [4]	✓	55.9 ± 1.2	79.7 ± 1.1	75.1 ± 0.6	94.6 ± 0.2	70.2
FROSTER [10]	✓	54.8 ± 1.3	84.8 ± 1.1	74.8 ± 0.9	-	71.5
TC-CLIP (Ours)	✓	**56.0** ± 0.3	**85.4** ± 0.8	**78.1** ± 1.0	**95.7** ± 0.3	**73.2**

3 Experiments

We conduct experiments on 5 video benchmarks: Kinetics-400 [16] & 600 [2], HMDB-51 [21], UCF-101 [37], and Something-Something v2 (SSv2) [9]. Following [34], our evaluation protocols include zero-shot, few-shot, base-to-novel generalization, and fully-supervised action recognition tasks. We adopt CLIP with ViT-B/16 for all experiments and our baseline is ViFi-CLIP [34]. All models are trained using 4 NVIDIA Tesla V100 GPUs. More details are in the appendix.

3.1 Quantitative Comparison

We mainly compare our method with CLIP-based video recognition models: Vanilla CLIP [32], ActionCLIP [39], A5 [14], X-CLIP [29], Vita-CLIP [41], ViFi-CLIP [34], Open-VCLIP [42], OST [4], and FROSTER [10]. For the fair comparisons with approaches based on Large Language Model (LLM) with text augmentation [4, 10, 24], we produce two versions of our results: one using the original action category names (colored in blue) and the other adopting the LLM-rephrased category names obtained from FROSTER [10] (colored in purple). Note that experiments on the SSv2 dataset do not involve LLM-rephrasing.

Zero-shot action recognition. Table 2 exhibits the zero-shot generalization ability of several models, where they are trained on K-400 and then directly evaluated on individual datasets. For fair comparisons with recent models [4,

Table 3: Comparison with state-of-the-arts on few-shot action recognition. All the models are directly fine-tuned from CLIP. Our results using the original and LLM-rephrased category names are highlighted in blue and purple, respectively.

	HMDB-51				UCF-101				SSv2				All
Method	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=16	K=2	K=4	K=8	K=16	Avg.
Vanilla CLIP [32]	41.9	41.9	41.9	41.9	63.6	63.6	63.6	63.6	2.7	2.7	2.7	2.7	36.1
ActionCLIP [39]	47.5	57.9	57.3	59.1	70.6	71.5	73.0	91.4	4.1	5.8	8.4	11.1	46.5
A5 [14]	39.7	50.7	56.0	62.4	71.4	79.9	85.7	89.9	4.4	5.1	6.1	9.7	46.8
X-CLIP [29]	53.0	57.3	62.8	64.0	76.4	83.4	88.3	91.4	3.9	4.5	6.8	10.0	50.2
ViFi-CLIP [34]	57.2	**62.7**	64.5	66.8	80.7	85.1	90.0	92.7	6.2	7.4	8.5	12.4	52.9
TC-CLIP (Ours)	**57.3**	62.3	**67.3**	**68.6**	**85.9**	**89.9**	**92.5**	**94.6**	**7.3**	**8.6**	**9.3**	**14.0**	**54.8**
Using LLM-based text augmentation													
OST [4]	**59.1**	62.9	64.9	68.2	82.5	87.5	91.7	93.9	7.0	7.7	8.9	12.2	53.9
TC-CLIP (Ours)	58.6	**63.3**	**65.5**	**68.8**	**86.8**	**90.1**	**92.0**	**94.3**	**7.3**	**8.6**	**9.3**	**14.0**	**54.9**

Table 4: Comparison with state-of-the-arts on base-to-novel generalization. All the models are directly fine-tuned from CLIP. † results are taken from [10].

	K-400			HMDB-51			UCF-101			SSv2			All (Avg.)		
Method	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM	Base	Novel	HM
Vanilla CLIP [32]	62.3	53.4	57.5	53.3	46.8	49.8	78.5	63.6	70.3	4.9	5.3	5.1	49.8	42.3	45.7
ActionCLIP [39]	61.0	46.2	52.6	69.1	37.3	48.5	90.1	58.1	70.7	13.3	10.1	11.5	58.5	37.9	46.0
A5 [14]	69.7	37.6	48.8	46.2	16.0	23.8	90.5	40.4	55.8	8.3	5.3	6.4	53.7	24.8	33.9
X-CLIP [29]	74.1	56.4	64.0	69.4	45.5	55.0	89.9	58.9	71.2	8.5	6.6	7.4	60.5	41.9	49.5
ViFi-CLIP [34]	76.4	61.1	67.9	**73.8**	53.3	61.9	92.9	67.7	78.3	16.2	12.1	13.9	64.8	48.6	55.5
Open-VCLIP [42]†	76.5	62.6	68.9	70.3	50.4	58.9	94.8	77.5	85.3	16.0	11.0	13.0	64.4	50.4	56.5
TC-CLIP (Ours)	**78.9**	**63.6**	**70.4**	73.3	**54.1**	**62.2**	**95.5**	**78.0**	**85.9**	**17.5**	**13.4**	**15.2**	**66.3**	**52.3**	**58.5**
Using LLM-based text augmentation															
FROSTER [10]	77.8	64.3	70.4	**74.1**	58.0	65.1	95.3	80.0	87.0	**18.3**	12.2	14.6	**66.4**	53.6	59.3
TC-CLIP (Ours)	**79.1**	**65.4**	**71.6**	73.3	**59.1**	**65.5**	**95.4**	**81.6**	**88.0**	17.5	**13.4**	**15.2**	66.3	**54.9**	**60.1**

10, 24, 42], we employ weight-space ensembling (WE) for all applicable models except those freezing a backbone during fine-tuning. Specifically, the weights of both vision and text encoders are linearly interpolated between CLIP and the fine-tuned model as $\theta_w = (1-w) \cdot \theta_{\text{CLIP}} + w \cdot \theta_{\text{fine-tuned}}$. TC-CLIP consistently outperforms others across all datasets, showing its superior generalization ability.

Few-shot action recognition. We verify the learning capacity of our method under a challenging few-shot scenario. In Table 3, models are directly fine-tuned from CLIP on each dataset using K-shot samples, where K is 2, 4, 8, and 16. TC-CLIP achieves the best performance with large margins from ViFi-CLIP [34].

Base-to-novel generalization. Similarly, models are directly fine-tuned from CLIP using the base classes of each dataset and evaluated for both base and novel classes. Table 4 reports top-1 accuracies on the base and novel classes with their harmonic mean (HM). TC-CLIP performs the best on the novel classes and HM across all datasets, especially showing solid results on the SSv2 dataset.

Fully-supervised action recognition. Table 5 shows performance comparison results under the fully-supervised setting, where the models are trained and

Table 5: Fully-supervised action recognition results on Kinetics-400. Views means (temporal clips) × (spatial crops), and F denotes number of frames.

Method	Top-1	Top-5	F	Views
ActionCLIP [39]	83.8	96.2	32	10 × 3
X-CLIP [29]	84.7	96.8	16	4 × 3
Vita-CLIP [41]	82.9	96.3	16	4 × 3
ViFi-CLIP [34]	83.9	96.3	16	4 × 3
OST [4]	83.2	-	16	1 × 1
TC-CLIP (Ours)	**85.2**	**96.9**	16	4 × 3

Table 6: Computational costs with the average top-1 accuracies of all protocols. The Throughput per view (TP) is measured on a single A6000 GPU. § denotes that TC is partly applied to the 4^{th}, 8^{th}, and 12^{th} layers of the vision encoder.

Method	Zero	Few	B2N	Full	Params	GFLOPs	TP
ActionCLIP [39]	64.5	46.5	46.0	83.8	143.7M	567	20
X-CLIP [29]	60.6	50.2	49.5	84.7	169.7M	288	36
Vita-CLIP [41]	63.7	-	-	82.9	161.8M	307	30
ViFi-CLIP [34]	69.0	52.9	55.5	83.9	**124.3M**	**285**	**38**
Open-VCLIP [42]	70.1	-	56.5	-	**124.3M**	308	29
TC-CLIP (Ours)	**71.0**	**54.8**	58.5	**85.2**	127.5M	304	24
TC-CLIP (Ours)§	70.7	54.4	**58.6**	84.9	127.5M	291	34

Table 7: Component-wise ablations on the zero-shot setting. Δ denotes the average top-1 accuracy gain over baseline.

	Without weight-space ensembling				With weight-space ensembling			
Case	HMDB-51	UCF-101	K-600	All (Δ)	HMDB-51	UCF-101	K-600	All (Δ)
Baseline	52.3 ± 0.2	78.9 ± 1.1	70.7 ± 0.8	67.3	52.2 ± 0.7	81.0 ± 0.9	73.9 ± 0.5	69.0
(a) +TC	53.6 ± 0.2	78.6 ± 1.0	71.8 ± 0.7	68.0 (+0.7)	54.3 ± 0.6	81.9 ± 1.0	75.5 ± 1.0	70.6 (+1.6)
(b) +VP	53.2 ± 0.8	80.5 ± 0.7	71.6 ± 0.9	68.4 (+1.1)	53.4 ± 0.8	82.0 ± 0.9	74.7 ± 0.7	70.0 (+1.0)
(c) +TC+VP	53.7 ± 0.7	80.4 ± 0.9	72.7 ± 0.5	68.9 (+1.6)	54.2 ± 1.1	82.9 ± 0.9	75.8 ± 0.4	71.0 (+2.0)

evaluated both on the K-400 dataset. TC-CLIP achieves top-1 accuracy of 85.2% in the validation split, improving 1.3%p over our baseline ViFi-CLIP [34].

Computational cost. Table 6 compares the computational cost with the average accuracy of all tasks. We introduce a lightweight implementation of TC-CLIP (denoted by §), where TC is only applied to the 4^{th}, 8^{th}, and 12^{th} layers of the vision encoder. Despite its reasonable cost, it performs best across all protocols by significant margins. In particular, compared to Open-VCLIP [42], this lightweight version improves accuracy by 0.6%p and 2.1%p in the zero-shot and base-to-novel tasks, respectively, while maintaining 17.2% higher throughput.

3.2 Analysis and Discussion

This section examines the design choices and impact of each component in our model: Temporal Contextualization (TC) and Video-conditional Prompting (VP). We mainly adopt the zero- and few-shot settings and report the average of top-1 accuracy with $K = 2, 4, 8, 16$ for the K-shot setup. In addition to the analyses discussed in this subsection, we present more analyses and qualitative results in the supplementary document.

Component-wise ablation. Table 7 shows the impact of TC and VP on our baseline in the zero-shot setting. Integrating TC gives an average gain of 0.7%p over the baseline and the gap increases to 1.6%p after adopting WE; WE is more favorable to our approach than the baseline. Adopting VP also leads to a substantial gain of 1.1%p, highlighting its own contribution. When both VP

Table 8: Effect of TC with various token aggregation strategies. TC consistently outperforms the frame-wise attention baseline across several different token selection and merging methods. K-shot action recognition results are reported with the top-1 accuracy averaged over $K = 2, 4, 8, 16$. Default settings are marked in gray.

(a) Seed token selection strategy.

Case	HMDB	UCF	SSv2	All	(Δ)
Baseline	62.6	89.2	8.7	53.5	
No selection	62.8	89.8	9.7	54.1	(+0.6)
Head-wise key norm	62.3	89.8	9.8	54.0	(+0.5)
Averaged key norm	62.5	89.4	9.3	53.7	(+0.2)
Head-wise CLS attn.	63.4	89.9	9.7	54.3	(+0.8)
Averaged CLS attn.	63.4	90.2	**9.9**	**54.5**	(+1.0)
Patch saliency [5]	62.9	**90.3**	9.6	54.2	(+0.7)
ATS [8]	**63.5**	**90.3**	9.8	**54.5**	(+1.0)

(b) Context token summarization strategy.

Case	HMDB	UCF	SSv2	All	(Δ)
Baseline	62.6	89.2	8.7	53.5	
No merge	57.2	85.6	7.7	50.2	(−3.3)
Random merge	58.8	87.1	7.5	51.2	(−2.3)
K-means [25]	62.1	89.7	9.0	53.6	(+0.1)
DPC-KNN [13]	63.3	**90.2**	9.8	54.4	(+0.9)
Bipartite soft matching [1, 15]	**63.4**	**90.2**	**9.9**	54.5	(+1.0)
Bipartite w/ attention weights	62.9	89.8	**9.9**	54.2	(+0.7)
Bipartite w/ saliency weights [5]	62.4	89.9	9.6	54.0	(+0.5)

Table 9: TC design ablation. We report K-shot training results where the top-1 accuracy in each dataset is averaged over $K = 2, 4, 8, 16$. Bias is defined in Eq. (8).

(a) Positional embedding design.

Case	HMDB	UCF	SSv2	All
Spatial embedding	62.9	90.0	9.8	54.2
Joint space-time embedding	63.2	**90.2**	9.8	54.4
Spatial embedding + Bias	**63.4**	**90.2**	**9.9**	**54.5**
Joint embedding + Bias	62.9	**90.2**	9.8	54.3

(b) Seed token ratio α.

α	HMDB	UCF	SSv2	All
0.2	62.6	90.1	9.8	54.2
0.3	**63.4**	90.2	**9.9**	54.5
0.4	63.2	**90.4**	9.8	54.5
0.5	63.3	90.3	9.8	54.5
0.6	63.1	90.2	9.8	54.4

(c) Context token k.

k	HMDB	UCF	SSv2	All
16	63.1	89.3	9.1	53.8
32	63.6	89.9	9.4	54.3
64	**63.7**	90.1	9.7	**54.5**
96	63.4	**90.2**	**9.9**	**54.5**
128	62.8	90.1	**9.9**	54.3

and TC are applied to the baseline, an average improvement goes up to 1.6%p, which finally leads to 2.0%p gain after applying WE.

Token aggregation strategies. In Table 8, we verify the effectiveness of TC across diverse token aggregation methods. Experiments are conducted on the few-shot setting using the baseline model with TC. (a) While TC still works well without token selection, we observe that collecting informative seed tokens based on token importance, such as attention or saliency scores, improves the quality of encoded tokens by suppressing the background. (b) Directly using the seed tokens without merging reduces performance due to the extrapolation issue. The degradation with random merging also highlights the requirement of token clustering based on relevance. Finally, consistent gains from various token merging approaches verify the robustness of TC regardless of algorithms.

Positional embedding. Table 9(a) shows that using the proposed learnable bias (Eq. (8)) with spatial positional embedding yields the best result. We conjecture that the bias effectively consolidates the local frame-level information and global video-level information in a layer-wise and head-wise manner.

Number of seed and context tokens. While TC is not sensitive to the choice of α, as shown in Table 9(b), we picked $\alpha = 0.3$ as our default value, *i.e.*, using 30% of total tokens as seed tokens. In Table 9(c), the context token number k is chosen to set a modest amount of merging degree.

Table 10: Text prompting design ablation on the zero-shot setting. All the models are evaluated without the weight ensemble.

Case	Use context tokens?	HMDB-51	UCF-101	K-600	All (Δ)
Baseline		52.3 ± 0.2	78.9 ± 1.1	70.7 ± 0.8	67.3
(a) Learnable prompt vectors		52.4 ± 0.4	78.4 ± 1.3	70.6 ± 0.7	67.1 (−0.2)
(b) Video-conditional prompting		53.2 ± 0.8	80.4 ± 0.7	71.6 ± 0.9	68.4 (+1.1)
(c) Video-conditional prompting	✓	53.7 ± 0.7	80.4 ± 0.9	72.7 ± 0.5	68.9 (+1.6)
(d) Vision-text late-fusion	✓	53.7 ± 0.7	79.0 ± 0.7	70.9 ± 0.6	67.9 (+0.6)

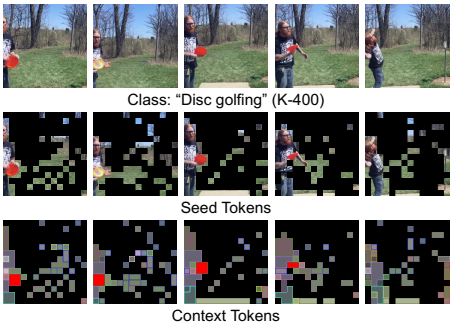

Fig. 6: Context token visualization. TC-CLIP selects the informative seed tokens and summarizes them into context tokens across frames. The disc (red) is merged into one token over the video.

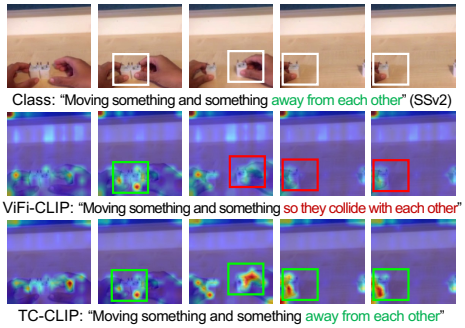

Fig. 7: Attention visualization. While ViFi-CLIP fails to attend to the hands moving away and misinterprets the action as colliding, TC-CLIP correctly predicts by exploiting temporal consistency.

Text prompting design. In Table 10, we observe that (a) a naïve integration of learnable prompt vectors without video instance conditioning is not particularly helpful for the zero-shot transferability, rather decreasing the average accuracy. In contrast, (b) employing VP design with `[CLS]` tokens consistently improves the accuracy across all datasets, and (c) using context tokens further enhances the performance, resulting in a 1.6%p gain. We also compare VP with (d) vision-text late-fusion design, *i.e.*, the cross-attention of context tokens and the final representation of the text embedding. This design performs worse in UCF-101 and K-600 datasets than our VP, verifying the effectiveness of our design choice.

Context token visualization. Fig. 6 visualizes the seed tokens and context tokens from the last layer of the vision encoder in TC-CLIP. In this video, the informative regions regarding the action of *disc golfing* in each frame, including the person and the disc, are selected as seed tokens. To visualize each context token, we colorize its corresponding source token positions using the average color of the input video patches of that region. Note that a single context token (highlighted in red) successfully tracks the disc across multiple frames.

Attention visualization. Fig. 7 visualizes the attention map of ViFi-CLIP [34] and TC-CLIP on the SSv2 dataset. In this video, where two hands grab objects and then move away, ViFi-CLIP [34] fails to attend to the hands from the middle of the sequence and misinterprets the action as *colliding with each other*. In

contrast, TC-CLIP considers the temporal context across the sequence by its design, and thus consistently attends to the hands throughout the entire video and correctly predicts the action as *moving away from each other*.

4 Related Work

Token aggregation. Recent studies on token aggregation aim to reduce the number of tokens given to image Transformers [1, 8, 19, 20, 23, 26, 28, 30, 33, 46, 47, 49] and video Transformers [5, 6, 22, 35, 36, 38] for efficient inference. While some of these approaches train additional networks for token selection [20,33,38] or fusion [30,36], we focus on parameter-free approaches, categorized into token pruning and merging. Pruning-based methods [8,23,46,47] eliminate uninformative tokens by measuring their importance using a metric such as a self-attention score, whereas merging-based methods combine tokens with large semantic similarity into single units using clustering algorithms such as k-means [28], DPC-KNN [49], and bipartite soft matching [1, 22, 35]. To minimize information loss, several studies [5,6,26] consider both token importance and similarities as aggregation criteria. While our primary goal is not to improve efficiency, we employ both pruning and merging techniques to connect relevant tokens and summarize essential contexts within videos. Although there are semantic segmentation studies [27, 44, 45] that link relevant spatial data, they rely on learnable cluster centers with slot- or cross-attention blocks, and thus differ from our approach.

Prompt learning. Several studies on prompt learning [12, 14, 17, 34, 41, 50, 51] transfer VLMs to downstream tasks by optimizing a discrete set of prompt vectors. In video recognition, [14] has introduced text prompt tuning, while ViFi-CLIP [34] and Vita-CLIP [41] perform prompting in both vision and text branches. However, these prompt vectors are separately optimized and not shared across the modalities. In image recognition, Co-CoOp [50] performs an instance-conditional prompt tuning by explicitly conditioning text prompts on the `[CLS]` tokens from image instances. MaPLe [17] learns multi-modal prompting by sharing layerwise context prompts for both branches. Unlikely, we generate video-conditional prompts by utilizing contextualized tokens as vision inputs and injecting summarized video information into text prompt vectors.

5 Conclusion

We have introduced TC-CLIP, a novel video understanding paradigm that leverages holistic video information within the encoding process. Unlike prior approaches that access only a limited range of tokens, the proposed temporal contextualization summarizes informative tokens from the entire video and utilizes them for attention operations. While these tokens are employed to infuse temporal information on the vision side, they also serve as a source for video-conditional text prompts, thus enhancing the instance-wise context on the text side. Extensive experiments and analyses on diverse benchmarks and evaluation protocols demonstrate the superiority of TC-CLIP and justify its design choices.

Acknowledgements

Experiments are based on the NAVER Smart Machine Learning NSML [18] platform. This research was partly supported by the Bio & Medical Technology Development Program of the National Research Foundation (NRF) (No. 2021M3A9E4080782) and the IITP grants [No. RS-2021-II212068; No. RS-2021-II211343] funded by the Korean government (MSIT).

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022) 5, 7, 12, 14
2. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. arXiv preprint arXiv:1808.01340 (2018) 9
3. Chen, S., Wong, S., Chen, L., Tian, Y.: Extending context window of large language models via positional interpolation. arXiv preprint arXiv:2306.15595 (2023) 2
4. 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: CVPR (2024) 1, 9, 10, 11
5. Choi, J., Lee, S., Chu, J., Choi, M., Kim, H.J.: vid-tldr: Training free token merging for light-weight video transformer. In: CVPR (2024) 5, 12, 14
6. Ding, S., Zhao, P., Zhang, X., Qian, R., Xiong, H., Tian, Q.: Prune spatio-temporal tokens by semantic-aware temporal accumulation. In: CVPR (2023) 14
7. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021) 4
8. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive token sampling for efficient vision transformers. In: ECCV (2022) 12, 14
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: ICCV (2017) 9
10. Huang, X., Zhou, H., Yao, K., Han, K.: Froster: Frozen clip is a strong teacher for open-vocabulary action recognition. In: ICLR (2024) 1, 9, 10
11. 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: ICML (2021) 1
12. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: ECCV (2022) 14
13. Jiang, J., Chen, Y., Meng, X., Wang, L., Li, K.: A novel density peaks clustering algorithm based on k nearest neighbors for improving assignment process. Physica A: Statistical Mechanics and its Applications **523**, 702–713 (2019) 5, 12
14. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: ECCV (2022) 1, 9, 10, 14
15. Karp, R.M., Vazirani, U.V., Vazirani, V.V.: An optimal algorithm for on-line bipartite matching. In: Proceedings of the twenty-second annual ACM symposium on Theory of computing (1990) 5, 7, 12

16. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017) 9
17. Khattak, M.U., Rasheed, H., Maaz, M., Khan, S., Khan, F.S.: Maple: Multi-modal prompt learning. In: CVPR (2023) 14
18. Kim, H., Kim, M., Seo, D., Kim, J., Park, H., Park, S., Jo, H., Kim, K., Yang, Y., Kim, Y., et al.: Nsml: Meet the mlaas platform with a real-world case study. arXiv preprint arXiv:1810.09957 (2018) 15
19. Kim, T., Han, D., Heo, B.: Morphing tokens draw strong masked image models. arXiv preprint arXiv:2401.00254 (2023) 14
20. Kong, Z., Dong, P., Ma, X., Meng, X., Niu, W., Sun, M., Shen, X., Yuan, G., Ren, B., Tang, H., et al.: Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In: ECCV (2022) 14
21. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: ICCV (2011) 9
22. Li, X., Ma, C., Yang, X., Yang, M.H.: Vidtome: Video token merging for zero-shot video editing. arXiv preprint arXiv:2312.10656 (2023) 14
23. Liang, Y., GE, C., Tong, Z., Song, Y., Wang, J., Xie, P.: EVit: Expediting vision transformers via token reorganizations. In: ICLR (2022) 14
24. Lin, W., Karlinsky, L., Shvetsova, N., Possegger, H., Kozinski, M., Panda, R., Feris, R., Kuehne, H., Bischof, H.: Match, expand and improve: Unsupervised finetuning for zero-shot action recognition with language knowledge. In: ICCV (2023) 9
25. Lloyd, S.: Least squares quantization in pcm. IEEE Transactions on Information Theory **28**(2), 129–137 (1982) 5, 12
26. Long, S., Zhao, Z., Pi, J., Wang, S., Wang, J.: Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers. In: CVPR (2023) 14
27. Luo, H., Bao, J., Wu, Y., He, X., Li, T.: Segclip: Patch aggregation with learnable centers for open-vocabulary semantic segmentation. In: ICML (2023) 14
28. Marin, D., Chang, J.H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token pooling in vision transformers. arXiv preprint arXiv:2110.03860 (2021) 14
29. 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: ECCV (2022) 1, 3, 5, 9, 10, 11
30. Pan, Z., Zhuang, B., He, H., Liu, J., Cai, J.: Less is more: Pay less attention in vision transformers. In: AAAI (2022) 14
31. Press, O., Smith, N.A., Lewis, M.: Train short, test long: Attention with linear biases enables input length extrapolation. In: ICLR (2022) 2
32. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 1, 4, 9, 10
33. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. In: NeurIPS (2021) 14
34. Rasheed, H., Khattak, M.U., Maaz, M., Khan, S., Khan, F.S.: Fine-tuned clip models are efficient video learners. In: CVPR (2023) 1, 2, 9, 10, 11, 13, 14
35. Ren, S., Chen, S., Li, S., Sun, X., Hou, L.: Testa: Temporal-spatial token aggregation for long-form video-language understanding. arXiv preprint arXiv:2310.19060 (2023) 14
36. Ryoo, M., Piergiovanni, A., Arnab, A., Dehghani, M., Angelova, A.: Tokenlearner: Adaptive space-time tokenization for videos. In: NeurIPS (2021) 14

37. 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) 9
38. Wang, J., Yang, X., Li, H., Liu, L., Wu, Z., Jiang, Y.G.: Efficient video transformers with spatial-temporal token selection. In: ECCV (2022) 14
39. Wang, M., Xing, J., Liu, Y.: Actionclip: A new paradigm for video action recognition. arXiv preprint arXiv:2109.08472 (2021) 1, 9, 10, 11
40. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X., Li, X., Chen, G., Chen, X., Wang, Y., et al.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. arXiv preprint arXiv:2307.06942 (2023) 1
41. Wasim, S.T., Naseer, M., Khan, S., Khan, F.S., Shah, M.: Vita-clip: Video and text adaptive clip via multimodal prompting. In: CVPR (2023) 1, 3, 5, 9, 11, 14
42. 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: ICML (2023) 1, 3, 5, 9, 10, 11
43. 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. arXiv preprint arXiv:2109.14084 (2021) 1
44. Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. In: CVPR (2022) 14
45. Xu, J., Hou, J., Zhang, Y., Feng, R., Wang, Y., Qiao, Y., Xie, W.: Learning open-vocabulary semantic segmentation models from natural language supervision. In: CVPR (2023) 14
46. Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., Zhang, L., Xu, C., Sun, X.: Evo-vit: Slow-fast token evolution for dynamic vision transformer. In: AAAI (2022) 14
47. Yin, H., Vahdat, A., Alvarez, J.M., Mallya, A., Kautz, J., Molchanov, P.: A-vit: Adaptive tokens for efficient vision transformer. In: CVPR (2022) 14
48. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021) 1
49. Zeng, W., Jin, S., Liu, W., Qian, C., Luo, P., Ouyang, W., Wang, X.: Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In: CVPR (2022) 14
50. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: CVPR (2022) 14
51. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. IJCV (2022) 14