Spatio-Temporal Context Prompting for Zero-Shot Action Detection

Wei-Jhe Huang[1] Min-Hung Chen[2] Shang-Hong Lai[1]
[1]National Tsing Hua University, Taiwan [2]NVIDIA

Abstract

Spatio-temporal action detection encompasses the tasks of localizing and classifying individual actions within a video. Recent works aim to enhance this process by incorporating interaction modeling, which captures the relationship between people and their surrounding context. However, these approaches have primarily focused on fully-supervised learning, and the current limitation lies in the lack of generalization capability to recognize unseen action categories. In this paper, we aim to adapt the pretrained image-language models to detect unseen actions. To this end, we propose a method which can effectively leverage the rich knowledge of visual-language models to perform Person-Context Interaction. Meanwhile, our Context Prompting module will utilize contextual information to prompt labels, thereby enhancing the generation of more representative text features. Moreover, to address the challenge of recognizing distinct actions by multiple people at the same timestamp, we design the Interest Token Spotting mechanism which employs pretrained visual knowledge to find each person's interest context tokens, and then these tokens will be used for prompting to generate text features tailored to each individual. To evaluate the ability to detect unseen actions, we propose a comprehensive benchmark on J-HMDB, UCF101-24, and AVA datasets. The experiments show that our method achieves superior results compared to previous approaches and can be further extended to multi-action videos, bringing it closer to real-world applications. The code and data can be found in ST-CLIP.

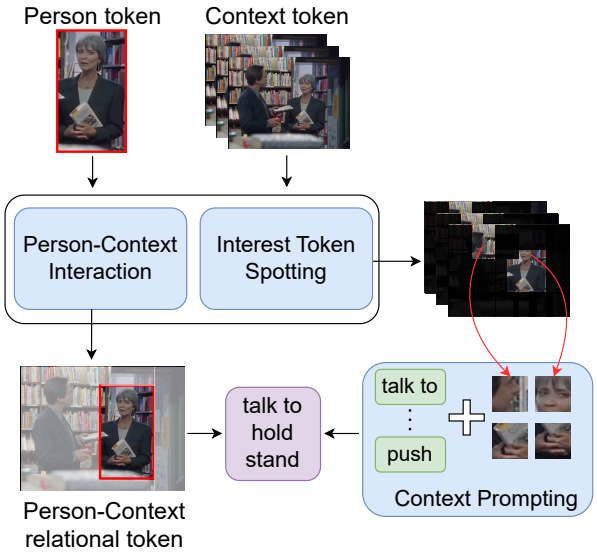


Figure 1. **Overview of our method.** We aim to transfer the knowledge of CLIP to detect unseen actions. We leverage the pretrained knowledge to model the interaction between people and their surrounding context. Besides, the Interest Token Spotting mechanism utilizes the knowledge to find the tokens most relevant to a person, then the Context Prompting uses these visual tokens to augment the text contents, which can make them easier to distinguish.

1. Introduction

The task of spatial-temporal action detection is to detect people and recognize their respective actions in both space and time, which holds broad applications in various fields, including self-driving cars, sports analysis, and surveillance. Recently, the rise of 3D CNN backbones [6,27,28] has strengthened the capabilities of representation learning in spatial-temporal context, which has greatly improved the performance of action detection. Furthermore, some recent studies have extended their focus by incorporating attention-based relation modeling [5, 20, 26]. These approaches aim to model the relationship between individuals and their surrounding environment, including other people, objects, and the contextual scene. By integrating more interaction information into the person feature, a more comprehensive representation of their actions is achieved, consequently enhancing the accuracy of action classification. However, these methods primarily center around fully supervised learning, limiting their capability to detect only the action classes included in the training phase. In real-world applications, numerous actions beyond the training classes are bound to occur. Therefore, our approach aims to push the boundaries further by detecting unseen actions in zero-

shot scenarios, alleviating the considerable labor-intensive efforts associated with the annotation process.

In recent years, visual-language models [12,21,35] have gradually become the models of choice in zero-shot video understanding due to their strong generalization capability. Nevertheless, the predominant focus in this domain is currently on video classification [13,19,29,31,32] and temporal action detection [13, 18], where the entire video or a short clip is considered for action classification. [10] is most similar to our goal, which is to detect individual unseen actions. However, the scenario they handle is too simple. They only process videos with single actions and target specific unseen labels, limiting their ability to effectively evaluate the robustness of the method. In contrast, our goal is to detect a variety of unseen actions and extend the method to videos containing multiple actions.

Towards the aforementioned goal, we propose a novel framework called ST-CLIP, which adapts CLIP [21] to Zero-Shot Spatio-Temporal Action Detection in both visual and textual aspects, as shown in Figure 1. In terms of vision, we propose to utilize the visual knowledge embedded in CLIP to perform Person-Context Interaction. This approach enables us to grasp the relationship between individuals and their surrounding context without the necessity for additional interaction modules, thereby preserving the generalization capabilities of CLIP and streamlining the interaction modeling process. In the textual domain, given that the class names in the dataset offer limited semantic information, the ambiguity between different labels may degrade the quality of the classification results. Our objective is to enhance the textual content through effective prompting. Inspired by [19], we design a multi-layer Context Prompting module, which incrementally utilizes visual clues from spatio-temporal context to augment text descriptions, thereby increasing the discrimination capability. Furthermore, given that real-world scenarios often involve multiple individuals concurrently performing different actions, we further introduce Interest Token Spotting, which aims to identify context tokens most relevant to each person's actions. Subsequently, these tokens are utilized in the prompting process to generate a description that aptly captures each individual's situation.

In order to assess the effectiveness of our method, we refer to [10] and propose a more complete benchmark on J-HMDB, UCF101-24 and AVA datasets. For the first two datasets, we conduct cross-validation with varying train/test label combinations. This approach, as opposed to [10], which exclusively experiments with a specific label split, provides a more comprehensive assessment of the method's robustness. For experiments on AVA, where a single video may involve multiple actions, we randomly select certain videos that lack common classes for training, then the subsequent evaluation will focus on assessing the performance of detecting these unseen actions. The experimental results demonstrate that, in comparison to other zero-shot video classification methods, our approach exhibits superior performance on J-HMDB and achieves competitive results on UCF101-24. Furthermore, experiments on AVA demonstrate that our method can detect various unseen actions individually within the same video, affirming its potential extension to real-world applications. To summarize, our contributions are as follows:

- We propose a novel method ST-CLIP that fully leverages the visual-language model to capture the relationship between people and the spatial-temporal context, without training extra interaction modules.
- We devise a multi-layer Context Prompting module that employs both low-level and high-level context information to prompt class names, enriching the semantic content. In addition, we introduce an Interest Token Spotting mechanism to identify tokens most relevant to individuals for prompting, thereby generating text features that are unique to each person.
- We propose a complete benchmark on J-HMDB, UCF101-24, and AVA datasets to evaluate performance on Zero-Shot Spatio-Temporal Action Detection. The experiments demonstrate the strong generalization capabilities of our method, and show the ability to individually detect unseen actions within the same video.

2. Related Work

Spatio-Temporal Action Detection. Typical action detection methods mostly use the two-stage pipeline, which means first localizing people in a video, and then performing action classification based on the features of these people. Most of these methods utilize additional person detectors like Faster R-CNN [22] to generate actor bounding boxes, which are used to perform RoIAlign [9] on the video features generated by the 3D CNN backbone to obtain the person features. While [6] directly utilizes naive actor features to classify actions, [5, 20, 26] further exploit relation modeling to combine more information about human and environmental interactions. Besides, some methods [2, 7, 25, 33] optimize the two-stage networks by a joint loss in an end-to-end framework. There are also several methods designed to handle similar tasks in unsupervised scenarios. The goal of [23] is to localize the time and space of actions in the video without using bounding box annotations during training. [1] propose a domain adaptation framework for action detection; however, their approach is limited to detecting fixed action classes, whereas our goal is to operate in an open-vocabulary setting.

Video Understanding with Visual-Language Models. Recently, large-scale visual-language models such as CLIP [21], ALIGN [12], and Florence [35] have demonstrated their usability to different visual-language tasks including image captioning [17], video-text retrieval [4] and scene text detection [34]. Due to a shared feature space that effectively aligns the visual and text domains, an increasing number of methods [13, 19, 29, 31, 32] choose to perform zero-shot video classification based on these foundation models. The main focus of these works is to design temporal modeling to adapt the image encoder to the video domain and to develop ways to prompt the text. On this basis, [10] processes zero-shot spatio-temporal action detection to further subdivide the scope of action classification to the individual level. They proposed extracting people and objects in the image and using different interaction blocks to model the relationship between them. Besides, they will use each person's interaction feature to prompt labels. To evaluate their performance, they selected specific unseen actions to detect on the two datasets, J-HMDB and UCF101-24. However, this benchmark is still not close enough to real-world scenarios. First, they did not extensively test a variety of unseen labels. Second, the videos in both datasets contain only single actions. Considering this, we propose a more complete benchmark to evaluate performance on multi-action videos, aiming for more practical applications.

3. Proposed Method

3.1. Preliminary: Visual-Language Model

As our method exploits the pretrained knowledge of the CLIP model, we briefly review the formulation of image and text encoders in this section. For the image encoder of ViT architecture [3], given an image $I \in \mathbb{R}^{H\times W\times 3}$ with height H and width W, it will be split into $N = \frac{H}{P} \times \frac{W}{P}$ patches, where the patch size is $P \times P$, then each patch will obtain its token through patch embedding. In this process, the Conv2D with kernel size $P \times P$ and output channel size D will be used to generate patch tokens $x \in \mathbb{R}^{N\times D}$. After that, an extra learnable token $x_{cls} \in \mathbb{R}^{D}$ is concatenated for classification. The input tokens for the transformer encoder layer is given by:

$$X = [x_{cls}, x_1, x_2, \ldots, x_N] + e \tag{1}$$

where e is the positional encoding. The classification token x_{cls} output from the last encoder layer is often regarded as an image feature. Similarly, the text encoder is also a transformer architecture, while it first tokenizes the text into embeddings, and then uses the EOS token of the last encoder layer output as the text feature.

3.2. Overall Architecture

The overview of our ST-CLIP is shown in Figure 2. As a two-stage framework, our model takes the person detected from video frames by a human detector as input and outputs the corresponding action classification results. Given an image with the person bounding boxes, we first extract these portions and obtain person-specific tokens through the image encoder. In this process, we utilize the adapter to make these person tokens more suitable for subsequent interaction modeling, which will be discussed later. Given the continuous nature of actions, in addition to considering a person's information, we sample neighboring frames to construct a spatial-temporal context, thereby capturing information across different spaces and times. To obtain these context tokens, we conduct temporal modeling on the patch tokens of different frames, enabling the tokens to aggregate information over this period of time.

Subsequently, to fully leverage CLIP's visual knowledge, we jointly input person and context tokens into the image encoder. The Multi-Head Self Attention (MHSA) in each encoder layer is employed to guide us in achieving the following three objectives: *(i)* performing further spatial modeling on the input context tokens to obtain spatial-temporal tokens. *(ii)* modeling person-person and person-context interaction through the mutual influence between tokens. *(iii)* identifying the interest tokens most relevant to each person's actions through attention weight. On the textual side, we initially utilize the CLIP text encoder to generate the original text features for class names. Then, followed by each image encoder layer, the Context Prompting layer will use context tokens to prompt each label. In this process, considering that the videos in J-HMDB and UCF101-24 only contain a single action, we can treat all context tokens as relevant to this action. Hence, we use all context tokens for prompting, resulting in everyone in the same frame sharing the same text features. However, in AVA, to discern different actions by multiple individuals, we utilize context tokens that each person deems important (*i.e.*, interest tokens) to prompt their respective text features. Finally, we use the person tokens and label features output by the last image encoder layer and Context Prompting layer to calculate the cosine similarities, which are used as the action classification scores.

In the training process, in order to retain the generalization capability of CLIP for zero-shot tasks, we freeze the pretrained weights in the image and text encoders, and only train our additional learnable modules. Besides, we insert the LoRA trainable matrices into the Feed-Forward Network (FFN) of each image encoder layer, which can further adapt the CLIP model to detect actions without affecting its well-aligned visual-language features.

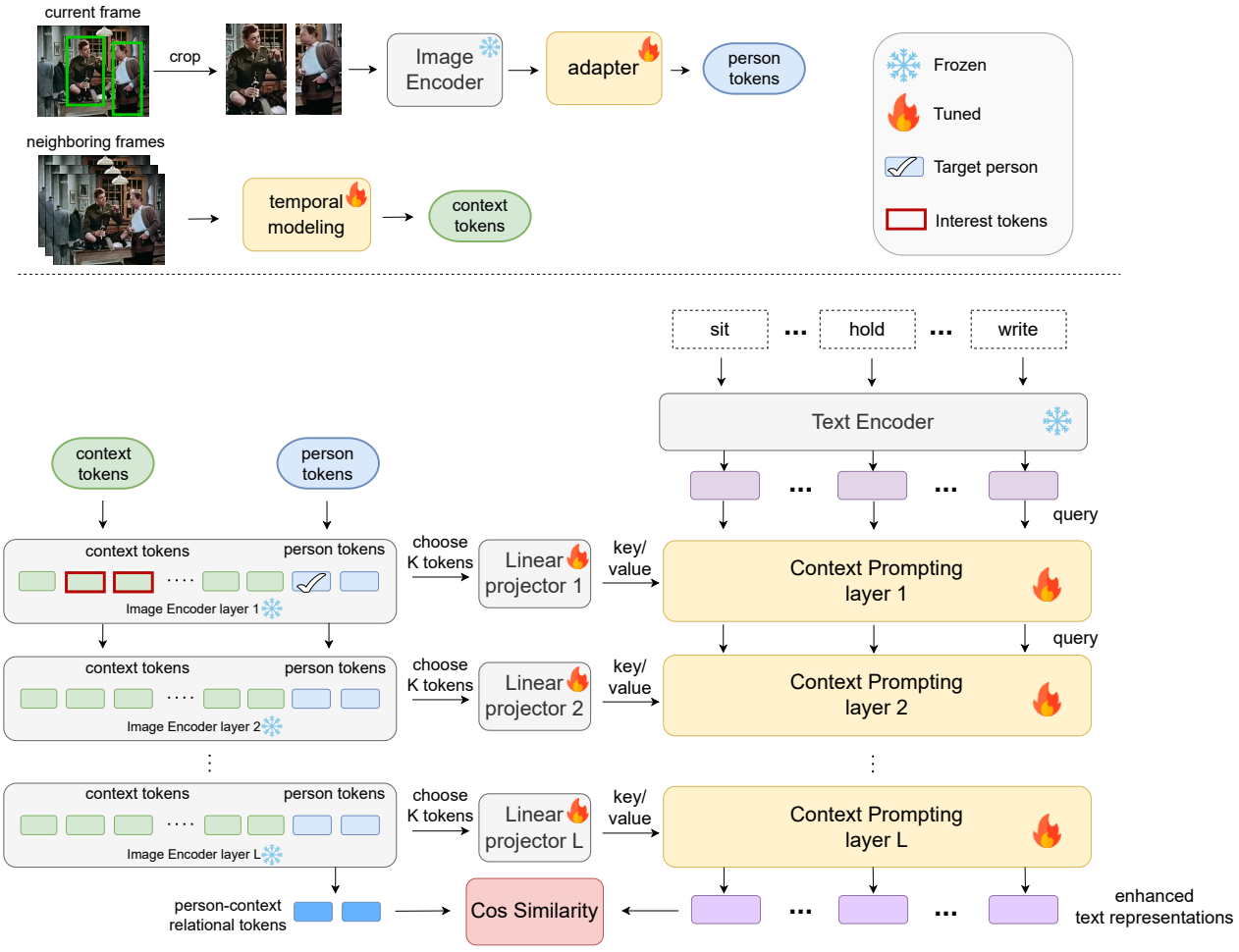


Figure 2. **ST-CLIP framework.** We first extract the person tokens for the person bounding boxes detected from each frame. Then, we perform temporal modeling on the neighboring frames to obtain the context tokens. After that, we leverage the CLIP's visual knowledge to perform person-context interaction on these tokens. In addition, we utilize the attention weight in each encoder layer to find the interest tokens for each person, then the Context Prompting layer will use these visual tokens to prompt the class names. Finally, the cosine similarities between person-context relational tokens and the label prompting features determine the classification scores for the actions.

3.3. Person-Context Interaction

As mentioned earlier, to utilize spatial-temporal context and facilitate the recognition of continuous actions, we sample T neighboring frames before and after the current frame. We first conduct temporal modeling on these frames to consolidate information at different times. Subsequently, we leverage the spatial modeling capability of the image encoder to further fuse these visual contents in both space and time, which results in the generation of spatial-temporal tokens. Our temporal modeling is shown in Figure 3. First, we use CLIP's pretrained patch embedding to obtain the patch tokens of each frame $X_t = [x_{t,1}, x_{t,2}, \ldots, x_{t,N}] \in \mathbb{R}^{N \times D}$, where $t \in \{1, \ldots, T\}$ denotes the frame index, N is the number of patch tokens, and D is the token dimension. Then we gather the tokens of each frame into $[X_1, X_2, \ldots, X_T] \in \mathbb{R}^{T \times N \times D}$. After that, we utilize MHSA to model the relationship between patch tokens at the same position in different frames $Z_i = [x_{1,i}, x_{2,i}, \ldots, x_{T,i}] \in \mathbb{R}^{T \times D}$, where $i \in \{1, \ldots, N\}$, as follows:

$$\begin{aligned} \bar{Z}_i &= Z_i + e^{temp} \\ \hat{Z}_i &= \bar{Z}_i + MHSA(LN(\bar{Z}_i)) \\ \tilde{Z}_i &= AvgPool(\hat{Z}_i), \end{aligned} \tag{2}$$

where e^{temp} is the temporal encoding and LN stands for layer normalization. After temporal modeling, we can obtain context tokens $\tilde{Z}_i \in \mathbb{R}^D$ at each position

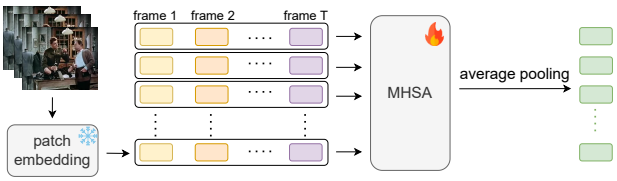


Figure 3. **Temporal modeling.** We apply self-attention along the temporal dimension to fuse the information.

$i \in \{1, \ldots, N\}$ that have aggregated temporal information, which will be used to model the interaction with a person.

Regarding the person tokens, we initially utilize the image encoder to obtain features for each person. Considering that these person features have undergone multiple transformer encoder layers, they are relatively high-level compared to the aforementioned context tokens, which are only at the patch-embedding level. To enhance the utilization of self-attention in modeling relationships among all tokens, we employ an adapter to adjust person features to the same level as other context tokens. The adapter is a straightforward two-layer FFN commonly used in the transformer encoder layer, and we have found it to be effective in adapting these person features. Specifically, we generate person tokens in the following ways:

$$\tilde{P}_i = P_i + FFN(LN(P_i)) \tag{3}$$

where P_i is the person feature output from the image encoder and $\tilde{P}_i$ is the person token we will use for subsequent interaction modeling.

3.4. Interest Token Spotting

In a multi-action dataset like AVA, the same timestamp may encompass various actions performed by multiple individuals, and all the context tokens will contain information about different actions. Therefore, we introduce Interest Token Spotting, a mechanism that employs a personal perspective to extract context tokens most relevant to each individual's actions. Subsequently, we utilize these tokens for prompting to generate personalized text features. In this process, we also leverage CLIP's pretrained visual knowledge to find each person's interest tokens. To be more specific, we exploit the attention weight calculated by the MHSA in each image encoder layer as an indicator of the token importance. The Multi-Head Self Attention will first calculate an attention map $M_i \in \mathbb{R}^{C \times C}$ based on the query and key of each head, where $i \in \{1, \ldots, num\_heads\}$, and C is the total number of person tokens and context tokens. Then, we average the attention maps of each head to obtain an importance score matrix $M \in \mathbb{R}^{C \times C}$. In this matrix, each row represents the importance of every token to a certain token. For instance, $M(i, j)$ represents how important token j is to token i. Based on this, we use a person's token as the row index, and select the top K highest among all C importance scores. These selected K indexes are the token positions that the person is interested in. After that, we pass each token through the MHSA and FFN of this encoder layer, and use the selected K indexes to obtain the interest tokens.

3.5. Context Prompting

The Context Prompting layer primarily consists of Cross-Attention (CA), where text features serve as the query, and the context tokens act as key and value. This setup facilitates the gradual absorption of visual information into the text content. In AVA, we further narrow down the scope of the extracted information for each person to their personal interest tokens. Given an image with B detected individuals, we need to classify their actions into N_L possible labels. First, we assign the same set of original text features that are obtained by the CLIP text encoder to these B people. Subsequently, we employ a linear projector to project each person's interest tokens to the same dimension as the text features. Following this, both the interest tokens and the text features are sent to the Context Prompting layer for prompting. The following equations describe how it works:

$$\begin{aligned} \bar{F_T} &= F_T + CA(F_T, F_I), \\ \hat{F_T} &= \bar{F_T} + FFN(\bar{F_T}), \\ \tilde{F_T} &= F_T + \rho \hat{F_T}, \end{aligned} \tag{4}$$

where $F_T \in \mathbb{R}^{B \times N_L \times D}$ consists of the text features, $F_I \in \mathbb{R}^{B \times K \times D}$ are the interest tokens, $\rho \in \mathbb{R}^D$ is a learnable weight vector, and $\tilde{F_T}$ will be input to the next prompting layer. For J-HMDB and UCF101-24, we straightforwardly use all context tokens for prompting, making all B individuals in an image utilize the same text features $F_T \in \mathbb{R}^{N_L \times D}$ for similarity calculation.

4. Datasets and Benchmarks

We establish benchmarks for zero-shot spatial-temporal action detection on three popular datasets: J-HMDB, UCF101-24, and AVA. For the first two datasets, we further extend the settings of [10] to include more diverse unseen actions. Besides, we also build benchmark on AVA which is more representative of real-world scenarios. We use frame mAP with 0.5 IoU threshold in all the benchmarks for evaluation. More details about the label splits are described in the supplementary materials.

4.1. ZS-JHMDB

J-HMDB dataset [11] is a subset of the HMDB51 dataset. It has 21 classes and 928 videos. The videos are

trimmed and there are totally 31,838 annotated frames in these videos. To assess the generalization capability of an action detection method, the zero-shot evaluation necessitates that the model has not seen samples related to test classes during the training process, which means the training and testing labels are disjoint. In this scenario, we refer to the evaluation settings proposed by [10], which exploits random sampling to take 75% action classes for training, and the remaining 25% for testing. However, they only employ a specific label split for evaluation, which is inadequate for fully measuring the effectiveness of the method. Instead, we perform cross-validation on multiple label splits as follows: we split J-HMDB into 4 label splits, each split has 15 training classes and 6 testing classes, and the testing classes in each split are disjoint (part of split 4 will overlap with split 1). The split 1 is the same as the split used in [10].

4.2. ZS-UCF

UCF101-24 dataset is a subset of UCF101 [24]. It consists of 3207 videos from 24 action classes, and each video contains a single action. In this benchmark, we employ the same setting as in ZS-JHMDB, which also divides all classes into 75% for training and 25% for evaluation. The label split 1 is also the same as used in [10].

4.3. ZS-AVA

AVA [8] is a large-scale action detection dataset that contains multiple actions within a single video. It consists of 235 videos for training and 64 videos for validation. Each video lasts for 15 minutes and is annotated by sampling one keyframe per second. For AVA, since the same video contains multiple actions, and some action categories exist in multiple videos, it becomes challenging to find a sufficient amount of training data and testing data with disjoint labels. Instead, we randomly select some training videos, ensuring that they all lack samples of the same classes. These missing classes are then treated as unseen classes for evaluation. During the evaluation phase, we test all classes in the validation videos, but the focus is solely on evaluating the performance of unseen classes. Under this setting, we propose three splits. When the amounts of training and testing data for these splits are nearly the same, we select different combinations of three action types — pose action, object interaction, and person interaction as unseen classes, allowing for a more comprehensive evaluation.

5. Experiments

5.1. Experimental Setup

Person Detector: In the following experiments, we employ Faster R-CNN [14] with a ResNet-50-FPN [15] backbone pretrained on MSCOCO [16] for person detection. For J-HMDB and AVA, we directly inference on the test data with pretrained person detector. For UCF101-24, since the images have lower resolution, we further use the ground truth of training classes to finetune the person detector for 10 epochs in each label split. Besides, to remove the false positives, we keep the detected box with the highest confident score S in each frame, then we select the boxes with scores higher than $S-T$ from the rest, where T is 0.001 for J-HMDB and 0.7 for both UCF101-24 and AVA.

Hyperparameters: For all the experiments on J-HMDB and UCF101-24, we employ ViT-B/16 as our CLIP backbone and use the same hyper-parameters as follows: Training for 3K iterations with a batch size of 8. We use SGD as our optimizer and the base learning rate is set to 2.5e-4. As for AVA, we use the ViT-L/14 backbone and train the model for 20K iterations with a base learning rate of 4e-4.

5.2. Compared Methods

In addition to comparing our approach with iCLIP [10], which addresses the same task as ours, we also evaluate it against the following methods.

Baseline: we follow [10] to implement a naive baseline for comparison. For a frame with detected individuals, the baseline utilizes the pretrained image encoder of CLIP to extract the image feature of this frame. Subsequently, it calculates the cosine similarities with the text features of each class name, which are then considered as the action classification scores for these individuals. Since the baseline regards people in the same frame as having the same actions, we also implement the person crop method for multi-action videos. This involves cropping out parts of each person to obtain their respective image features for classification.

ViCLIP [30]: We also experiment with a video-language model for individual action classification. First, we extract the video feature map using a video encoder and apply average pooling along the temporal dimension. Then, for each person, we use their bounding box to perform ROIAlign and max pooling on the feature map, obtaining a person-specific feature for classification.

Video classification methods [13, 19, 29, 31]: since each video in both ZS-JHMDB and ZS-UCF contains only a single action, a straightforward approach is to conduct action detection through video classification. These methods can initially classify the entire video into an action class and then consider all detected individuals in the video as performing this action. As for ZS-AVA, the main distinction between our method and these approaches is our capability to detect different actions within the same video, which is difficult to achieve. Firstly, in the training process of these methods, providing a fixed video label is challenging due to the presence of numerous different actions. Additionally, during the inference stage, these methods tend to detect that everyone in the video has the same action. To study the feasibility of these methods, we further narrow the scope

of classification from the entire video to tracklets to avoid misclassifying different actions into the same category.

5.3. Zero-Shot Spatial-Temporal Action Detection

Method	Frame mAP@0.5						
	split 1	split 2	split 3	avg	H	avg*	H*
baseline	8.67	8.57	2.77	6.67	7.38	10.09	10.58
baseline (person crop)	8.22	5.05	3.04	5.44	6.42	7.55	8.63
iCLIP [10]	4.04	9.08	1.91	5.01	7.59	6.91	10.37
Vita-CLIP [31]	4.25	4.29	0.64	3.06	4.49	10.45	13.94
ST-CLIP (Ours)	**12.85**	**10.17**	**4.01**	**9.01**	**11.41**	**11.76**	**15.05**

Table 1. **Evaluation on ZS-AVA.** * denotes using the ground-truth bounding boxes of the test data. All methods employ the ViT-L/14 backbone. In the inference stage of Vita-CLIP [31], we use two different tracklets: (1) tracklets obtained by associating detected boxes with ByteTrack [36], and (2) tracklets with ground-truth boxes provided by AVA official.

In this section, we present the experimental results for unseen classes in each label split. We also calculate the harmonic mean (H) of the average performance of both base and unseen classes, to provide a more comprehensive assessment. In addition to using detected person boxes to measure the actual results on zero-shot spatial-temporal action detection, we also provide results using ground-truth bounding boxes of the test data. This allows us to analyze the performance without the influence of localization errors.

Table 1 shows our results on detecting unseen actions in AVA. First of all, the baseline method aligns image features during both the pretraining phase of CLIP and the inference phase of action detection, leading to good performance on unseen classes. It is worth noting that although iCLIP [10] and the baseline method achieve similar harmonic mean, iCLIP's performance on unseen classes is relatively poor. This suggests that their interaction modeling affects generalization ability. In contrast, our Person-Context Interaction approach leverages CLIP's pretrained knowledge, resulting in superior performance on both base and unseen classes.

We present the experimental results on ZS-JHMDB and ZS-UCF in Tabs. 2 and 3. Since our method focuses on detecting different actions for each person, individuals in the same video may be classified into different actions. This general setting can result in other video classification methods having an advantage over us in these two datasets. For a fair comparison with the other methods, we further adopt the assumption that a video contains only one action. In this context, we perform soft voting on each person's classification score, extending our method to suit this scenario.

We first present the results on ZS-JHMDB in Table 2. Firstly, the video-language model [30] with ROI align has difficulty accurately detecting unseen actions. We speculate that this is primarily due to the alignment of global video features (CLS token) during the pretraining process of ViCLIP. As a result, the person-specific features extracted from the video feature map could not align well with the text features. Without the assumption, our ST-CLIP achieves the highest average performance on unseen classes, along with the best harmonic mean. With the assumption applied, our method still outperforms other video classification approaches, achieving an average performance 3.82 mAP higher than [31]. Furthermore, by using ground-truth bounding boxes to eliminate localization errors, our performance improves further to 90.12 mAP.

Method	Frame mAP@0.5							
	split 1	split 2	split 3	split 4	avg	H	avg*	H*
Without the assumption of single-action video								
baseline	64.63	71.04	82.13	75.88	73.42	65.45	81.21	71.82
ViCLIP [30]	52.29	72.29	53.59	70.58	62.19	68.03	68.15	74.05
iCLIP [10]	66.53	69.99	82.88	71.84	72.81	74.47	79.01	81.09
ST-CLIP (Ours)	**74.55**	**74.97**	**83.59**	**83.22**	**79.08**	**77.65**	**85.53**	**84.32**
With the assumption of single-action video								
ActionCLIP [29]	69.18	75.28	77.11	76.55	74.53	75.82	82.07	83.25
A5 [13]	50.92	66.06	69.07	60.21	61.57	68.94	67.39	75.34
X-CLIP [19]	72.91	72.62	80.02	77.78	75.83	77.13	83.11	84.34
Vita-CLIP [31]	68.60	**82.35**	84.95	80.19	79.02	80.46	86.02	87.49
ST-CLIP (Ours)	**79.62**	78.70	**85.84**	**87.19**	**82.84**	**80.96**	**90.12**	**88.48**

Table 2. **Evaluation on ZS-JHMDB.** * denotes using the ground-truth boxes of the test data. All methods use the ViT-B/16 backbone.

Method	Frame mAP@0.5							
	split 1	split 2	split 3	split 4	avg	H	avg*	H*
Without the assumption of single-action video								
baseline	48.37	52.84	39.76	51.92	48.22	49.00	90.49	82.45
ViCLIP [30]	41.37	43.93	24.14	34.04	35.87	43.72	63.07	72.50
iCLIP [10]	**50.34**	52.75	39.73	47.95	47.69	**54.46**	85.47	90.35
ST-CLIP (Ours)	49.09	**54.95**	**42.05**	**53.92**	**50.00**	54.41	**91.80**	**91.34**
With the assumption of single-action video								
ActionCLIP [29]	**52.64**	55.01	38.04	51.84	49.38	55.36	89.72	92.91
A5 [13]	46.78	47.36	41.23	53.18	47.14	54.24	85.60	90.86
X-CLIP [19]	50.10	56.94	44.52	55.35	51.73	56.96	94.12	95.30
Vita-CLIP [31]	52.52	**57.22**	**45.12**	**57.32**	**53.05**	**57.65**	**96.57**	**95.94**
ST-CLIP (Ours)	51.36	56.30	43.12	54.52	51.33	55.94	95.02	94.75

Table 3. **Evaluation on ZS-UCF.** * denotes using the ground-truth boxes of the test data. All methods use the ViT-B/16 backbone.

Table 3 presents the results on ZS-UCF. It is worth mentioning that the localization errors have a noticeable impact on our performance in this dataset. Since there are instances where irrelevant people who are not performing actions are detected, these individuals should be considered as part of the background. However, these false positive cases will also contribute to the soft voting process, leading to our classification results being slightly inferior to other methods that solely rely on sampled frames to determine video

labels. In this case, our method still outperforms [13, 29], and exhibits similar performance to [19]. With ground-truth bounding boxes, our method achieves the second-best average performance. However, it is important to note that the UCF101-24 dataset has relatively low image quality, with most characters occupying only a small portion of the frame. This impacts the accuracy of our individual classification, putting us at a slight disadvantage compared to [31], which relies on the entire sampled frame.

5.4. Ablation Study

We present an ablation study in Tabs. 4 to 8 to investigate different design choices in our method. The experiments are performed on the label split 1 of ZS-JHMDB and ZS-AVA.

Proposed components: We first investigate the importance of each component in Table 4. On J-HMDB, our proposed Person-Context Interaction effectively models the relationship between individuals and their surroundings, resulting in a 2.35 mAP improvement compared to the baseline. Furthermore, the Context Prompting module leverages context information to enhance text content, leading to an additional improvement of 7.57 mAP. On AVA, the above two components also demonstrate their effectiveness. Additionally, in multi-action videos, our Interest Token Spotting can identify context tokens most relevant to individual actions for prompting, further enhancing performance.

Components	J-HMDB	AVA
Baseline	64.63	8.67
+ Person-Context Interaction	66.98	10.41
+ Context Prompting	**74.55**	12.07
+ Interest Token Spotting	-	**12.85**

Table 4. **Proposed components**

Comparison with iCLIP [10]: We demonstrate the advantages of our method compared to [10] in Table 5. Without prompting, our method performs slightly better than [10], and we do not need to use additional object detectors and interaction blocks in the interaction modeling process. In addition, our prompting strategy uses multiple levels of context tokens to augment text content, resulting in an improvement of 7.57 mAP. However, the prompting method of [10] relies heavily on the results of interaction modeling, which limits their performance improvement.

Context prompting: The results in Table 6 show the effectiveness of our prompting strategy. The results demonstrate that our prompting method, which utilizes tokens from low-level to high-level, yields better outcomes compared to using only high-level tokens from the last layer.

Method	+ prompt
iCLIP [10]	66.02 $\xrightarrow{+0.51}$ 66.53
Ours	**66.98** $\xrightarrow{+7.57}$ **74.55**

Table 5. **Comparison with iCLIP**

Prompting	mAP
w/o prompting	66.98
only in last layer	66.02
in every layer	**74.55**

Table 6. **Context prompting**

Person tokens: Table 7 explores different ways of generating person tokens. Initially, the simple approach of pooling over the embeddings of all patches in the person crop fails to deliver satisfactory performance. Furthermore, equipping adapters can perform better than using person features at the image encoder level. This demonstrates that our adapter can effectively adapt person tokens, making them more suitable for the subsequent person-context interaction.

Context tokens: Table 8 shows the importance of temporal modeling. Using only the current frame to obtain context tokens results in the worst performance, indicating that aggregating temporal information is beneficial when identifying continuous actions. Additionally, equipping only a 1-layer MHSA can significantly improve performance compared to simple average pooling.

Person tokens	mAP
Without the adapter	
Patch-embedding	70.92
Image encoder	70.29
With the adapter	
1-layer FC	67.37
2-layer FFN	**74.55**

Table 7. **Person tokens**

Temporal modeling	mAP
w/o temporal	68.03
average pooling	68.63
1-layer MHSA	**74.55**
2-layer MHSA	72.06

Table 8. **Context tokens**

6. Conclusion

In this paper, we explore zero-shot spatio-temporal action detection. We propose a complete benchmark on J-HMDB, UCF101-24, and AVA. Besides, we propose a method to adapt the visual-language model for this task. The Person-Context Interaction employs pretrained knowledge to model the relationship between people and their surroundings, and the Context Prompting module utilizes visual information to augment the text content. To address multi-action videos, we further introduce the Interest Token Spotting mechanism to identify the visual tokens most relevant to each individual action. The experiments demonstrate that our method achieves competitive performance compared to other video classification methods and can also handle multi-action videos.

Acknowledgement: This work was supported by NVIDIA Taiwan Research & Development Center (TRDC).

Supplementary Material

In the supplementary material, we present additional experimental results to substantiate the efficacy of our method and provide more details on the experimental settings. Initially, we elaborate our proposed benchmark for Zero-Shot Spatio-Temporal Action Detection in Sec. 1. Subsequently, we showcase a visualization depicting interest tokens on ZS-AVA in Sec. 2. Then, in Sec. 3, we illustrate the distribution of text features on both ZS-JHMDB and ZS-UCF to assess the impact of prompting. We then provide the complexity analysis of our method and others in Sec. 4. Besides, we provide more experimental analysis in Sec. 5 and describe how we handle multi-label prediction in Sec. 6. We also present the results using ground-truth bounding boxes on each benchmark in Sec. 7. Finally, we discuss some limitations of our approach in Sec. 8, and give the implementation details of other methods in Sec. 9.

1. Label Splits Details

In this section, we provide details of each label split used in our benchmarks. For ZS-JHMDB and ZS-UCF, we perform cross-validation on 4 label splits to assess the efficacy of our method. Each label split of ZS-JHMDB has 15 training classes and 6 testing classes, and each split of ZS-UCF has 18 training classes and 6 testing classes. More specifically, within each label split, there are 6 classes for testing, and all the remaining classes in the dataset are designated as training classes. In each label split, we follow the official split 1 of the two datasets, J-HMDB and UCF101-24, obtaining the training videos of the training classes and the testing videos of the test classes.

For ZS-AVA, to ensure an adequate volume of training data, we refrain from utilizing classes that frequently appear in most training videos as unseen classes. Our objective is to diversify each split by incorporating various types of unseen classes, including pose actions, object interactions, and person interactions. The split 1 contains 5 pose actions and 13 object interactions, split 2 contains 2 pose actions, 6 object interactions and 4 person interactions, and the split 3 contains 5 object interactions and 1 person interactions. The testing classes in each label split are shown in Table 9.

2. Interest Tokens

We first explore the effects of utilizing only interest tokens to prompt labels, as opposed to incorporating all context tokens. In Figure 4, we showcase the current frame along with a bounding box, indicating our objective of detecting the person's action. Additionally, we include neighboring frames to facilitate the observation of changes in the video over this period. Since the context tokens have not undergone spatial modeling in the first image encoder layer, we choose to visualize the results obtained from this layer to improve the interpretability of the identified interest tokens. In Figure 4a, the target person is engaged in the action of brushing teeth, while another person undertaking distinct actions is highlighted within the circle. In this scenario, our Interest Token Spotting mechanism can identify the context tokens most relevant to the action of brushing teeth. This ensures that the prompting process selectively incorporates these crucial visual clues, enhancing the focus on pertinent information. Likewise, in Figure 4b, the individual highlighted as the target person is engaged in fishing, while the person within the circled area is swimming. In this instance, the identified interest tokens exclude information associated with swimming, consequently enhancing the quality of the prompting process. Furthermore, in comparison to utilizing all context tokens for prompting, the use of only interest tokens can elevate the confidence score for the "brush teeth" action from 0.34 to 0.45, and the score of "fishing" can also be improved from 0.30 to 0.34. The results demonstrate that even when dealing with an unseen action not encountered during the training process, our method excels in identifying information most relevant to this action within a multi-person environment.

We provide more visualization results in Figure 5. In Figure 5a and 5b, when the action performed by the target person has limited relevance to others, the identified interest tokens will exhibit reduced emphasis on areas involving other people. Conversely, in Figure 5c and 5d, where the person's actions interact with others, our method adeptly recognizes context tokens within other people's areas as interest tokens, effectively extracting relevant information from those regions. Moreover, our Interest Token Spotting possess the capability to identify crucial objects, enhancing our ability to recognize actions, such as the chair and the computer in Figure 5a, and the cellphone in Figure 5d.

3. Text Features Distribution

As class names inherently carry limited semantic information, relying solely on the text features of these words to calculate similarity with person tokens may introduce ambiguity, potentially impacting the accuracy of action classification. To examine the distribution of text features in the feature space, we employ Principal Component Analysis (PCA) to reduce each feature to two dimensions. Subsequently, we illustrate the text features distribution of the CLIP text encoder and various Context Prompting layer outputs in Figure 6.

Initially, the results reveal that regardless of the label group, the original text features derived solely from class names are relatively close, potentially resulting in misclassification of actions. However, our Context Prompting module will utilize different visual information in each layer to augment the text content, thereby gradually increasing the discriminability between each class name. For example, in

ZS-JHMDB						
split 1	clap	sit	wave	throw	pullup	catch
split 2	kick ball	run	climb stairs	stand	shoot gun	pick
split 3	brush hair	push	pour	shoot bow	jump	shoot ball
split 4	golf	swing baseball	walk	clap	sit	wave
ZS-UCF						
split 1	Ice Dancing	Floor Gymnastics	Salsa Spin	Skate Boarding	Soccer Juggling	Volleyball Spiking
split 2	Basketball	Skiing	Biking	Golf Swing	Cliff Diving	Diving
split 3	Fencing	Horse Riding	Surfing	Long Jump	Pole Vault	Basketball Dunk
split 4	Rope Climbing	Skijet	Tennis Swing	Trampoline Jumping	Cricket Bowling	Walking With Dog
ZS-AVA						
split 1	brush teeth	chop	cook	crawl	dance	play board game
	extract	fishing	jump/ leap	kick (an object)	martial art	dig
	row boat	sail boat	shovel	stir	swim	take a photo
split 2	hand clap	lift (a person)	play board game	swim	shovel	take a photo
	kick (a person)	martial art	play with kids	row boat	play with pets	work on a computer
split 3	hug (a person)	press	shovel	stir	text on/look at a cellphone	turn (e.g., a screwdriver)

Table 9. **Testing classes in each label split on ZS-JHMDB, ZS-UCF and ZS-AVA.**

Figure 6c, if we rely solely on the original text features to categorize the action "Skateboarding," it may result in misclassification due to the ambiguity with "VolleyballSpiking" in the feature space. However, as these text features extract visual information across multiple prompting layers, the distinction between them becomes more pronounced, which aids in easier differentiation between various actions.

4. Complexity Analysis

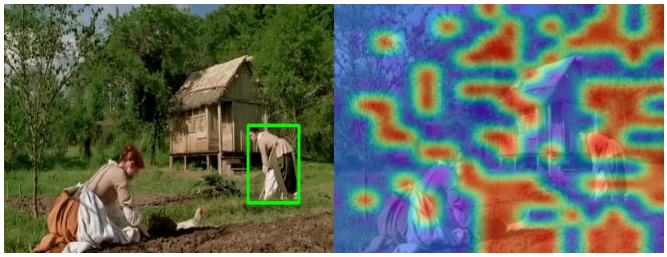
We report the complexity analysis on the label split 1 of ZS-JHMDB in Table 10. We calculate the GFLOPs required to infer a video and the training/inference time is reported on all the videos in the train/test split. Compared to [10] and our method, which classifies actions for individuals, these video classification methods only need to classify the entire video once to infer the actions of all detected people in it, potentially resulting in lower GFLOPs and inference time. As for training time, unlike [10] and our method, which

(a) **Unseen action: brush teeth.** Compared with utilizing all context tokens for prompting, the use of only interest tokens increases the confidence score from 0.34 to 0.45.

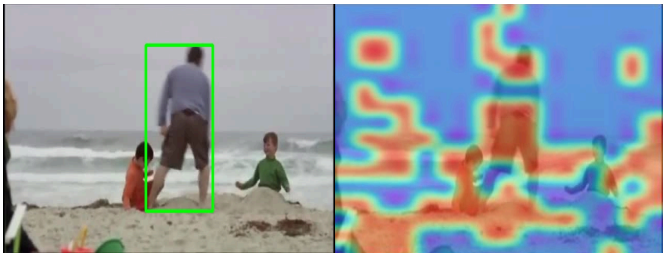
(b) **Unseen action: fishing.** Compared with utilizing all context tokens for prompting, the use of only interest tokens increases the confidence score from 0.30 to 0.34.

Figure 4. **The impact of interest tokens.**

(a) sit, **work on a computer**



(b) bend/bow (at the waist), carry/hold (an object), **shovel**



(c) bend/bow (at the waist), watch (a person), **lift (a person)**, **play with kids**

(d) walk, listen to (a person), **text on/look at a cellphone**

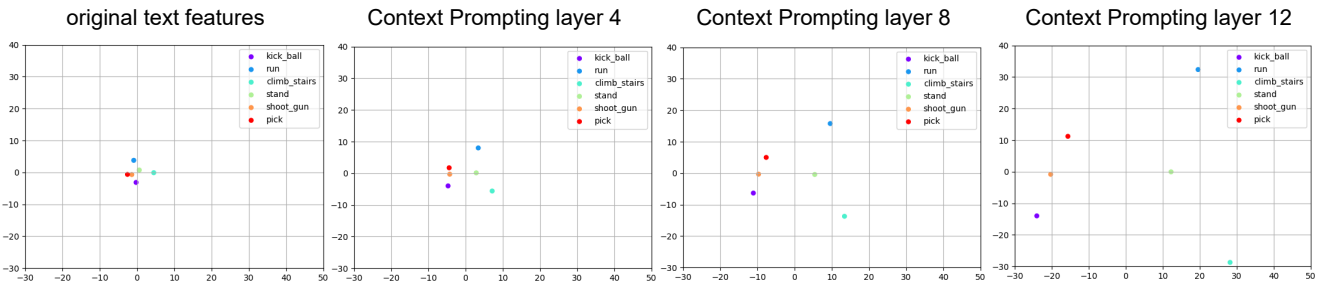
Figure 5. **More visualization of interest tokens.** Bold text indicate unseen actions.

optimize on an individual basis during the training phase, these methods optimize on an entire video unit, thus requiring fewer training iterations. However, it is worth mentioning that our method can achieve **79.62** frame mAP with soft voting, which is a substantial improvement over other methods. Besides, these video classification methods only work on single-action videos, while we can handle videos with multiple actions. Compared to [10], which also uses individuals as the classification unit, our method requires less training time and achieves 74.55 mAP without soft voting, surpassing [10] by 8.02 mAP.

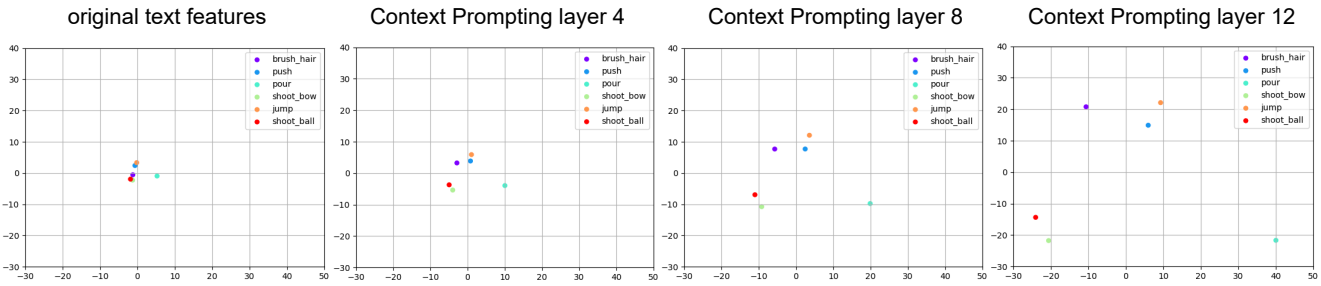
5. Additional Experimental Analysis

5.1. Interest Token Spotting on Single-Action Video

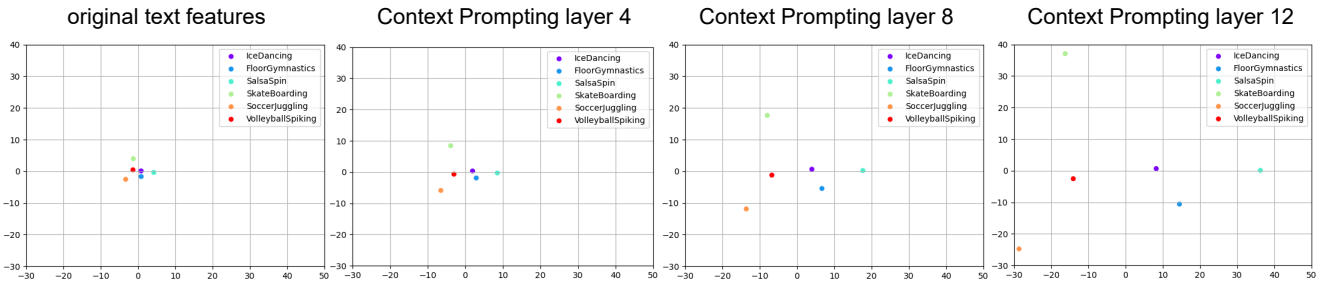
We further conduct experiments applying Interest Token Spotting on single-action video. We present the results on ZS-JHMDB label split 1 with detected boxes and ZS-UCF



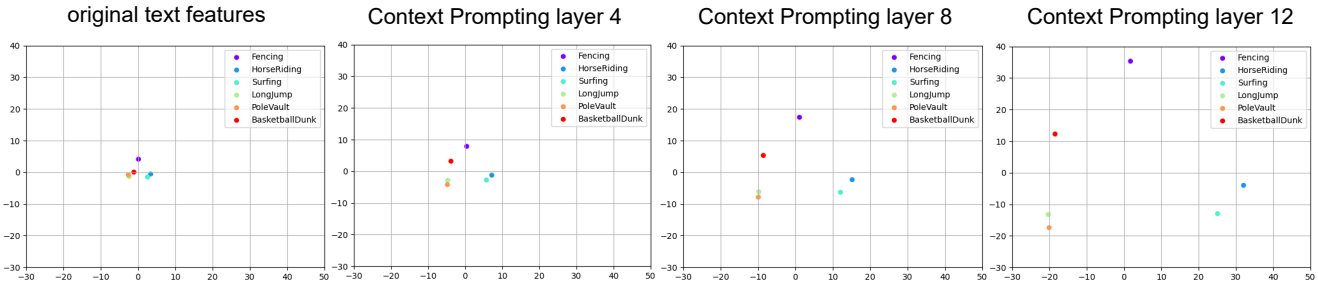
(a) **Text features distribution prompted by the "kick ball" visual tokens.**



(b) **Text features distribution prompted by the "push" visual tokens.**



(c) **Text features distribution prompted by the "SkateBoarding" visual tokens.**



(d) **Text features distribution prompted by the "HorseRiding" visual tokens.**

Figure 6. **Text Features Distribution.**

label split 1 with ground-truth boxes in Table 11. The results show that when a video contains only single action, all context tokens can be considered relevant to that action. Therefore, relying solely on the information from interest tokens during the prompting process is insufficient.

5.2. Complexity Analysis on Proposed Components

We present the complexity analysis on each of our proposed components in Table 12. The computational complexity (GFLOPs) is calculated on 1 AVA frame with 3 detected people, and we report the trainable parameters

Method	GFLOPs	Training time (s)	Inference time (s)	Throughput (FPS)	Trainable parameters (M)	mAP
baseline	721.11	-	3.95	671.14	-	64.63
ActionCLIP [29]	409.7	199.06	9.04	293.25	18.92	69.18
A5 [13]	264.15	239.34	10.33	256.63	6.35	50.92
X-CLIP [19]	159.91	185.36	6.61	401.06	57.93	72.91
Vita-CLIP [31]	145.26	259.52	5.88	450.85	35.61	68.60
iCLIP [10]	2410.07	2493.1	58.11	45.62	11.6	66.53
ST-CLIP (Ours)	2431.51	806.76	31.77	83.44	55.11	**79.62**

Table 10. **Complexity analysis on the label split 1 of ZS-JHMDB.** We train/test all the methods on 8 Tesla V100 GPUs.

Prompting Tokens	ZS-JHMDB	ZS-UCF
Interest Tokens	74.16	86.50
All Tokens	**74.55**	**87.11**

Table 11. **Interest Token Spotting on single-action video**

Device	Time (s)	FPS
1 GPU	190.94	4.64
8 GPUs	40.22	22.03

Table 13. **Latency on 1 AVA video**

with ViT-L backbone. It is worth noting that the majority of Person-Context Interaction's computational complexity arises from using CLIP's pretrained image encoder to extract person features. Additionally, when applying Interest Token Spotting, we must prompt each person's text features individually, rather than using a shared set of text features, which further increases the computational complexity.

Components	GFLOPs	Trainable params (M)
Person-Context Interaction	583.04	22.97
Context Prompting	21.67	189
Interest Token Spotting	40.88	0

Table 12. **Complexity Analysis on Proposed Components**

5.3. Latency on Multi-Action Video

We report the latency of our ST-CLIP for processing multi-action videos in Table 13. We inference on 1 AVA video which contains 886 frames. Our method takes about 3 minutes for inference when using a single GPU.

5.4. Additional Ablation Study

We present more ablation studies on the label split 1 of ZS-JHMDB and ZS-AVA.

LoRA in FFN: In Table 14, we investigate the impact of LoRA ranks. The results show that when we additionally train learnable matrices with rank 8, we can perform better than relying solely on CLIP's pretrained weight.

Interest tokens: In table 15, we conduct experiments using different numbers of interest tokens to observe their impact on the results. Our findings indicate that in multi-action videos, introducing too many tokens can potentially sample background noise unrelated to the action, thereby impacting the effectiveness of prompting.

Rank	mAP
w/o LoRA	72.80
r = 2	64.52
r = 4	71.70
r = 8	**74.55**
r = 16	68.39

Table 14. **LoRA in FFN**

Number	mAP
80	11.77
100	**12.85**
150	12.46
200	11.46

Table 15. **Interest tokens**

6. Multi-Label Prediction on AVA

In the AVA dataset, each person must perform a pose action, and they may also engage in two additional types of actions: object interaction and person interaction. To handle multi-label prediction, after calculating the cosine similarity using person-context relational tokens and text features, we apply softmax as the activation function for pose actions, and sigmoid for the other two types — object interaction and person interaction.

7. Results with Groundtruth Bounding Boxes

In this section, we present the results using ground-truth bounding boxes of the test data on ZS-AVA, ZS-JHMDB, and ZS-UCF in Tabs. 16 to 18 respectively, to analyze the performance of all methods without localization errors.

The results in Table 16 show that our performance is better than most other methods, except for splits 2 and 3, where it is slightly inferior to [31]. However, the performance of video classification methods (e.g., [31]) on multi-action videos largely depends on the quality of tracklets. In the case of using detected boxes to comply with the zero-shot scenario, these methods require additional trackers instead of using ground-truth tracklets, which significantly impacts their performance (avg drops from 10.45 to 3.06). In contrast, our ST-CLIP is not constrained by the tracker, so the performance will not be significantly reduced if we switch to using detected boxes.

Method	Frame mAP@0.5			
	split 1	split 2	split 3	avg
baseline	13.49	13.23	3.55	10.09
baseline (person crop)	11.88	7.06	3.72	7.55
iCLIP [10]	5.58	12.59	2.56	6.91
Vita-CLIP [31]	10.88	**14.69**	**5.78**	10.45
ST-CLIP (Ours)	**15.80**	14.32	5.16	**11.76**

Table 16. **Evaluation on ZS-AVA.** We report the results using the ground-truth bounding boxes of the test data.

Tabs. 17 and 18 present the results on ZS-JHMDB and ZS-UCF. On ZS-JHMDB, our method has the best average performance whether using detected boxes or ground-truth boxes. On ZS-UCF, when we use ground-truth boxes to avoid false positives affecting the soft voting results, our method can achieve the second-best performance on most splits and on average.

8. Limitations

In certain scenarios, there may be unrelated individuals who are not engaged in any actions. In such cases, these individuals should be treated as background elements. However, achieving this necessitates the human detector to learn from samples of a specific action class to effectively detect only the person executing this action. In a zero-shot setting, the person detector is prone to experiencing more localization errors as it has not been exposed to samples of the testing classes. Taking Figure 7 as an example, the Faster R-CNN may detect two person bounding boxes with high confidence scores, although only one of them is genuinely performing the action "SoccerJuggling". Consequently, the detection of the other box will be considered a false positive.

Given our method's two-stage pipeline nature, the aforementioned localization error will influence our performance in two aspects: (1) In processing single-action videos, these false positive instances will contribute to soft voting, thereby compromising the classification performance

Method	Frame mAP@0.5				
	split 1	split 2	split 3	split 4	avg
Without the assumption of single-action video					
baseline	70.17	79.91	**95.51**	79.26	81.21
ViCLIP [30]	54.30	81.32	63.44	73.55	68.15
iCLIP [10]	71.06	76.57	93.44	74.98	79.01
ST-CLIP (Ours)	**79.02**	**82.99**	94.83	**85.28**	**85.53**
With the assumption of single-action video					
ActionCLIP [29]	74.92	83.33	89.46	80.58	82.07
A5 [13]	54.81	74.36	78.20	62.19	67.39
X-CLIP [19]	77.85	80.21	92.71	81.68	83.11
Vita-CLIP [31]	73.13	**89.83**	97.05	84.07	86.02
ST-CLIP (Ours)	**85.02**	87.87	**97.33**	**90.27**	**90.12**

Table 17. **Evaluation on ZS-JHMDB.** We report the results using the ground-truth bounding boxes of the test data.

Method	Frame mAP@0.5				
	split 1	split 2	split 3	split 4	avg
Without the assumption of single-action video					
baseline	85.70	95.11	88.17	**92.99**	90.49
ViCLIP [30]	75.21	74.79	44.07	58.22	63.07
iCLIP [10]	**91.30**	89.91	81.56	79.12	85.47
ST-CLIP (Ours)	87.11	**96.73**	**91.11**	92.23	**91.80**
With the assumption of single-action video					
ActionCLIP [29]	91.33	94.39	85.26	87.90	89.72
A5 [13]	84.74	79.67	90.00	87.98	85.60
X-CLIP [19]	89.85	98.28	93.69	94.64	94.12
Vita-CLIP [31]	**94.31**	**98.80**	**96.56**	**96.59**	**96.57**
ST-CLIP (Ours)	92.90	98.75	93.97	94.44	95.02

Table 18. **Evaluation on ZS-UCF.** We report the results using the ground-truth bounding boxes of the test data.

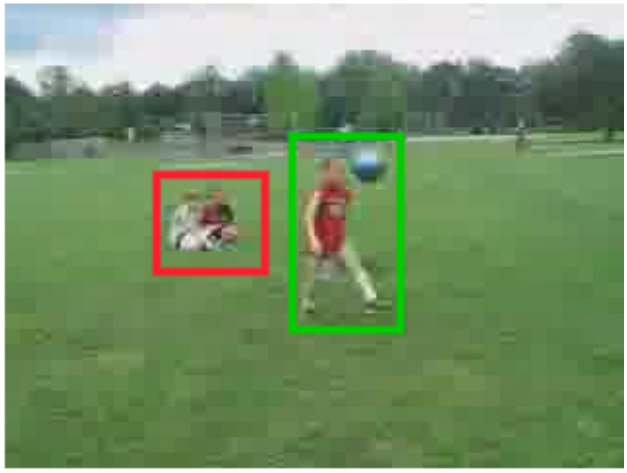


Figure 7. **Visualization of wrong detected person bounding box.** The image is from the test data of class "SoccerJuggling", and the boxes are detected by Faster R-CNN. The red box is counted as a false positive according to the labeling in this dataset since the people inside are not doing the "SoccerJuggling" action.

to some extent. (2) When handling multi-action videos, the person tokens from these incorrectly detected boxes will be employed for interaction modeling, which will influence the effectiveness of the person-person interaction involving the target individual.

9. Implementation Details

In this section, we provide the implementation details for each compared method in our experiments. For iCLIP [10], we employ the best-performing model featuring four types of interaction modules and Interaction-Aware Prompting. For ActionCLIP [29], to follow their approach, we sample 8 frames from each video, then utilize 6 transformer encoder layers to perform temporal modeling. Besides, we adopt the handcrafted prompts they proposed to prompt labels. For Efficient-Prompting [13], we sample 16 frames from each video, and employ the A5 model they proposed, which uses 2 encoder layers for temporal modeling and prepends/appends 16 vectors with the textual embeddings. For X-CLIP [19], we sample 8 frames from each video, and use the proposed Cross-frame Communication and 1-layer Multi-frame Integration Transformer to generate video features. Besides, we also leverage the Video-specific Prompting in their method to prompt labels. For Vita-CLIP [31], we sample 8 frames from each video, and utilize 8 Global Video-Level Prompts following their method. For ViCLIP [30], we sample 8 frames from each video and use the video encoder to obtain the video feature map. Then, based on each person's bounding box, we apply ROIAlign to extract individual person features for classification. As for the training iterations, for iCLIP, we train the network for 7K iterations on ZS-JHMDB and 10K iterations on ZS-UCF as in their method. We train other video classification methods for 1K iterations on ZS-JHMDB and ZS-UCF, and more iterations will lead to lower performance due to overfitting. For all the aforementioned methods except ViCLIP [30], we utilize the CLIP backbone and freeze the image and text encoder, consistent with our approach. As for ViCLIP [30], we freeze the text encoder and finetune the video encoder during training. The video encoder is pretrained on the InternVid-10M-FLT [30] dataset. We provide the details of training hyperparameters for all methods on ZS-AVA, ZS-JHMDB, and ZS-UCF in Tabs. 19 to 21.

Method	Iterations	Learning rate	Warmup iterations	Warmup factor	Optimizer	Batch size
iCLIP [10]	30000	0.0004	2000	0.25	SGD	8
Vita-CLIP [31]	10000		1000			
ST-CLIP (Ours)	20000		2000			

Table 19. **hyperparameters on ZS-AVA.** All methods employ ViT-L/14 backbone.

Method	Iterations	Learning rate	Warmup iterations	Warmup factor	Optimizer	Batch size
ActionCLIP [29] A5 [13] X-CLIP [19] Vita-CLIP [31] ViCLIP [30]	1000	0.0002	700	0.25	SGD	8
iCLIP [10]	7000	0.0002	700	0.25	SGD	8
ST-CLIP (Ours)	3000	0.00025	800			

Table 20. **hyperparameters on ZS-JHMDB.** All methods employ ViT-B/16 backbone.

Method	Iterations	Learning rate	Warmup iterations	Warmup factor	Optimizer	Batch size
ActionCLIP [29] A5 [13] X-CLIP [19] Vita-CLIP [31] ViCLIP [30]	1000	0.0002	1000	0.25	SGD	8
iCLIP [10]	10000	0.0002	1000	0.25	SGD	8
ST-CLIP (Ours)	3000	0.00025	800			

Table 21. **hyperparameters on ZS-UCF.** All methods employ ViT-B/16 backbone.

References

[1] Nakul Agarwal, Yi-Ting Chen, Behzad Dariush, and Ming-Hsuan Yang. Unsupervised domain adaptation for spatio-temporal action localization. *arXiv preprint arXiv:2010.09211*, 2020. 2

[2] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8178–8187, 2021. 2

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[4] Han Fang, Pengfei Xiong, Luhui Xu, and Yu Chen. Clip2video: Mastering video-text retrieval via image clip. *arXiv preprint arXiv:2106.11097*, 2021. 3

[5] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai. Holistic interaction transformer network for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3340–3350, 2023. 1, 2

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 1, 2

[7] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019. 2

[8] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6047–6056, 2018. 6

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 2

[10] Wei-Jhe Huang, Jheng-Hsien Yeh, Min-Hung Chen, Gueter Josmy Faure, and Shang-Hong Lai. Interaction-aware prompting for zero-shot spatio-temporal action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 284–293, 2023. 2, 3, 5, 6, 7, 8, 10, 11, 13, 14, 15, 16

[11] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013. 5

[12] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR, 2021. 2, 3

[13] Chen Ju, Tengda Han, Kunhao Zheng, Ya Zhang, and Weidi Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. 2, 3, 6, 7, 8, 13, 14, 15, 16

[14] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 6

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 6

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[17] Ron Mokady, Amir Hertz, and Amit H Bermano. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*, 2021. 3

[18] Sauradip Nag, Xiatian Zhu, Yi-Zhe Song, and Tao Xiang. Zero-shot temporal action detection via vision-language prompting. In *European Conference on Computer Vision*, pages 681–697. Springer, 2022. 2

[19] Bolin Ni, Houwen Peng, Minghao Chen, Songyang Zhang, Gaofeng Meng, Jianlong Fu, Shiming Xiang, and Haibin Ling. Expanding language-image pretrained models for general video recognition. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022. 2, 3, 6, 7, 8, 13, 14, 15, 16

[20] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021. 1, 2

[21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2, 3

[22] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[23] Khurram Soomro and Mubarak Shah. Unsupervised action discovery and localization in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 696–705, 2017. 2

[24] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[25] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018. 2

[26] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 71–87. Springer, 2020. 1, 2

[27] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 1

[28] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 1

[29] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 2, 3, 6, 7, 8, 13, 14, 15, 16

[30] Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, et al. Internvid: A large-scale video-text dataset for multimodal understanding and generation. *arXiv preprint arXiv:2307.06942*, 2023. 6, 7, 14, 15, 16

[31] Syed Talal Wasim, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Mubarak Shah. Vita-clip: Video and text adaptive clip via multimodal prompting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23034–23044, 2023. 2, 3, 6, 7, 8, 13, 14, 15, 16

[32] Wenhao Wu, Xiaohan Wang, Haipeng Luo, Jingdong Wang, Yi Yang, and Wanli Ouyang. Bidirectional cross-modal knowledge exploration for video recognition with pre-trained vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6620–6630, 2023. 2, 3

[33] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019. 2

[34] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6978–6988, 2023. 3

[35] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, et al. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 2, 3

[36] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *European conference on computer vision*, pages 1–21. Springer, 2022. 7