Concept-Guided Open-Vocabulary Temporal Action Detection

Songmiao Wang
Tianjin University
songmiaow@tju.edu.cn

Ruize Han
Shenzhen Institute of Advanced Technology,
Chinese Academy of Sciences
rz.han@siat.ac.cn

Wei Feng
Tianjin University
wfeng@tju.edu.cn

Abstract

Vision-Language (ViL) models have shown strong open-vocabulary learning abilities in various video understanding tasks. However, when applied to Open-Vocabulary Temporal Action Detection (OV-TAD), existing OV-TAD methods often face challenges in generalizing to unseen action categories due to their reliance on visual features, resulting in limited generalization. In this paper, we propose a novel Concept-guided Semantic Projection framework to enhance the generalization ability of OV-TAD models. By projecting video features into a unified action concept space, guiding the model to leverage the abstracted action concepts present in the video for action detection, rather than solely relying on visual details. To further improve feature consistency across action categories, we introduce a Mutual Contrastive Loss, ensuring semantic coherence and better feature discrimination. Extensive experiments on ActivityNet and THUMOS14 benchmarks demonstrate that our method outperforms state-of-the-art OV-TAD models. Code and data are available at Concept-Guided-OV-TAD.

***Keywords:** Open-Vocabulary, Temporal Action Localization, Visual-language Models*

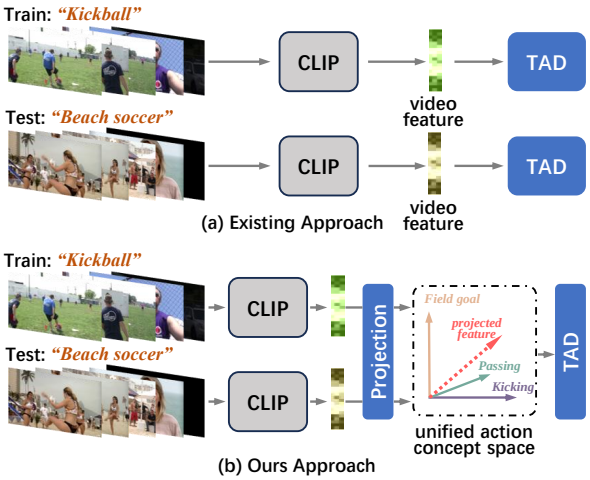


Figure 1. Illustration of (a) existing OV-TAD methods that rely on visual features, struggling to generalize to unseen actions due to the visual discrepancy between training and testing sets. (b) Our approach addresses this by projecting video features into a unified action concept space, allowing better generalization through semantic guidance.

1. Introduction

Temporal Action Detection (TAD) addresses the task of identifying and classifying actions within untrimmed videos [1, 2, 3]. In recent years, significant progress has been made in TAD using deep learning techniques for video understanding and analysis [4, 5, 6, 7]. However, a critical bottleneck in traditional TAD methods is their dependence on large-scale labeled datasets, which limits their scalability to new (unseen) action categories. To tackle this issue, Open-Vocabulary Temporal Action Detection (OV-TAD), also called as Zero-Shot Temporal Action Detection (ZS-TAD) [8], has emerged as a more promising problem. OV-TAD aims to detect actions from categories that have and have not been observed during training [9], thereby reducing the need for extensive labeled data for various action categories. OV-TAD typically relies on transferring knowledge from seen to unseen action categories. An effective way to achieve this is by leveraging Vision-Language (ViL) models, such as CLIP [10], due to their strong generalization capabilities, making them a promising tool for OV-TAD [11, 8, 12].

Nevertheless, applying ViL models to dense prediction tasks like OV-TAD presents unique challenges. A main issue lies in the over-reliance on visual features to localize potential target action segments. Specifically, OV-TAD mainly contains two sub-tasks, localizing the interested action segments from the untrimmed videos, and recognizing the action category of the video segment. Existing approaches [11, 8, 12] mainly aim to address the action classification task (especially for the unseen new actions), but not the previous localization task. Note that, the action localization is also significantly important for the OV-TAD, in which abundant training on the seen actions may make

the model overfitted on these actions, and difficult to identify the unseen (open-vocabulary) action segments from the background video (sequences without action). As shown in Figure 1(a), prior methods, no matter using a one-stage design or decoupling method for localization and classification, *primarily rely on the visual cues* to localize the potential action segments. However, this reliance on visual features can be problematic in OV-TAD, since the visual appearance of unseen action categories during testing may differ significantly from that of the seen categories during training. This visual mismatch restricts the model's generalization ability to localize the unseen actions, since the model may struggle to identify the actions that do not resemble those (seen actions) in the training set.

To overcome this limitation, as shown in Figure 1(b), we propose a novel concept-guided semantic projection (CSP) framework, which projects video features into a unified action concept space. Instead of relying solely on visual details, this approach focuses on *capturing the underlying semantics of actions, making the model generalize to identify (localize) the unseen action categories based on the action semantics rather than the specific visual features*. Through aligning the video features with action semantics, our approach addresses the fundamental challenge in OV-TAD which is the adaptive representation ability from the seen to unseen categories for action localization.

Moreover, to project the video feature into the concept space, a new challenge arises. How to ensure the projected features are semantically meaningful and discriminative. In other words, after projecting the features into a shared space, these features should accurately reflect the underlying action concepts and be discriminative among different action categories. For this purpose, we develop a novel mutual contrastive loss (MCL) that encourages the projected concept features (from CSP) to maintain the same semantic structure as the (language based) action categories. This loss aims to constrain the projected features not only to be expressive in action semantics but also discriminative, making the action concept space more representative and effective for OV-TAD.

We evaluate our approach on two widely-used video benchmarks, ActivityNet [13] and THUMOS14 [14], demonstrating that our method significantly outperforms current state-of-the-art OV-TAD/ZS-TAD models. Experimental results highlight the importance of addressing the visual representation deficiency for action localization and the effectiveness of semantically representative and discriminative features for OV-TAD task. The main contributions of this work are outlined as follows:

1) We propose a novel Concept-guided Semantic Projection (CSP) framework for OV-TAD, which projects video features into an action concept space, making the model effectively generalize to identify and localize the unseen action categories by focusing on underlying action semantics rather than specific visual cues.
2) We develop a Mutual Contrastive Loss (MCL) that encourages the concept features to mirror the semantic structure of language-based action categories, prompting the projected features to maintain semantic representation ability and be well-discriminative within the action concept space.
3) Extensive experimental results on ActivityNet and THUMOS14 demonstrate the superior performance of our approach, significantly outperforming state-of-the-art methods in OV-TAD.

2. Related Works

2.1. Temporal Action Detection

Close-set temporal action detection methods can broadly be categorized into two categories: two-stage and single-stage methods.

Two-stage methods. Two-stage methods first generate action proposals, then classify them, and further refine the boundaries of the action candidates. In the proposal generation stage, some methods rely on classifying anchor windows [4, 15, 16], while others generate proposals by accurately locating action boundaries [17, 18, 19]. Additionally, some works focus on modeling intra- and inter-proposal relationships using graph based networks [20, 21, 22]. For instance, Li *et al*. [22] proposed an intra-attention-based GCN and an inter-attention-based GCN, which are further fused to simultaneously model long-range dependencies and inter-proposal relationships. In subsequent studies [23, 24, 25], the self-attention mechanism has also been applied to model these relationships.

Single-stage methods. Different from Two-stage methods, single-stage methods do not rely on proposals and instead localize actions in a single shot. Some single-stage methods [26, 27, 28, 29, 30, 31] predict both the temporal boundaries and action categories of each action instance simultaneously [32]. Lin *et al*. [26] proposed SSAD network based on 1D temporal convolutional layers to jointly predict action proposals and refine the temporal boundaries. Similarly, Gao *et al*. [27] also skip the proposal generation step by temporal coordinate regression. While these methods are effective at identifying action instances, their performance is inherently restricted by the reliance on predefined anchors [32]. To address this issue, Lin *et al*. [33] proposed an anchor-free framework that includes a simple anchor-free predictor for generating coarse temporal boundaries. Recently, Zhang *et al*. [34] proposed ActionFormer, a transformer-based model that detects actions and classifies them in a single step, without the need for action proposals or pre-defined anchors. In [35], a long-memory transformer was introduced by Cheng *et al*. to enhance long-range tem-

poral boundary localization.

2.2. Open-Vocabulary Temporal Action Detection

Open-vocabulary learning aims to recognize new classes that have not been seen during the training phase [36]. Traditional open-vocabulary learning methods utilize manually defined visual attributes, such as shape and color, to enable the model to recognize unseen classes [12]. Subsequent research [37, 38, 39, 40, 41] has explored replacing manually defined visual attributes with word vectors derived from models such as Word2Vec [42] and GloVe [43], which enhances the scalability of the models. Zhang *et al*. first introduced open-vocabulary learning to Temporal Action Detection [9] and defined the Open-Vocabulary Temporal Action Detection (OV-TAD) task, which aims to detect unseen actions in untrimmed videos. They proposed an end-to-end network with a Detection Subnet that classifies activity proposals using cosine distance between proposal features and Word2Vec [42] label features. More recently, several works have attempted to leverage the generalization capabilities of CLIP [10] to enhance open-vocabulary temporal action detection. Ju *et al*. [11] firstly utilizes pre-trained CLIP as proposal classifier. Owing to its two-stage design, it struggles to mitigate the interference between the localization and classification processes. To resolve this issue, Nag *et al*. [12] introduced a one-stage OV-TAD architecture, which removes the dependence between these two processes. Differently, Li *et al*. [8] decoupled the generation of action proposals and action classification and trained separate networks to avoid interference between the two tasks. Differing from existing methods, this paper introduces a concept-guided semantic projection framework that projects visual features into an action concept space, allowing for more effective generalization to unseen actions.

3. Methodology

3.1. Overview

The overall structure of our method is shown in Figure 2. First, given an untrimmed video V, following previous practices [17, 6, 12], we uniformly sample T temporal points from the video, denoted as $X = \{x_0, \ldots, x_T\}$. Then, we extract frame-wise features using image encoder Φ_{img} from CLIP. Since the image encoder of CLIP lacks temporal modeling capabilities, we leverage a transformer encoder to capture temporal dependencies, resulting in the video feature $\mathbf{F}_{\text{vid}}$ as

$$\mathbf{F}_{\text{vid}} = \mathcal{T}(\Phi_{\text{img}}(X)) \in \mathbb{R}^{T \times D}, \tag{1}$$

where $\mathcal{T}$ is the temporal transformer encoder and D is the feature dimension. To further enhance the open-vocabulary capability of the model, we propose a Concept-guided Semantic Projection (CSP) framework. This framework projects the raw video feature $\mathbf{F}_{\text{vid}}$ into a predefined action concept space, resulting in concept feature $\mathbf{F}_{\text{con}}$, and guides the model to leverage the semantic concepts of actions present in the video for action detection. This will be detailed in Section 3.2. To better train the CSP, we propose a Mutual Contrastive Loss (MCL), which leverages CLIP's text encoder Φ_{text} to compute action label features for each segment. This loss encourages the concept features to align with the semantic relationships between action categories, ensuring that the projection into the concept space remains both semantically meaningful and distinctive. Further details are provided in Section 3.3. In Section 3.4, we utilize the action localization and classification heads to accomplish the OV-TAD task.

3.2. Concept-guided Semantic Projection

In the OV-TAD setting, the disjoint action categories between the training and test sets pose a significant challenge for generalization. Existing methods rely heavily on pre-trained visual encoders to extract visual features for action localization and classification. However, this reliance on specific visual features can lead to overfitting, as the visual appearance of actions in the test set may differ significantly from those in the training set.

To address this limitation, we propose a concept-guided semantic projection (CSP) strategy. The core idea of CSP is to map the extracted visual features into a unified action concept space, enabling the model to focus on the semantic meanings of actions rather than low-level visual patterns. Specifically, this involves two key components: (1) constructing a semantically rich action concept space, independent of any specific dataset, and (2) designing a learnable concept interaction projection mechanism to align visual features with this concept space. By leveraging the action concept space, which is built using language-based representations, CSP captures the underlying semantics of actions and alleviates overfitting on specific visual features. This approach significantly enhances the model's ability to generalize to unseen action categories, addressing the core challenge of OV-TAD.

Action concept space. We aim to establish a unified, dataset-agnostic action concept space. The action concept space serves as the basis for aligning visual features with high-level action semantics, helping the model to focus on semantic meanings rather than low-level visual patterns.

Action concept generation. First, we use GPT-4 [44] to generate conceptually diverse action concepts. Specifically, we ask GPT-4 to categorize human activities into X broad categories, such as physical movement & exercise, work & productivity, etc. For each category, it generates Z action concepts, ensuring that these concepts are semantically diverse. For example, for the category Physical movement & exercise, concepts such as Ballet dancing, Mountain bik-

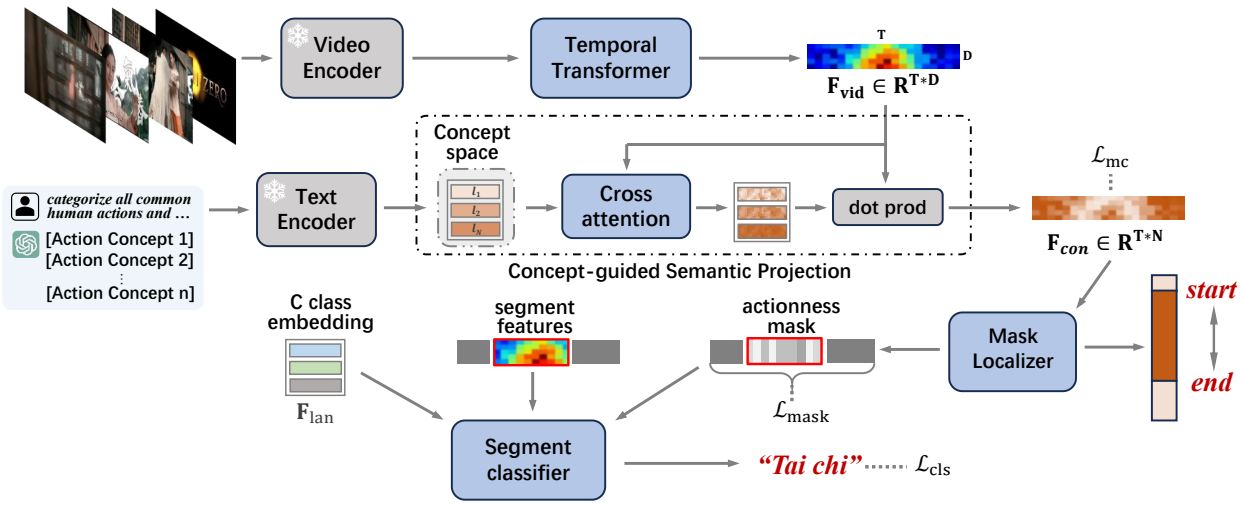


Figure 2. Overview of our proposed Open-Vocabulary Temporal Action Detection (OV-TAD) model. We first extract frame-level features from a pre-trained, frozen video encoder and capture temporal dependencies using a temporal transformer. The Concept-guided Semantic Projection (CSP) module then projects the video features into a unified action concept space. A Mutual Contrastive Loss (MCL) ensures that the projected features align with the semantic structure of action categories. Finally, actionness mask localizer and action category classifier are used for temporal action detection.

ing, and Rock climbing are generated. In total, we generate $X \times Z = N$ concepts.

Action concept encoding. After obtaining all the action concepts, we use CLIP's text encoder to generate text features θ_i for each concept. After normalization, these features are used as the basis concept vectors for the action concept space. With its powerful text encoding capability, the text encoder can capture subtle differences between categories, producing semantically rich concept vectors:

$$l_i = \theta_i / \|\theta_i\|_2^2 ; i = 1, 2, ..., N, \tag{2}$$

$$\mathbf{L} = [l_1, l_2, \dots l_N]^\top ; \mathbf{L} \in \mathbb{R}^{N \times D}, \tag{3}$$

where l_i is the concept vector, N is the number of concept vectors (which is also the number of action concepts), and D is the feature dimension. Note that these basis concept vectors are not necessarily orthogonal.

Alternative strategies. Besides prompting an LLM to generate action concepts, we also explored other strategies, such as incorporating action labels from existing action recognition datasets. Detailed experiments are provided in Section 4.4.

Concept interaction projection mechanism. To project the visual features $\mathbf{F}_{\text{vid}}$ into the action concept space, a straightforward approach is to calculate the dot-product similarity between $\mathbf{F}_{\text{vid}}$ and each basis concept vector of the concept space [44]. This yields a similarity score distribution across all concept vectors, serving as the projection of $\mathbf{F}_{\text{vid}}$ in the action concept space. However, when calculating the dot-product, much of the useful information in the original visual features can be lost. Additionally, using a fixed dot-product projection makes the process non-optimizable and non-learnable. Related experiments are detailed in Section 4.4. Therefore, we propose a learnable Concept Interaction Projection mechanism, which leverages cross-attention to facilitate cross-modal information interaction, enabling a more flexible and adaptive projection.

We first employ a multi-head cross-attention mechanism [45] to perform interaction between video features and concept vectors. Specifically, we use concept vectors $\mathbf{L}$ as the query $\mathbf{Q}$, and video features $\mathbf{F}_{\text{vid}}$ as the key $\mathbf{K}$ and value $\mathbf{V}$. First, the video and concept features are projected into the query, key, and value spaces through linear transformations

$$\mathbf{Q} = \mathbf{L}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{F}_{\text{vid}}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{F}_{\text{vid}}\mathbf{W}_V, \tag{4}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V$ are learnable projection matrices. For the multi-head attention mechanism, the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ are split into multiple heads, resulting in $\mathbf{Q}_h$, $\mathbf{K}_h$, and $\mathbf{V}_h$ for each head. Next, for each head h, we compute the scaled dot-product attention between the query $\mathbf{Q}_h$ and key $\mathbf{K}_h$ to obtain the attention weights

$$\mathbf{A}_h = \text{softmax}\left(\frac{\mathbf{Q}_h \mathbf{K}_h^\top}{\sqrt{d_h}}\right), \tag{5}$$

where d_h is the dimensionality of each attention head. Using these attention weights $\mathbf{A}_h$, we perform a weighted sum over the value $\mathbf{V}_h$ to obtain the updated concept vectors for each head

$$\mathbf{V}'_h = \mathbf{A}_h \mathbf{V}_h. \tag{6}$$

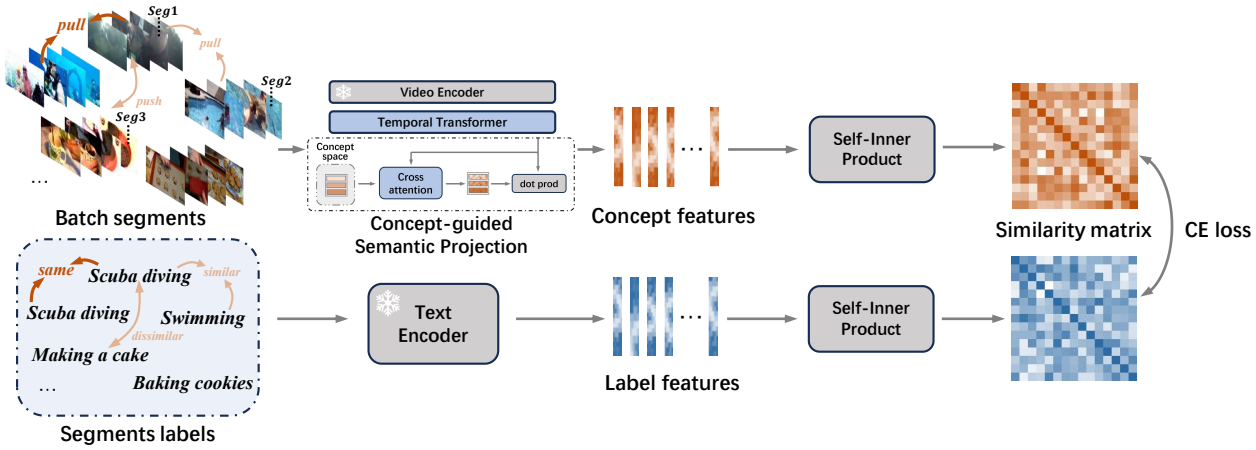


Figure 3. Illustration of the proposed mutual contrastive loss (MCL). After obtaining the concept features for each segment in the batch through Concept-guided Semantic Projection, we use CLIP's text encoder to extract the label features corresponding to each segment. We then compute the similarity matrices between the concept features and the label features. By constraining the distance between these two matrices, we encourage the concept features to reflect the semantic structure of the language-based action categories, ensuring that the projected features retain strong semantic representation and are highly discriminative within the action concept space.

Once we have the updated concept vectors $\mathbf{V}'_h$ for each head, we combine them to form the final updated concept vectors $\mathbf{V}'$. We then project the video features onto these updated concept vectors as

$$\mathbf{F}_{\text{con}} = \mathbf{F}_{\text{vid}}\mathbf{V}'^{\top} \in \mathbb{R}^{T\times N}, \tag{7}$$

where $\mathbf{F}_{\text{con}}$ represents the concept feature. Through the Concept Interaction Projection mechanism, $\mathbf{F}_{\text{con}}$ contains rich action semantic information.

3.3. Mutual Contrastive Loss for Video-Text Alignment

The Concept-guided Semantic Projection model projects the video features $\mathbf{F}_{\text{vid}}$ into the action concept space, resulting in the concept features $\mathbf{F}_{\text{con}}$, which capture the semantic information of the action concepts present in the video. To ensure the effectiveness of this projection, we further propose a Mutual Contrastive Loss (MCL) to specifically optimize the projection process.

A simple approach to achieve this would be to use a classic contrastive loss, such as InfoNCE [46], which pulls the features of segments belonging to the same action category closer and pushes those of different categories apart. However, this approach *overlooks the relative semantic relationships between categories*. For example, as shown in Figure 3, consider three video segments, $Seg1$, $Seg2$, and $Seg3$, corresponding to the action categories 'Scuba diving,' 'Swimming,' and 'Making a cake.' While all three actions are different, the semantic relationship between 'Scuba diving' and 'Swimming' is relatively similar, whereas the relationship between 'Scuba diving' and 'Making a cake' is more distant. To address this, MCL embeds the mutual relationships between categories into the concept space, ensuring that semantically related actions remain closer while unrelated actions are pushed further apart.

Specifically, during the training phase, we utilize the ground truth (GT) action annotations to identify the frames corresponding to specific action segments. For each action segment i, we aggregate the frame-level concept features $\mathbf{F}_{\text{con}}$ using mean pooling to obtain its segment-level feature:

$$s_i = \text{meanpool}(\mathbf{F}_{\text{con}}). \tag{8}$$

Here, s_i denotes the concept feature of the i-th segment, where $i \in [1, K]$ and K is the total number of segments in the batch. The corresponding action label for the i-th segment is denoted as t_i. Using CLIP's text encoder, we compute the text feature of each segment's action label as

$$e_i = \Phi_{\text{text}}(t_i), \tag{9}$$

where Φ_{text} represents CLIP's text encoder.

For each segment i and j in the batch, we compute the similarity between their concept features s_i and s_j using the inner product

$$q_{ij} = \langle s_i, s_j \rangle = s_i^{\top} s_j. \tag{10}$$

The similarity matrix calculated from the segments' concept features is denoted as $\mathbf{Q}$, where $\mathbf{Q} \in \mathbb{R}^{K\times K}$. Similarly, for each segment i and j, we compute the similarity between their label features e_i and e_j using the inner product

$$b_{ij} = \langle e_i, e_j \rangle = e_i^{\top} e_j. \tag{11}$$

The similarity matrix calculated from the segments' label features is denoted as $\mathbf{B} \in \mathbb{R}^{K\times K}$.

Finally, we compute the cross-entropy loss between matrix $\mathbf{Q}$ and matrix $\mathbf{B}$ as mutual contrastive loss:

$$\mathcal{L}_{\text{mc}} = -\frac{1}{N^2-N}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\Big(\mathbf{B}_{ij}\log\mathbf{Q}_{ij} + (1-\mathbf{B}_{ij})\log(1-\mathbf{Q}_{ij})\Big), \tag{12}$$

where we exclude the elements on the main diagonal of the matrices since the diagonal elements represent the relationship of segments with themselves, which is not meaningful.

The mutual contrastive loss between the similarity matrices $\mathbf{Q}$ and $\mathbf{B}$ ensures that the learned concept features not only differentiate between different action categories but also respect the relative semantic relationships between them. By minimizing the cross-entropy loss, the concept features are encouraged to mirror the semantic structure of the action categories, making the projection into the concept space both semantically meaningful and discriminative.

3.4. Action Localization and Classification Heads

Through the Concept-guided Semantic Projection model, we project video features $\mathbf{F}_{\text{vid}}$ into a unified action concept space, obtaining the concept features $\mathbf{F}_{\text{con}}$. Based on these features, we further perform action segment localization and classification using the actionness mask localizer and action category classifier task heads.

Actionness mask localizer. Similar to [12], we use a temporal 1D convolution to predict the actionness mask $\mathcal{M}$ as

$$\mathcal{M} = \text{sigmoid}\left(\Phi_{\text{conv}}\left(\mathbf{F}_{\text{con}}\right)\right). \tag{13}$$

The actionness mask $\mathcal{M}$ represents the probability of an action occurring at each temporal point, where $\mathcal{M} \in \mathbb{R}^{1\times T}$. By leveraging the rich action semantic information in the concept features $\mathbf{F}_{\text{con}}$, the prediction of the actionness mask no longer relies on specific visual elements, allowing for better generalization to unseen classes. We train the prediction of the actionness mask using cross-entropy loss as

$$\mathcal{L}_{\text{mask}} = \text{CrossEntropy}(\mathcal{M}, \mathcal{G}), \tag{14}$$

where $\mathcal{G}$ is the ground truth actionness mask, with the same size of $1 \times T$, where each temporal point is either 0 or 1. Similar to [12], based on the actionness mask, we apply a set of thresholds for thresholding, followed by further post-processing techniques, including Soft-NMS [47], to obtain the final localization results.

Action category classifier. For action segment classification, we adopt a process similar to the standard CLIP. As described in Eq. (9), we obtain the text features of the C target action categories, denoted as $\mathbf{F}_{\text{lan}} \in \mathbb{R}^{C\times D}$, using the text encoder. Given the visual features extracted from CLIP's image encoder, we obtain the foreground action segment feature by extracting the part where the action occurs, denoted as $\mathbf{F}_{\text{actn}} \in \mathbb{R}^{T'\times D}$, where T' is the temporal length of the foreground segment. Further details can be found in Section 3.5. We then compute the inner product between this segment feature and the target action categories' text features as

$$\mathcal{P} = \mathbf{F}_{\text{lan}} \cdot \left(\mathbf{F}_{\text{actn}}\right)^{\top}, \tag{15}$$

where $\mathcal{P} \in \mathbb{R}^{C\times T'}$ represents the classification output, with each column corresponding to a temporal location $t \in T'$. The t-th column, denoted as $\mathcal{P}_t \in \mathbb{R}^{C\times 1}$, represents the probability distribution over the C action categories at temporal location t. Combined with the actionness mask $\mathcal{M}$, we further compute the action category prediction $\mathcal{P}_{\text{actn}} \in \mathbb{R}^{C}$ for the segment as

$$\mathcal{P}_{\text{actn}} = \frac{\sum_{t=1}^{T'}\mathcal{M}(t)\mathcal{P}_t}{\sum_{t=1}^{T'}\mathcal{M}(t)}. \tag{16}$$

Here, we use the actionness mask as a weight to aggregate the classification results $\mathcal{P}$ over all foreground temporal points, because some frames may contain background or irrelevant information, while others may clearly indicate the occurrence of the action. The actionness mask $\mathcal{M}(t)$ serves as a mechanism to dynamically weigh the classification results at each temporal point. For foregrousegment classification, we train using the standard cross-entropy loss as

$$\mathcal{L}_{\text{cls}} = \text{CrossEntropy}(\mathcal{P}_{\text{actn}}, \hat{\mathcal{P}}_{\text{actn}}), \tag{17}$$

where $\hat{\mathcal{P}}_{\text{actn}} \in \mathbb{R}^{C}$ is ground-truth class label of the foreground segment.

3.5. Implementation Details

Following previous practices [17, 6, 12], we sample and rescale each video's feature to $T = 128$ and 256 temporal points for the ActivityNet and THUMOS datasets, respectively, using linear interpolation. The feature dimension D is set as 512. For action concept space, we set X, Z and N to 20, 100 and 2000, respectively. During the training phase, our overall objective loss function is $\mathcal{L} = \mathcal{L}_{\text{mc}} + \mathcal{L}_{\text{mask}} + \mathcal{L}_{\text{cls}}$. For calculating $\mathcal{L}_{\text{cls}}$, we use the ground truth start and end position labels of action segments to obtain the foreground segment feature $\mathbf{F}_{\text{actn}}$. During inference, we extract it using the predicted results from the actionness mask localizer. Our model is trained for 40 epochs using stochastic gradient descent (SGD) with the Adam optimization method, with a learning rate of 10^{-4}, on a GTX 3090 GPU. For the CLIP encoders, we freeze the parameters of both the image encoder and the text encoder. Therefore, during training, the temporal transformer, concept-guided semantic projection model, and action localization and classification heads update their parameters.

4. Experiments

4.1. Setup

We evaluate our method on two widely-used Temporal Action Detection (TAD) benchmarks: ActivityNet-v1.3 [13] and THUMOS14 [14]. ActivityNet-v1.3 consists of 19,994 videos across 200 action categories, split into training, validation, and testing sets in a 2:1:1 ratio. THUMOS14 contains 200 validation and 213 testing videos annotated with temporal boundaries from 20 action categories. For both datasets, we report mean Average Precision (mAP) at various temporal Intersection over Union (tIoU) thresholds: [0.3:0.1:0.7] for THUMOS14 and [0.5:0.05:0.95] for ActivityNet-v1.3. To evaluate in a open-vocabulary setting, we follow the protocol from [11], ensuring that the action categories in the training and testing sets are disjoint. Specifically, we use two splits: (1) training on 75% of the categories and testing on the remaining 25%, and (2) training on 50% and testing on the other 50%. Each setting is repeated with 10 random splits of the categories.

4.2. Comparison to State of The Arts

Comparison methods. In this section, we compare our approach with several existing OV-TAD methods, with results shown in Table 1. Following the setup from [8], we include TMaxer [48], ActionFormer [34], and TriDet [49], which were originally designed for closed-set TAD but have been adapted for the OV-TAD task. Additionally, following [12], we also include B-I, a two-stage model with BMN for proposals and CLIP with handcrafted prompts, and a one-stage model, referred to as B-II, which uses CLIP's pre-trained image encoder for dense prediction with TAD. Eff-Prompt [11], STALE [12], and DeTAL [8] are methods specifically designed for OV-TAD, and we also include them in our comparison.

Comparison results. As shown in Table 1, our method demonstrates superior performance compared to existing approaches across both the THUMOS14 and ActivityNet1.3 datasets, achieving higher mAP scores at multiple tIoU thresholds. Specifically, on ActivityNet1.3, our model achieves significant gains across all tIoU thresholds. For example, compared to STALE, an OV-TAD method with a similar one-stage design, a key difference is our introduction of concept-guided semantic projection. In the 75 Seen/25% Unseen and 50% Seen/50% Unseen splits, we achieve an average mAP of 28.1% and 25.7%, respectively, compared to STALE's 24.9% and 20.5%, suggesting that our approach might benefit from the Concept-guided Semantic Projection. This design likely helps the model capture underlying action concepts more effectively by leveraging semantic information, which could reduce over-reliance on specific visual features that might hinder generalization.

4.3. Ablation Study

To validate the effectiveness of our design, we conduct ablation experiments by removing one sub-module of the model, as shown in Table 2.

- w/o CSP: We remove the Concept-guided Semantic Projection model in Section 3.2, and replace $\mathbf{F}_{\text{con}}$ with $\mathbf{F}_{\text{vid}}$ in Eq. (13).
- w/o $\mathcal{L}_{\text{mc}}$: We remove the mutual contrastive loss $\mathcal{L}_{\text{mc}}$ in Eq. (12).
- $\mathcal{P}_{\text{actn}}$ w/o $\mathcal{M}$: We remove the actionness mask from the aggregation operation Eq. (16), replacing it with a simple average across all temporal points.
- w/o $\mathcal{T}$: We remove the transformer encoder $\mathcal{T}$ from Eq. (1).

As shown in Table 2, each component of our model contributes to the overall performance, with every ablation resulting in a decrease in mAP. Notably, removing the CSP model leads to the largest drop in mAP (3.2%), highlighting its crucial role in our model. Since we replace $\mathbf{F}_{\text{con}}$ with $\mathbf{F}_{\text{vid}}$, the subsequent task heads can no longer leverage the action semantics encoded in $\mathbf{F}_{\text{con}}$, which may result in a significant degradation in open-vocabulary performance. Additionally, the removal of the mutual contrastive loss $\mathcal{L}_{\text{mc}}$ also results in a noticeable decrease in performance. This loss is designed to ensure that the projection into the concept space is both semantically meaningful and discriminative, and removing it may reduce the effectiveness of the projection module.

Effectiveness of the concept-guided semantic projection. For the proposed CSP model, similar to [44], we replaced it with a simple dot product operation for comparison, as shown in Table 3.

- w dot product proj: We replace the proposed CSP model with the dot product operation.

After replacing the proposed CSP model with the dot product operation, the model's performance drops significantly, slightly above the result of directly removing the CSP model. This could be because, compared to using a simple dot product, our CSP model effectively preserves useful semantic features from the visual representations during the projection into the action concept space, proving the effectiveness of CSP.

Effectiveness of the mutual contrastive loss. To further investigate the effectiveness of the proposed Mutual Contrastive Loss, we conducted additional experiments by replacing it with other loss functions, as shown in Table 4.

- w $\mathcal{L}_{\text{ce}}$ loss: We replace the proposed $\mathcal{L}_{\text{mc}}$ with a simpler alignment loss, where cross-entropy (CE) loss is directly used to align each segment's concept feature s with its corresponding label feature e.
- w $\mathcal{L}_{\text{info}}$ loss: We replace the proposed $\mathcal{L}_{\text{mc}}$ with the standard infoNCE loss [46].

As shown in Table 4, when we replace $\mathcal{L}_{\text{mc}}$ with the sim-

Table 1. OV-TAD results on THUMOS14 and ActivityNet1.3, where the score for the top and the second best performances are bolded and underlined respectively. The methods marked with * indicate that they were originally designed for close-set TAD while adapted to the OV-TAD task in the experiment.

Data Split	Methods	THUMOS14						ActivityNet1.3			
		0.3	**0.4**	**0.5**	**0.6**	**0.7**	**Avg mAP**	**0.5**	**0.75**	**0.95**	**Avg mAP**
75% Seen 25 Unseen	TMaxer* [48]	19.8	17.2	14.2	10.8	7.6	13.9	20.0	11.5	0.5	11.5
	ActionFormer* [34]	22.7	20.0	16.5	12.8	8.5	16.1	25.0	15.1	2.0	15.2
	TriDet* [49]	25.9	22.5	18.2	13.1	8.2	17.6	25.5	15.2	2.0	15.3
	B-II [10]	28.5	20.3	17.1	10.5	6.9	16.6	32.6	18.5	5.8	19.6
	B-I [10]	33.0	25.5	18.3	11.6	5.7	18.8	35.6	20.4	2.1	20.2
	Eff-Prompt [11]	39.7	31.6	23.0	14.9	7.5	23.3	37.6	22.9	3.8	23.1
	STALE [12]	40.5	32.3	23.5	15.3	7.6	23.8	38.2	25.2	6.0	24.9
	DeTAL [8]	39.8	33.6	25.9	17.4	9.9	25.3	39.3	26.4	5.0	25.8
	Ours	**42.7**	**35.5**	**26.4**	**18.5**	**12.0**	**27.0**	**41.1**	**28.8**	**7.4**	**28.1**
50% Seen 50% Unseen	TMaxer* [48]	10.6	9.4	8.0	6.2	4.4	7.7	15.0	8.5	0.4	8.6
	ActionFormer* [34]	11.3	10.0	8.4	6.6	4.6	8.2	17.9	10.8	1.3	10.8
	TriDet* [49]	15.2	13.2	10.8	7.9	5.2	10.5	19.1	11.5	1.1	11.4
	B-II [10]	21.0	16.4	11.2	6.3	3.2	11.6	25.3	13.0	3.7	12.9
	B-I [10]	27.2	21.3	15.3	9.7	4.8	15.7	28.0	16.4	1.2	16.0
	Eff-Prompt [11]	37.2	29.6	21.6	14.0	7.2	21.9	32.0	19.3	2.9	19.6
	STALE [12]	38.3	30.7	21.2	13.8	7.0	22.2	32.1	20.7	**5.9**	20.5
	DeTAL [8]	38.3	32.3	24.4	16.3	9.0	24.1	34.4	23.0	4.0	22.4
	Ours	**41.2**	**33.4**	**24.8**	**17.3**	**10.9**	**25.5**	**38.4**	**26.4**	5.2	**25.7**

Table 2. Ablation study of the proposed method with its variations on ActivityNet1.3 dataset with 50% split.

Method	mAP			
	0.5	**0.75**	**0.95**	**Avg**
w/o CSP	35.4	23.0	3.9	22.5
w/o $\mathcal{L}_{\text{mc}}$	36.5	23.3	3.6	23.0
$\mathcal{P}_{\text{actn}}$ w/o $\mathcal{M}$	39.8	24.7	3.0	24.4
w/o $\mathcal{T}$	37.7	23.7	6.1	24.2
Ours	38.4	26.4	5.2	25.7

Table 3. Ablation study of concept projection on ActivityNet1.3 with 50% split.

Method	mAP			
	0.5	**0.75**	**0.95**	**Avg**
w/o CSP	35.4	23.0	3.9	22.5
w dot product proj	35.4	22.5	4.8	22.6
Ours	38.4	26.4	5.2	25.7

pler alignment loss $\mathcal{L}_{\text{ce}}$, the model's performance declines. This could be because *it directly pushes the concept feature of a segment to align closely with the corresponding label's text feature*, potentially leading to the loss of valuable information unique to the concept feature during training. However, in our proposed $\mathcal{L}_{\text{mc}}$, we optimize the *relative relationships between samples* by calculating cross-entropy (CE) loss over the similarity matrix in Eq. (12), avoiding the information loss caused by direct alignment. Additionally, when we replace $\mathcal{L}_{\text{mc}}$ with the standard contrastive learning loss (infoNCE loss), the model's performance also drops. Unlike our $\mathcal{L}_{\text{mc}}$, which *considers the relative semantic distance between segments*, the infoNCE loss simply trains the projection model by pulling the concept features of segments within the same class closer and pushing those from different classes further apart. This approach *ignores the relative distances between segments*, and treating all segments from different classes equivalently may hinder the model's ability to capture finer semantic distinctions. Finally, if we entirely remove $\mathcal{L}_{\text{mc}}$, the model experiences the largest performance drop, which suggests that the CSP model benefits from targeted optimization. Our proposed $\mathcal{L}_{\text{mc}}$ effectively leverages the mutual relationships between the action categories of the segments to optimize the relative relationships between their concept features, ensuring that the projection into the concept space is both semantically meaningful and discriminative.

4.4. In-depth Analysis of Action Concept Space

To analyze the impact of different concept spaces on the performance of our method, we built and compared concept spaces using various approaches. In Section 3.2, we utilized a large language model (LLM) and CLIP's text encoder to generate a concept space, referred to as concept space A.

Table 4. Ablation study of Mutual Contrastive Loss on ActivityNet1.3 with 50% split.

Method	mAP			
	0.5	**0.75**	**0.95**	**Avg**
w/o $\mathcal{L}_{mc}$	36.5	23.3	4.2	23.0
w $\mathcal{L}_{ce}$ loss	38.3	24.3	3.6	24.1
w $\mathcal{L}_{info}$ loss	37.2	24.7	4.2	24.3
Ours	38.4	26.4	5.2	25.7

Specifically, this concept space was created by prompting GPT-4 [50] to generate conceptually diverse action labels, which were then fed into CLIP's text encoder to produce concept vectors. Concept space A is a concept space that is not tied to any specific dataset and contains 2,000 concept vectors. Additionally, we constructed another concept space by extracting all action labels from three common action recognition datasets: Kinetics-400 [51], UCF-101 [52], and Moments in Time [53]. After removing duplicate and semantically redundant labels, we used CLIP's text encoder to generate 644 concept vectors, forming concept space B. As shown in Table 5, our method exhibits stable performance when using different concept spaces, with no significant fluctuations. This reflects the robustness of our approach.

Table 5. Analysis of concept space on ActivityNet1.3 with 50% split.

Method	mAP			
	0.5	**0.75**	**0.95**	**Avg**
concept space A	38.4	26.4	5.2	25.7
concept space B	38.2	25.0	2.8	24.9

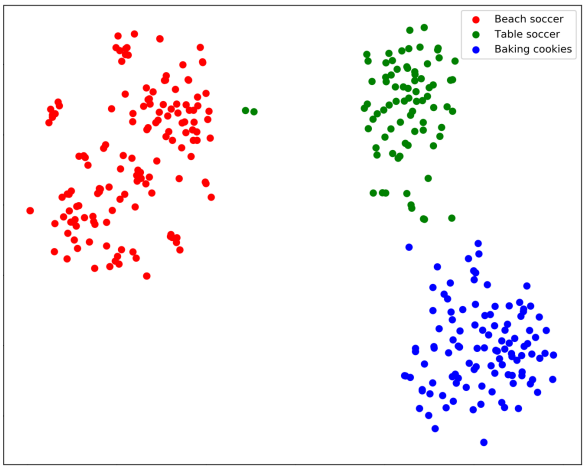
In Table 6, we further analyze the impact of using different numbers of concept vectors on the model's performance. In Section 3.2, we prompted GPT-4 [50] to categorize human activities into 20 broad categories and generate 100 action concepts for each category, resulting in 2,000 concept vectors. We randomly sampled 250 and 500 action concepts from these categories, obtaining 500 and 1,000 concept vectors, and conducted experiments using these subsets. As shown, while the model's performance slightly decreases as the number of concept vectors is reduced, it remains relatively stable and still outperforms other comparison methods. This further demonstrates the robustness of our approach.

4.5. Visualization of Projection Effect

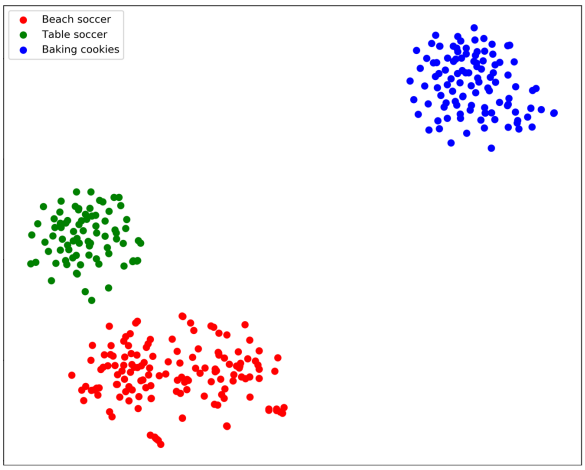
To demonstrate the impact of the Concept-guided Semantic Projection (CSP) module and the Mutual Contrastive Loss (MCL), we visualize the video clip features before and after projection using t-SNE, as shown in Figure 4.

Table 6. Analysis of the number of concept vectors on ActivityNet1.3 with 50% split.

concept number	mAP			
	0.5	**0.75**	**0.95**	**Avg**
500	39.0	26.0	4.1	25.1
1000	40.2	25.8	2.5	25.1
2000	38.4	26.4	5.2	25.7



(a) t-SNE Visualization of Video Clip Feature before projection



(b) t-SNE Visualization of Video Clip Feature after projection

Figure 4. t-SNE visualization of video clip features before (a) and after (b) projection.

Before projection, the semantic relationships between action categories are not well captured. For instance, some of the features of Table soccer (green points), which are semantically consistent remain relatively far apart from each other, while some of them are closer to the unrelated cate-

gory Baking cookies. This indicates that the raw features fail to encode the underlying action semantics effectively. After applying CSP and MCL, the features are projected into the action concept space, where semantic relationships are better preserved. Semantically consistent categories such as those of Table soccer are closer, while unrelated categories like Baking cookies are pushed further apart. Furthermore, MCL improves the intra-class compactness by aligning features within the same action label, as reflected in the tighter distributions of samples within each cluster.

5. Conclusion

In this work, we propose a novel framework for Open-Vocabulary Temporal Action Detection (OV-TAD) by introducing a Concept-guided Semantic Projection mechanism. We use the Concept-guided Semantic Projection (CSP) model to project video features into an action concept space, which effectively addresses the challenge of detecting unseen actions by leveraging their semantic information. Additionally, we introduced a Mutual Contrastive Loss (MCL), which ensures semantic consistency and improves feature discrimination in the concept space. Extensive experiments on the ActivityNet and THUMOS14 datasets demonstrate that the proposed model achieves superior performance. Ablation studies further confirm the effectiveness of the CSP and MCL in improving OV-TAD.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 62072334, 62402490, and the China Postdoctoral Science Foundation under Grant 2024M753397.

References

[1] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2914–2923, 2017. 1

[2] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang. Autoloc: Weakly-supervised temporal action localization in untrimmed videos. In *Proceedings of the european conference on computer vision (ECCV)*, pages 154–171, 2018. 1

[3] E. Vahdani and Y. Tian. Deep learning-based action detection in untrimmed videos: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):4302–4320, 2022. 1

[4] H. Xu, A. Das, and K. Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017. 1, 2

[5] J. Gao, Z. Yang, and R. Nevatia. Cascaded boundary regression for temporal action detection. *arXiv preprint arXiv:1705.01180*, 2017. 1

[6] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10156–10165, 2020. 1, 3, 6

[7] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4325–4334, 2017. 1

[8] Z. Li, Y. Zhong, R. Song, T. Li, L. Ma, and W. Zhang. Detal: Open-vocabulary temporal action localization with decoupled networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1, 3, 7, 8

[9] L. Zhang, X. Chang, J. Liu, M. Luo, S. Wang, Z. Ge, and A. Hauptmann. Zstad: Zero-shot temporal activity detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 879–888, 2020. 1, 3

[10] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 3, 8

[11] C. Ju, T. Han, K. Zheng, Y. Zhang, and W. Xie. Prompting visual-language models for efficient video understanding. In *European Conference on Computer Vision*, pages 105–124. Springer, 2022. 1, 3, 7, 8

[12] S. Nag, X. Zhu, Y.-Z. Song, and T. Xiang. Zero-shot temporal action detection via vision-language prompting. In *European Conference on Computer Vision*, pages 681–697. Springer, 2022. 1, 3, 6, 7, 8

[13] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015. 2, 7

[14] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017. 2, 7

[15] F. C. Heilbron, J. C. Niebles, and B. Ghanem. Fast temporal activity proposals for efficient detection of human actions in untrimmed videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1914–1923, 2016. 2

[16] V. Escorcia, F. Caba Heilbron, J. C. Niebles, and B. Ghanem. Daps: Deep action proposals for action understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 768–784. Springer, 2016. 2

[17] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3889–3898, 2019. 2, 3, 6

[18] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2

[19] Q. Liu and Z. Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11612–11619, 2020. 2

[20] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7094–7103, 2019. 2

[21] C. Zhao, A. K. Thabet, and B. Ghanem. Video self-stitching graph network for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13658–13667, 2021. 2

[22] J. Li, X. Liu, Z. Zong, W. Zhao, M. Zhang, and J. Song. Graph attention based proposal 3d convnets for action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4626–4633, 2020. 2

[23] L. Wang, H. Yang, W. Wu, H. Yao, and H. Huang. Temporal action proposal generation with transformers. *arXiv preprint arXiv:2105.12043*, 2021. 2

[24] J. Tan, J. Tang, L. Wang, and G. Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13526–13535, 2021. 2

[25] S. Chang, P. Wang, F. Wang, H. Li, and Z. Shou. Augmented transformer with adaptive graph for temporal action proposal generation. In *Proceedings of the 3rd International Workshop on Human-Centric Multimedia Analysis*, pages 41–50, 2022. 2

[26] T. Lin, X. Zhao, and Z. Shou. Single shot temporal action detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 988–996, 2017. 2

[27] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017. 2

[28] K. Yang, P. Qiao, D. Li, S. Lv, and Y. Dou. Exploring temporal preservation networks for precise temporal action localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. 2

[29] C. Wang, H. Cai, Y. Zou, and Y. Xiong. Rgb stream is enough for temporal action detection. *arXiv preprint arXiv:2107.04362*, 2021. 2

[30] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 344–353, 2019. 2

[31] Y. Huang, Q. Dai, and Y. Lu. Decoupling localization and classification in single shot temporal action detection. In *2019 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1288–1293. IEEE, 2019. 2

[32] B. Wang, Y. Zhao, L. Yang, T. Long, and X. Li. Temporal action localization in the deep learning era: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 2

[33] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3320–3329, 2021. 2

[34] C.-L. Zhang, J. Wu, and Y. Li. Actionformer: Localizing moments of actions with transformers. In *European Conference on Computer Vision*, pages 492–510. Springer, 2022. 2, 7, 8

[35] F. Cheng and G. Bertasius. Tallformer: Temporal action localization with a long-memory transformer. In *European Conference on Computer Vision*, pages 503–521. Springer, 2022. 2

[36] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018. 3

[37] F. Wang, J. Liu, S. Zhang, G. Zhang, Y. Li, and F. Yuan. Inductive zero-shot image annotation via embedding graph. *IEEE Access*, 7:107816–107830, 2019. 3

[38] M. Elhoseiny, Y. Zhu, H. Zhang, and A. Elgammal. Link the head to the" beak": Zero shot learning from noisy text description at part precision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5640–5649, 2017. 3

[39] Y. Zhu, M. Elhoseiny, B. Liu, X. Peng, and A. Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1004–1013, 2018. 3

[40] Y. Le Cacheux, A. Popescu, and H. Le Borgne. Webly supervised semantic embeddings for large scale zero-shot learning. In *Proceedings of the Asian Conference on Computer Vision*, 2020. 3

[41] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 59–68, 2016. 3

[42] Y. Goldberg. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014. 3

[43] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014. 3

[44] K. Ranasinghe and M. S. Ryoo. Language-based action concept spaces improve video self-supervised learning. *Advances in Neural Information Processing Systems*, 36:74980–74994, 2023. 3, 4, 7

[45] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 4

[46] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 5, 7

[47] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 6

[48] T. N. Tang, K. Kim, and K. Sohn. Temporalmaxer: Maximize temporal context with only max pooling for temporal

action localization. *arXiv preprint arXiv:2303.09055*, 2023. 7, 8

[49] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, and D. Tao. Tridet: Temporal action detection with relative boundary modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18857–18866, 2023. 7, 8

[50] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 9

[51] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 9

[52] K. Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 9

[53] M. Monfort, A. Andonian, B. Zhou, K. Ramakrishnan, S. A. Bargal, T. Yan, L. Brown, Q. Fan, D. Gutfreund, C. Vondrick, et al. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):502–508, 2019. 9