

# PLOT-TAL - Prompt Learning with Optimal Transport for Few-Shot Temporal Action Localization

Edward Fish, Jon Weinbren, and Andrew Gilbert

University of Surrey, Guildford, UK

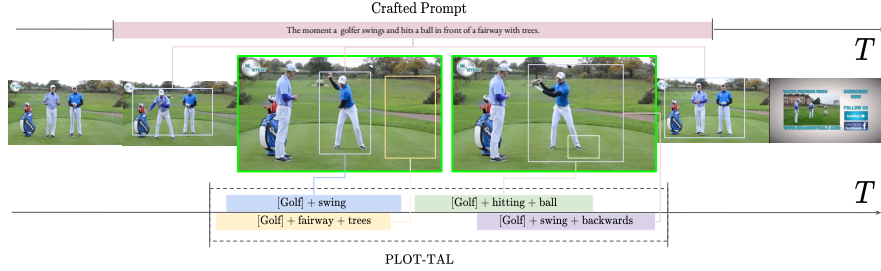
**Abstract.** This paper introduces a novel approach to temporal action localization (TAL) in few-shot learning. Our work addresses the inherent limitations of conventional single-prompt learning methods that often lead to overfitting due to the inability to generalize across varying contexts in real-world videos. Recognizing the diversity of camera views, backgrounds, and objects in videos, we propose a multi-prompt learning framework enhanced with optimal transport. This design allows the model to learn a set of diverse prompts for each action, capturing general characteristics more effectively and distributing the representation to mitigate the risk of overfitting. Furthermore, by employing optimal transport theory, we efficiently align these prompts with action features, optimizing for a comprehensive representation that adapts to the multifaceted nature of video data. Our experiments demonstrate significant improvements in action localization accuracy and robustness in few-shot settings on the standard challenging datasets of THUMOS-14 and EpicKitchens100, highlighting the efficacy of our multi-prompt optimal transport approach in overcoming the challenges of conventional few-shot TAL methods.

**Keywords:** Temporal Action Localization · Few Shot Learning · Video Understanding

## 1 Introduction

Temporal Action Localization (TAL) detects the onset and offset of actions and their class labels in untrimmed and unconstrained videos. The necessity for few-shot learning methods in temporal action localization (TAL) stems from the inherent challenge of annotating video data. Unlike static images, videos encompass temporal dynamics, requiring annotations for the presence of an action and its precise timing within the video stream. This requirement significantly increases the annotation burden, as annotators must watch videos in their entirety to accurately label action instances. Given the vast amount of video content generated daily and the wide variety of actions that can be performed, obtaining a comprehensive, annotated dataset for training traditional machine learning models is both time-consuming and costly. Few-shot learning emerges as a pivotal

solution to this challenge by aiming to generalize from a limited number of examples, thus reducing this dependency. However, because of the diversity of video content with respect to camera angles, temporal speeds, and locations, it's hard to design a system that can generalize to multiple views for classification while remaining discriminative in identifying the start and end of actions.



**Fig. 1:** Existing methods learn a single prompt to identify the location and class of a given action, but multiple complimentary views can help with both class generalization and temporal discrimination within the video. Green frames indicate the foreground features.

Current approaches to few-shot learning for temporal action localization take a meta-learning approach [34, 37], where each test video is aligned to a small subset of the training data in many ‘episodes.’ These methods require learning a model from initialization, with no priors, consuming large amounts of memory and compute. Adapting a pre-trained image encoder from a large-scale vision-language pre-trained (VLP) model such as CLIP [38] and Align [27] is one conventional paradigm for training. However, these networks are prone to overfitting in the few-shot scenario, where we have as few as five samples per a class to learn from. Furthermore, adapting image CLIP encoders to video ignores the rich temporal dynamics that can offer vital clues for classification.

A recent training paradigm, prompt learning, has been used to reduce the number of trainable parameters, where all parameters of the model are fixed, and a learnable context vector is added to the prompt to improve the alignment between the prompt and the image features. [8, 15, 60, 61]. However, as demonstrated in [8], single prompt learning methods will optimize towards the mean of all features in the image. As previously stated, in a temporal action localization setting, we need to determine the discriminative boundaries of an action. This means prompts will likely have high cosine similarity over a wide standard deviation of temporal features, leading to poor action boundaries. Instead, it would be beneficial to consider multiple prompts for each action, which could encompass many views over the video. For example, in Fig. 1, we show an illustrative comparison of a handcrafted prompt, which will align with many frames in the video segment,

and a collection of refined learnable context prompts that, when combined, can learn more discriminative features.

To achieve this, we present a prompt-learning framework with optimal transport (OT). OT is chosen as a learning strategy in few-shot temporal action localization because it effectively aligns distributions, leverages the geometric structure of the embedding space, handles variability in action characteristics, and offers robustness against noise and outliers. Additionally, OT’s ability to formulate a transport plan adapts to the distributional discrepancies between the visual and textual modalities improving generalization over multiple temporal speeds and contexts. Furthermore, the fine-grained matching facilitated by OT ensures that the model can accurately identify and classify actions based on nuanced differences in visual features while enhancing the model’s robustness to variations in action appearance, execution speed, and contextual settings.

Our contributions are as follows.

- We introduce a new few-shot prompt-learning methodology for Temporal Action Localization that removes the requirement for extensive episodic meta-training.
- We demonstrate the effectiveness of optimizing multiple views to improve discriminative boundaries in the action localization task.
- We present an efficient method of optimizing the transport maps between multiple temporal views and prompts via a temporal feature pyramid to account for various temporal speeds and action scales.

Through this approach, we aim to significantly reduce the barrier to entry for TAL applications, making it more feasible to apply these technologies to a broader range of domains and datasets with limited annotations.

## 2 Related Work

### 2.1 Temporal Action Localization (TAL)

Methods in TAL can be separated into single- and two-stage. Where two-stage methods generate a large number of proposal segments which are then passed to a classification head [5, 9, 17, 17, 21, 22, 28, 30, 31, 31, 33, 59]. These methods include the use of graph neural networks [1, 51, 51, 57] and more recently, transformers [7, 42, 48]. Recent progress in single-stage TAL has shown improvements in accuracy and efficiency over two-stage methods, combining both action proposal and classification in a single forward pass. Works inspired by object detection [32, 39], saliency detection [29], and hierarchical CNN’s [29, 54, 55] all combine proposal and classification. Current SOTA methods in TAL utilize transformer-based [45] feature pyramid networks (FPN’s) [11, 40, 49, 58], which combine multi-resolution transformer features with classification and regression heads.

## 2.2 Prompt Learning

Prompt learning is a methodology that introduces learnable context prompts to enhance generalization and robustness in various visual understanding tasks. Initially introduced in few-shot image recognition by CoOP [61] and later expanded upon by CoCoOp [62], prompt learning has demonstrated its efficacy in improving open-world visual understanding. This approach has also found applications in action recognition [24] and video-to-text alignment [27], facilitating tasks such as video-question answering. While the adaptation of visual-language models for video has predominantly focused on video retrieval [2, 15, 16, 19], there have been very few works exploring how to leverage VIL models for temporal action localization efficiently. In [49], the authors use prompt learning for several video tasks, including temporal activity localization, but in a fully supervised setting. Also, in [34, 35], the authors use learnable prompts as part of a masked transformer network for classifying video region proposals.

## 2.3 Few Shot Learning for Action Localization

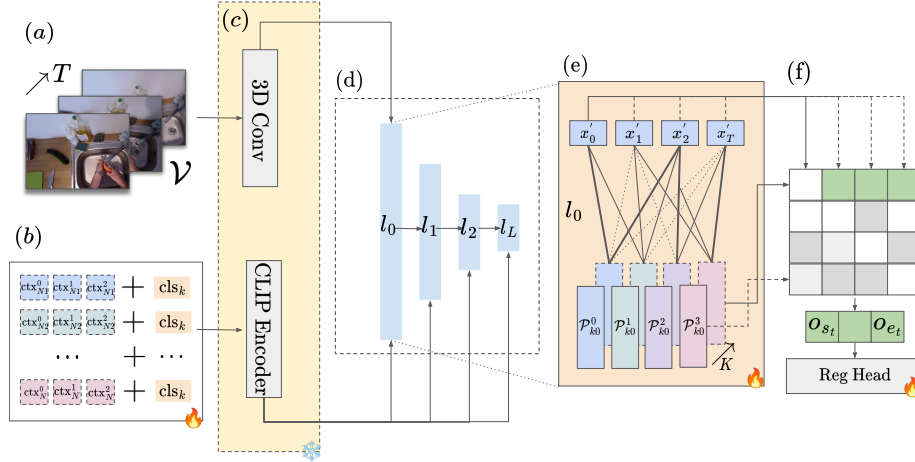
In [52], the authors introduce few-shot action-class localization in time, where a few (or at least one) positive labeled and several negative labeled videos steer the localization via an end-to-end meta-learning strategy. The strategy uses sliding windows to swipe over the untrimmed query video to generate fixed boundary proposals. In [50], the authors temporally localize an action from a few positive and negative labeled videos. They adopt a regional proposal network to produce proposals with flexible boundaries. In [36] and [49], the authors propose the challenge of zero-shot temporal action localization. However, these methods use external class scores from Untrimmed Net [47] for labeling proposals, so it is hard to evaluate their true potential in few-shot cases. In [34], the authors introduce a few-shot prompt meta-learning using additional multi-modal learnable context prompts with a transformer architecture. However, they train and evaluate on a 5-way / 5-shot meta-learning strategy and also use score fusion for the classification results.

## 2.4 Optimal Transport

The Optimal Transport (OT) problem, rooted in the work of Gaspard Monge in the 18th century [46], seeks the most cost-efficient way of transporting mass from one distribution to another. This problem has found profound applications in various fields, including economics [20], fluid dynamics [10], and, more recently, machine learning and computer vision [44].

Despite its conceptual appeal, the OT problem’s computational expense and sensitivity to outliers limited its direct applicability, especially in high-dimensional spaces common in machine learning and computer vision, until entropy regularization to the OT problem [12]. This regularization term leads to a problem

that can be efficiently solved using iterative scaling algorithms, notably Regularization. Regularization yields a unique, stable solution and enables leveraging parallel computing architectures, dramatically reducing computation time [3, 13]. In [53], the authors employ optimal transport for efficient attention assignment between optical flow, and RGB features to learn a structure matrix that captures dependencies among modalities in each frame. In [8], the authors demonstrated a novel application of optimal transport to align multiple prompts to feature maps for few-shot image classification. The paper proposed formulating the local visual features and multiple prompts as the samplings of two discrete distributions and using OT to encourage fine-grained cross-modal alignment. The work extracts feature maps as visual representations using the CLIP multi-head self-attention layer outputs. In this work, we assess the feasibility of applying a similar method to the task of temporal activity localization, where we can formulate the optimal transport between temporal features and multiple prompts as the sampling of two discrete distributions.



**Fig. 2:** An overview of the approach. **A.)** We sample  $T$  overlapping segments of videos  $\mathcal{V}$ . **B.)** For each class label  $K$ , we randomly initialize  $N$  learnable vectors concatenated with the class label. **C.)** Video features are extracted via a pre-trained 3D CNN encoder (I3D) while  $N$  prompts for each class  $k$  are also extracted via the pre-trained CLIP text encoder. **D.)** We temporally downsample the features using max-pooling. **E.)** We search the optimal transport plan between the  $N$  prompt features and video segments at each temporal level. Following this stage, we sum all  $N$  vectors for each  $K$ . **F.)** At each temporal level  $L$ , we compute the cosine similarity between each prompt vector  $P_k$  and each video segment  $x_i$  and then apply a threshold to retrieve action candidates. These candidates are passed to the regression head, which minimizes the distance between the start and end actions and each embedding. Only components with the fire symbol are trained, and all others are frozen.

### 3 Method

We propose a novel framework for Temporal Action Localization (TAL) in untrimmed videos. Our approach integrates pre-trained feature extraction, adaptive prompt learning, and efficient feature-prompt alignment via the Sinkhorn algorithm. An overview of the method is shown in Fig 2.

#### 3.1 Problem Definition

In addressing few-shot TAL, our framework aims to learn a generalizable representation for each action instance tuple,  $(s_i, e_i, a_i)$ , where  $s_i$  represents the starting time or onset of the action instance,  $e_i$  denotes the ending time or offset of the action instance, and  $a_i$  specifies the action category or label. We use minimal annotated examples, optimizing for the accurate classification of action types and precise localization of their temporal boundaries. Integrating pre-trained feature extractors minimizes the model’s dependency on extensive training data, aligning with the resource-intensive nature of video processing tasks.

Considering an untrimmed input video as  $\mathcal{V}$ , we represent it as a set of feature vectors tokenized as  $\mathcal{V} = \{x_1, x_2, \dots, x_T\}$ . Each  $x_t$  corresponds to discrete time steps,  $t = \{1, 2, \dots, T\}$ . Notably, the total duration  $T$  is not constant and may differ across videos. For illustrative purposes,  $x_t$  can be envisaged as a feature vector extracted from a 3D convolutional network at a specific time  $t$  within the video. The primary objective of TAL is to identify and label action instances in the input video sequence  $\mathcal{V}$ . These instances are collectively denoted as  $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$ , where  $N$  signifies the total number of action instances in a given video. This value can be variable across different videos.

The parameters must adhere to the conditions:  $s_i, e_i \in \{1, \dots, T\}$ ,  $a_i \in \{1, \dots, C\}$  (with  $C$  indicating the total number of predefined action categories), and  $s_i < e_i$ , which ensures the starting time precedes the ending time for every action instance.

In the few-shot setting, we aim to learn some general representation of each action instance  $y_i$  using only a limited number of annotations that we can later classify the action onsets, offsets, and class labels of unseen videos. Since video understanding tasks are typically resource and data-intensive, we also want to minimize the number of trainable parameters in the model.

#### 3.2 Feature Extraction and Representation

Given the untrimmed input video  $\mathcal{V}$ , we extract a sequence of feature vectors  $\{x_1, x_2, \dots, x_T\}$  corresponding to each time step  $t$ , using a 3D convolutional network. The extraction process is formalized as:

$$x_t = f_{\text{CNN}}(v_t), \quad t = 1, 2, \dots, T, \quad (1)$$

where  $v_t$  denotes the input from the video at time  $t$ , and  $f_{\text{CNN}}$  represents the 3D convolutional network function.

To further refine these features and incorporate contextual information, we apply a 1D convolutional layer:

$$x'_t = f_{\text{conv}}(x_t), \quad t = 1, 2, \dots, T \quad (2)$$

Where  $f_{\text{conv}}$  denotes the convolutional operation which is designed to enhance the local temporal feature representation as demonstrated in [41, 43, 58].

### 3.3 Adaptive Prompt Learning

To ensure we can align with multiple views in various temporal dimensions, we introduce additional learnable prompts for each class in the few-shot training set. For each action category  $k$ , we generate  $N$  prompts  $\mathcal{P}_k = \{P_{k1}, P_{k2}, \dots, P_{kN}\}$ , each consisting of the class label and  $n_{\text{ctx}}$  context vectors, encoded as:

$$P_{ki} = f_{\text{CLIP}}(\text{label}_k, \text{ctx}_{k1}, \dots, \text{ctx}_{kn_{\text{ctx}}}), \quad (3)$$

Where  $f_{\text{CLIP}}$  signifies the encoding function from a pre-trained CLIP model, integrating the semantic content of the action categories into the model.

### 3.4 Optimal Transport with Sinkhorn Algorithm

We aim to align each class's  $N$  learnable prompts with the most similar video features in cosine similarity. This is performed via optimal transport with the Sinkhorn Algorithm [12] to ensure the method is tractable and efficient.

To align the refined video features  $\{x'_1, x'_2, \dots, x'_T\}$  with the adaptive prompts  $\mathcal{P}_k$  for each action category  $k$ , we employ the Optimal Transport (OT) metric as a critical tool. The OT metric quantifies the discrepancies between two distributions, which, in our context, are the distributions of video features and prompt embeddings. Formally, let us denote the distributions of video features and prompts as:

$$U = \sum_{t=1}^T u_t \delta_{x'_t} \quad \text{and} \quad V_k = \sum_{i=1}^N v_{ki} \delta_{P_{ki}}, \quad (4)$$

where  $\delta_{x'_t}$  and  $\delta_{P_{ki}}$  represent Dirac delta functions centered at the video feature  $x'_t$  and prompt embedding  $P_{ki}$ , respectively. The vectors  $u$  and  $v_k$  are normalized such that  $\sum_{t=1}^T u_t = 1$  and  $\sum_{i=1}^N v_{ki} = 1$ , ensuring they represent discrete probability distributions.

The cost matrix  $C$ , with elements  $C_{ti}$ , defines the cost of transporting mass from video feature  $x'_t$  to prompt embedding  $P_{ki}$ . The cost is typically inversely related

to the similarity between  $x'_t$  and  $P_{ki}$ , such as  $C_{ti} = 1 - \text{sim}(x'_t, P_{ki})$ . The goal of OT is to find a transport plan  $T$  that minimizes the total transport cost:

$$d_{\text{OT}}(U, V_k | C) = \min_T \langle T, C \rangle, \quad \text{subject to} \quad T \mathbf{1}_N = u, \quad T^\top \mathbf{1}_T = v_k, \quad T \in \mathbb{R}_+^{T \times N}. \quad (5)$$

Due to the computational intensity of solving this problem, we apply the Sinkhorn Algorithm [12] with entropic regularization for efficient optimization. The regularization introduces an entropy term  $h(T) = -\sum_{t,i} T_{ti} \log T_{ti}$  to the optimization objective:

$$d_{\text{OT},\lambda}(U, V_k | C) = \min_T \langle T, C \rangle - \lambda h(T), \quad \text{subject to} \quad T \mathbf{1}_N = u, \quad T^\top \mathbf{1}_T = v_k. \quad (6)$$

The Sinkhorn algorithm iteratively adjusts  $T$  to satisfy the constraints efficiently, using updates based on matrix scaling operations. The iterative process converges to an optimal transport plan  $T^*$ , representing the optimal alignment between video features and prompts. This alignment guides identifying and classifying action instances by optimally matching video segments to their corresponding semantic labels.

### 3.5 Temporal Pyramid and Feature Integration

Since actions can occur at a wide range of speeds and temporal intervals, we utilize a temporal feature pyramid network [43] to optimize prompt alignments over multiple temporal scales. To do so, we construct a temporal pyramid from the refined features  $\{x'_1, x'_2, \dots, x'_T\}$  by applying a max-pooling operation at each level of the pyramid with a stride of 2, effectively halving the temporal dimension at each step:

$$X'_l = \text{MaxPool}(X'_{l-1}), \quad l = 2, \dots, L, \quad (7)$$

where  $X'_l$  represents the set of features at level  $l$  of the pyramid. This hierarchical structure is crucial for integrating temporal information across different scales, enabling the nuanced capture of action dynamics from coarse to fine temporal resolutions.

### 3.6 Multi-Resolution Temporal Alignment

For each level of the temporal pyramid, we employ the Optimal Transport (OT) metric to align the video features at that scale,  $\{x'_{1,l}, x'_{2,l}, \dots, x'_{T,l}\}$ , with the adaptive prompts  $\mathcal{P}_k$  corresponding to each action category  $k$ . This alignment is performed separately at each level  $l$  of the feature pyramid, allowing for a multi-scale analysis sensitive to the temporal granularity of actions.

Incorporating the notion of pyramid levels into the OT framework allows for a hierarchical approach to feature matching, accommodating various scales of



feature distribution alignment. At each pyramid level  $l$ , we define the OT problem as follows:

$$d_{\text{OT},\lambda}(U_l, V_{k,l}|C_l) = \min_{T_l} \langle T_l, C_l \rangle - \lambda h(T_l), \quad (8)$$

subject to  $T_l \mathbf{1}_N = u_l$ ,  $T_l^\top \mathbf{1}_T = v_{k,l}$ , for each pyramid level  $l$ . Here,  $C_l$  represents the cost matrix at level  $l$ , encapsulating the dissimilarity between the distributions of video features  $U_l$  and prompts  $V_{k,l}$  at this specific scale. This formulation ensures that the transport plan  $T_l$  minimizes the cost of transporting the distribution  $U_l$  to  $V_{k,l}$  while conforming to the constraints that ensure the distributions' masses are preserved.

We then adopt the same two-stage optimization process as proposed in [8]:

**Inner Loop** Within the inner loop, for each level  $l$  of the temporal pyramid, we fix the feature sets  $F_l$  and prompt sets  $G_{k,l}$ , and minimize the OT distance to optimally align  $G_{k,l}$  to  $F_l$ . The cost matrix  $C_l$  is computed to reflect the cosine similarity between the features and prompts at that scale:

$$C_l = 1 - F_l^\top G_{k,l}. \quad (9)$$

This minimization results in an optimized transport plan  $T_l^*$  and the corresponding OT distance  $d_{\text{OT},l}(k)$ .

**Outer Loop** In the outer loop, with the transport plans  $T_l^*$  determined for each level of the pyramid, we update the learnable vectors across all scales. This holistic optimization ensures that our model can adaptively align video features with textual prompts at varying temporal resolutions, enhancing its ability to localize and classify actions within untrimmed videos accurately.

### 3.7 Decoder Architecture

Following the multi-scale alignment of video features with adaptive prompts through the Optimal Transport framework, our decoder architecture is designed to leverage these aligned features for sequence labeling and action boundary detection. Unlike conventional CNN-based decoders, our approach utilizes the optimally aligned video features across each scale of the temporal pyramid to predict a sequence of action labels  $\Psi = \{\hat{\psi}_1, \hat{\psi}_2, \dots, \hat{\psi}_T\}$ .

For each temporal scale  $l$ , the decoder generates a probability distribution for action classifications using a sigmoid activation function:

$$C_l = \sigma(x'_{t,l}), \quad (10)$$

where  $x'_{t,l}$  denotes the aligned video feature at time  $t$  and scale  $l$ .

Furthermore, to accurately predict the start and end times of actions, a lightweight regression mechanism is employed:

$$O_l = \text{ReLU}(\mathbf{W}_o \cdot x'_{t,l}), \quad (11)$$

where  $\mathbf{W}_o$  represents trainable weights for the regression task.

By integrating the optimally aligned features from multiple scales of the temporal pyramid, the decoder architecture effectively enhances the model’s capability to recognize and localize actions, considering the diverse temporal scales inherent in video data. This multi-scale, feature-aligned approach ensures superior performance in both action classification and localization tasks, addressing the challenges posed by the temporal variation and complexity of actions in untrimmed video content.

### 3.8 Learning Objective

The learning objective aims to minimize the total loss, encompassing TAL’s (Temporal Action Localization) classification and localization aspects. This is achieved through a combination of Focal Loss for handling class imbalance in action classification and DIOU Loss for improving the accuracy of action boundary predictions:

$$L_{\text{total}} = \sum_{t=1}^T (L_{\text{cls}}(\hat{c}_t, c_t) + \mathbb{1}_{(c_t > 0)} L_{\text{reg}}(\hat{o}_{s_t}, \hat{o}_{e_t}, o_{s_t}, o_{e_t})) \quad (12)$$

Where  $L_{\text{cls}}$  is the classification loss computed using Focal Loss,  $L_{\text{reg}}$  is the regression loss computed using DIOU Loss,  $\hat{c}_t$  and  $c_t$  represent the predicted and true action categories, respectively, and  $\hat{o}_{s_t}, \hat{o}_{e_t}, o_{s_t}, o_{e_t}$  denote the predicted and true start and end times of actions. The indicator function  $\mathbb{1}_{(c_t > 0)}$  ensures that regression loss is only applied to positive samples, i.e., time steps where an action is present.

## 4 Evaluation

We evaluate our method against two standard benchmark datasets for Temporal Activity Localization and report our results. Unless otherwise stated, we randomly select five samples for each class in each dataset, train for 100 epochs, and evaluate over the whole test set. In the few-shot setup, this is referred to as 5-shot,  $C$ -way. In THUMOS’14 [23] this is 5-shot 200-way, and for EPIC-Kitchens [14] it is 5-shot 300-way for nouns and 5-shot 97-way for the verb partition.

## 4.1 Datasets

**THUMOS’ 14** The THUMOS’14 dataset [23] is a benchmark dataset for the Temporal Action Localization (TAL) task, deriving from the THUMOS challenge. It is specifically designed to evaluate the ability of algorithms to identify and temporally localize actions within untrimmed videos accurately. The dataset features 5600 temporally annotated actions over 200 untrimmed training videos and 213 untrimmed test videos. We extract overlapping segments from the training videos to evaluate the dataset using the I3D network pretrained on Kinetics 400. The video consists of real-world videos with shot changes and diverse camera and object views. For features we encode 16 frames per a time step with a stride of 8 following existing TAL feature extraction setups [11, 34, 36, 37, 58]. The 3D CNN for THUMOS’14 features is Inception I3D [6] network pretrained on Kinetics-400. [63]

**EPIC-Kitchens 100** EPIC-Kitchens 100 [14] is an egocentric dataset containing two tasks: noun localization (e.g., door) and verb localization (e.g., open the door). It has 495 and 138 videos, with 67,217 and 9,668 action instances for training and inference, respectively. The number of action classes for nouns and verbs is 300 and 97. We follow all other methods [11, 30, 43, 57, 58], and report the mean average precision (mAP) at different intersections over union (IoU) thresholds with the average mAP computed over [0.1:0.5:0.1]. We use the features provided by existing works in TAL [30, 51, 58]. Features are extracted using a SlowFast network [18] pre-trained on Kinetics-400 [63]. During extraction we use a 32-frame input sequence with a stride of 16 to generate a set of 2304-D features.

## 4.2 Comparative Methods

To the best of our knowledge, no current methods approach the task of few-shot temporal action localization with multiple prompt learning for each class. In [34], the authors present a multi-modal setup with single prompt learning. However, they train the network in a meta-learning setup, train and test on disjoint sets, and use score fusion (combining scores from UntrimmedNet [47]). Therefore, we assess our method’s effectiveness against SOTA prompt learning frameworks and apply them to the task of TAL. First, for CoOP [61] we initialize 16 learnable ctx tokens for each prompt. We also include two further baselines. Baseline *I* removes the optimal transport component labeled section (e) in Fig. 2 by taking the mean of the  $N$  learnable prompts to form one prompt for further processing in section (f). We also apply a linear probe (Baseline *II* - LP) as outlined in [38] and [8] replacing sections (e) and (f) with local self-attention and a CNN layer directly to the pre-trained I3D embeddings.

**Table 1:** Performance comparison of our proposed method PLOT-TAL on the THUMOS dataset against baselines.

Method	mAP@0.3	mAP@0.4	mAP@0.5	mAP@0.6	mAP@0.7	Avg (mAP)
Baseline I (avg)	37.3	32.93	26.88	18.17	8.83	24.82
Baseline II (lp)	51.98	46.5	36.79	25.62	14.66	35.11
CoOP	48.73	43.67	36.64	27.24	16.97	34.65
PLOT-TAL CLS	53.46	48.93	38.2	30.2	18.8	38.24
PLOT-TAL Verbose	<b>56.42</b>	<b>50.54</b>	<b>42.48</b>	<b>32.35</b>	<b>21.17</b>	<b>40.59</b>

### 4.3 Results

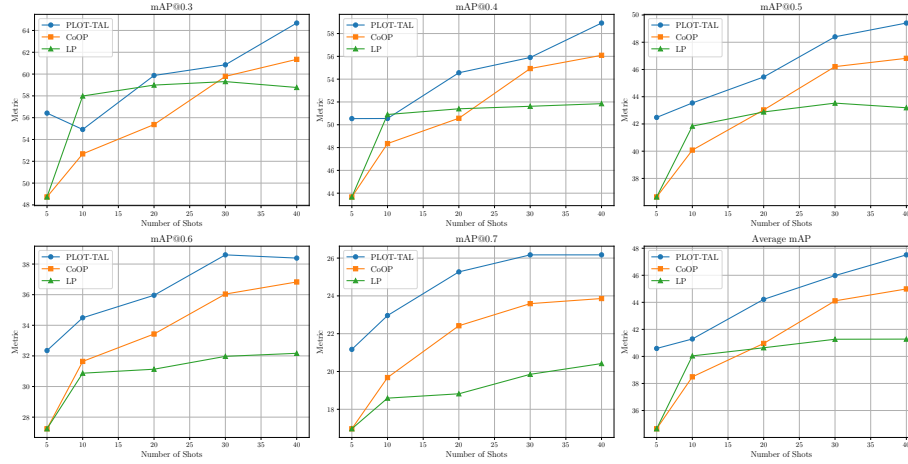
This section evaluates our approach against existing methods for both few-shot temporal action localization and prompt learning. To compare with previous works, we report the mean average precision (mAP) at various intersections over union for all results. In Tab. 1, we show results for 5-shot 20-way TAL on the THUMOS’14 dataset for our approach *PLOT-TAL CLS*. Adding additional class prompts can improve performance over a single prompt by a large margin ( $\uparrow 5.9$ ). We also show how it’s possible to achieve higher accuracy by handcrafting prompts (Verbose). In this setting, we use GPT-3.5 [4] to produce additional descriptions of the actions that will replace the class label.

The Baseline *I* method represents performance when we add additional prompts but exclude optimal transport, demonstrating how optimal transport is highly effective at aligning the features ( $\uparrow 15.77$ ). While Baseline *II* based on the work of [38] and [8] has an average performance of 5% less than our method, demonstrating the importance of the sections (e) and (f).

In Fig. 3, we demonstrate how the optimal transport improves performance at higher IoU thresholds than single prompt or linear probe methods. At low IoU thresholds, the predicted segment only needs to overlap with a small section of the ground truth, meaning that single prompt methods and linear probes achieve relatively good performance as they distribute the attention between prompts and features across the temporal domain. However, as we increase the IoU threshold, we can see that our PLOT-TAL method becomes more effective, demonstrating the network’s higher discriminative ability.

In Tab 2, we show results on the EPIC Kitchens verb and noun partitions, showing a small improvement over single prompt methods for the noun classes ( $\uparrow 1.19$ ) but achieve a larger performance boost for the verb classes ( $\uparrow 2.96$ ).

This demonstrates the effectiveness of the additional prompts in discriminating between temporal features, while for nouns, the features are primarily spatial and can be localized well with a single prompt. In Tab. 3, we compare with other SOTA methods for few-shot temporal action localization, which utilize meta-learning and perform few-shot localization at a 5-shot, 5-way setting, whereas our results are from the 5-shot, 20-way configuration. Not only is the 5-shot,



**Fig. 3:** mAP over various IoU thresholds for the THUMOS’ 14 dataset. We show that the additional prompts improve tray performance over a single learnable prompt, as in CoOp.

**Table 2:** Performance comparison on EPIC-Kitchens dataset for noun and verb recognition.

Method	Epic-Kitchens Noun (mAP)						Epic-Kitchens Verb (mAP)					
	0.1	0.2	0.3	0.4	0.5	Avg	0.1	0.2	0.3	0.4	0.5	Avg
Baseline I	14.3	13.5	13.1	10.3	9.3	12.1	21.2	19.9	18.0	15.2	11.9	17.3
Baseline II	<b>18.0</b>	15.4	14.1	12.2	9.5	13.9	<b>22.5</b>	<b>21.3</b>	19.2	17.1	13.3	18.7
CoOp	16.1	15.0	13.8	11.8	9.5	13.3	18.5	17.6	16.3	14.6	12.5	15.9
PLOT-TAL cls	17.9	<b>16.7</b>	<b>15.1</b>	<b>12.7</b>	<b>10.0</b>	<b>14.5</b>	21.8	20.9	<b>19.4</b>	<b>17.6</b>	<b>14.6</b>	<b>18.9</b>

20-way few-shot setting more challenging, but PLOT-TAL also benefits from being trained end-to-end without the requirement for pre-training and episodic adaptive contrastive learning as in current meta-learning approaches.

In Tab. 4, we perform an ablation experiment on the number of learnable prompts  $N$ . The results show the optimum number of prompts is  $N = 6$ , while with an increased number of prompts, e.g.,  $N = 10$ , we can achieve better results in the more difficult IoU thresholds. This is due to the increased temporal discriminative ability of the additional prompts.

#### 4.4 Visualisation Results

In Fig. 4, we show the normalized transport cost for each frame and  $N$  embedding for the class label ‘Cricket Shot’. This figure shows how each of the  $N$  prompts

**Table 3:** Additional comparisons with existing Meta-Learning (ML), Prompt Learning (PL), and End to End (E2E) methods for few-shot temporal action localization on the THUMOS’14 dataset. **ML** results are not directly comparable due to different training and testing partitions and experimental setups.

Method	Approach	Shot/Way	Avg (mAP)
Common Action Localization [56]	ML	5/5	22.8
MUPPET [34]	ML + PL	5/5	24.9
Multi-Level Alignment [26]	ML	5/5	31.8
Query Adaptive Transformer [37]	ML	5/5	32.7
CoOP [62]	E2E + PL	5/20	34.65
PLOT TAL CLS	E2E + PL	5/20	<b>38.24</b>
PLOT-TAL Verbose	E2E + PL	5/20	<b>40.59</b>



**Fig. 4:** The normalized transport cost of each  $N$  prompt for the class ‘Cricket Shot’ after training. Prompt one aligns with global information, while the other prompts learn additional, complementary views. In the transport cost algorithm, a lower value indicates closer alignment.

**Table 4:** Ablation experiment varying the number of additional learnable prompts for each class.

N prompts	0.3	0.4	0.5	0.6	0.7	avg
N=4	55.88	50.21	<b>43.06</b>	31.97	21.16	40.46
N=6	<b>56.42</b>	<b>50.54</b>	42.48	32.35	21.17	<b>40.59</b>
N=8	53.60	48.72	41.74	31.68	20.70	39.29
N=10	54.96	50.27	43.45	<b>32.53</b>	<b>21.44</b>	40.53
N=12	53.74	48.25	41.02	30.57	20.06	38.73
N=14	54.25	48.94	40.90	30.78	18.86	38.75
N=16	53.66	48.28	41.04	30.84	20.15	38.79

diverge and focus on different elements and views within the videos. For example, we can see that  $N_1$  or Prompt 1 learns global information across all frames. This shows how, in a single prompt framework, we may distribute alignment across all frames and lose discriminative ability, since it learns global information over the whole video. In the figure, we can note that Prompt 4 appears to learn background information and is more closely aligned to frames where we can see the stadium stands. Prompts 2 and 3, however, indicate a closer alignment with objects related to the class of ‘cricket shot’, including when the cricket strip is in the shot and there are people on the field.

#### 4.5 Number of Learnable Context Tokens

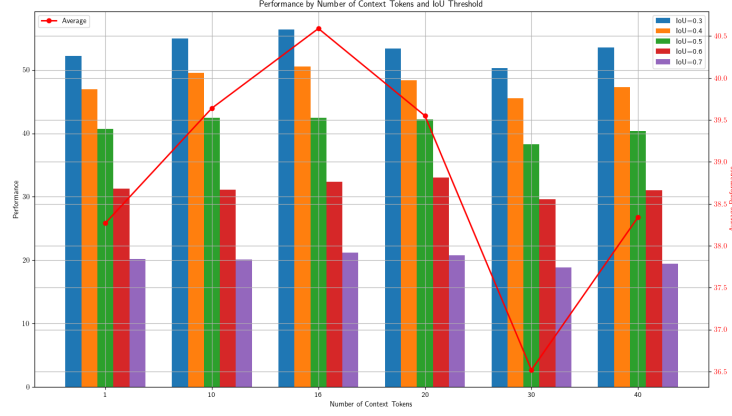
Each prompt has several learnable context tokens as described in [60] and [49]. These context tokens are randomly initialized so that for the class ‘Basketball Dunk’ with 4 *ctx* tokens, the full prompt will be

$$P = \{X, X, X, X, \text{Basketball Dunk}\} \quad (13)$$

In Fig. 5 and Tab. 5, we show the effect of varying the number of learnable *ctx* tokens appended to each prompt. For each  $N$  prompt,  $n_{ctx}$  tokens are randomly initialized. The figure shows that the optimum number of tokens is between 10 and 20. As per the existing literature [60, 61], we select 16 tokens for all methods unless otherwise stated and train and test using the 5-shot, 20-way setup as described in the paper.

#### 4.6 Visual Feature Embeddings

To evaluate the effectiveness of adding motion information via optical flow, we also performed additional experiments using only the RGB embeddings, the optical flow embeddings, and RGB CLIP embeddings from a ViT-B-16 encoder, with results shown in Tab. 6. The results show that the CLIP embeddings



**Fig. 5:** mAP over various IoU thresholds for the THUMOS' 14 dataset with variable number of additional context tokens appended to each  $N$  prompt.

**Table 5:** Ablation experiment on the number of context tokens on the THUMOS'14 Dataset.

Ctx Tokens	0.3	0.4	0.5	0.6	0.7	avg
1	52.25	46.94	40.73	31.26	20.17	38.27
10	54.94	49.55	<b>42.49</b>	31.14	20.08	39.64
16	<b>56.42</b>	<b>50.54</b>	42.48	32.35	<b>21.17</b>	<b>40.59</b>
20	53.39	48.38	42.19	<b>33.00</b>	20.78	39.55
30	50.27	45.54	38.30	29.64	18.83	36.52
40	53.55	47.30	40.35	31.06	19.46	38.34



perform better than the RGB embeddings from the I3D network  $\uparrow 2.67$ . This is because of the implicit alignment between the image and text encoder embeddings before temporal convolution. However, when combined with optical flow, the performance is improved by a large margin of  $\uparrow 7.56$ , demonstrating the improved classification ability of the network when we add additional temporal information via optical flow.

**Table 6:** Comparison of mAP scores for various visual input embeddings on the THUMOS’14 dataset.

Embeddings	0.3	0.4	0.5	0.6	0.7	avg (mAP)
CLIP	46.99	42.09	34.26	25.34	15.82	32.90
RGB	43.13	38.76	31.71	23.15	14.46	30.24
Optical Flow	26.03	23.10	19.54	14.07	8.93	18.33
RGB + Flow	<b>55.88</b>	<b>50.21</b>	<b>43.06</b>	<b>31.97</b>	<b>21.16</b>	<b>40.46</b>

#### 4.7 Prompt Engineering

We demonstrate how including crafted prompts can help to boost performance. In Tab. 7, we show the prompts generated by GPT 3.5 with the prompt - ‘*Generate prompts for a temporal action localization task for the following class IDs. The prompts should include objects, the action, and some indication of the moment when the action occurs.*’. We anticipate that further prompt engineering strategies will yield improved results.

## 5 Implementation Details

Here we share additional information about the implementation of the model architecture.

### 5.1 Feature Extraction

Features are extracted from a pre-trained I3D network [63] trained on the Kinetics-600 dataset [25, 63] in a supervised setting. We extract the optical flow and RGB output embeddings, which are then concatenated to form a  $2048 \times T$  embedding, where  $T$  is the total number of video segments. Each video segment refers to 16 frames sampled at 30 FPS with a stride of 4 frames. This is the standard feature extraction pipeline used in all previous TAL works [41, 58]. To deal with variable frame lengths  $T$ , we pad all samples to  $T = 2048$ , which accounts for the length of all videos. During training, we include a mask to represent the zero-padded regions and apply the mask after each operation.

**Table 7:** GPT generated descriptions for PLOT-TAL Verbose on THUMOS’14 Dataset.

ID	Description
7	The precise moment a baseball player winds up and releases the ball towards the batter
9	The instant a basketball player leaps into the air to forcefully slam the ball through the hoop
12	The exact moment the cue stick strikes the cue ball, initiating the billiards shot
21	The moment a weightlifter hoists the barbell from the ground to overhead in one fluid motion
22	The split second a diver leaps off the cliff edge, beginning their descent into the water below
23	The moment a cricket bowler releases the ball towards the batsman with a swift arm motion
24	The precise moment the batsman swings the bat to strike the cricket ball
26	The instant a diver jumps off the board, tucking and twisting before plunging into the pool
31	The moment a frisbee is caught by a leaping player, securing it firmly in their hands
33	The exact moment a golfer swings the club, making contact with the ball to send it flying
36	The moment an athlete spins and releases the hammer, propelling it into the air
40	The split second an athlete takes off over the high jump bar, attempting to clear it without touching
45	The precise moment the javelin is thrown, with the athlete’s arm extending forward in a powerful motion
51	The instant an athlete sprints and leaps into the air to cover the maximum distance before landing in the sand pit
68	The moment an athlete plants the pole in the box and vaults over the bar, pushing themselves upwards
79	The exact moment the shot is put from the neck, using one hand, in a pushing motion through the air
85	The moment a soccer player strikes the ball with their foot aiming to score a penalty kick
92	The precise moment a tennis player swings their racket to strike the incoming ball
93	The instant an athlete spins and releases the discus, hurling it into the designated sector
97	The moment a volleyball player jumps and forcefully spikes the ball over the net towards the opponent’s court

## 5.2 Training

We train each model for 100 epochs, except for when we increase the number of shots above 15, in which case we train for 200. We randomly initialize the *ctx* embedding vectors and append them to the start of the prompt. All models are trained with a batch size of 2 on a single NVIDIA RTX 3090 24GB GPU. The memory required for training the model on THUMOS'14 with a batch size of 2 and when  $N = 4$  is 5GB. We include a summary of the method in Algorithm 1.

---

### Algorithm 1 Overview of TAL-PLOT method

---

**Input:** Untrimmed input video  $\mathcal{V}$   
**Output:** Action instances  $\mathcal{Y} = \{y_1, y_2, \dots, y_N\}$

- 1: **Feature Extraction and Representation:**
- 2: **for**  $t = 1$  to  $T$  **do**
- 3:   Extract feature vector  $x_t = f_{\text{CNN}}(v_t)$  using a 3D CNN
- 4:   Refine features  $x'_t = f_{\text{conv}}(x_t)$  with a 1D convolutional layer
- 5: **end for**
- 6: **Adaptive Prompt Learning:**
- 7: **for** each action category  $k$  **do**
- 8:   Generate  $N$  prompts  $\mathcal{P}_k = \{P_{k1}, P_{k2}, \dots, P_{kN}\}$  using  $f_{\text{CLIP}}$
- 9: **end for**
- 10: **Optimal Transport with Sinkhorn Algorithm:**
- 11: **for** each action category  $k$  **do**
- 12:   Align features  $\{x'_1, \dots, x'_T\}$  with prompts  $\mathcal{P}_k$  using OT
- 13: **end for**
- 14: **Temporal Pyramid and Feature Integration:**
- 15: Construct temporal feature pyramid  $X'_l$  with max-pooling
- 16: **Multi-Resolution Temporal Alignment:**
- 17: **for**  $l = 1$  to  $L$  **do**
- 18:   Align features at level  $l$  of the pyramid with  $\mathcal{P}_k$
- 19: **end for**
- 20: **Decoder Architecture:**
- 21: Use aligned features to predict action labels  $\Psi$  and boundaries  $O_t$
- 22: **Learning Objective:**
- 23: Minimize total loss  $L_{\text{total}}$  with Focal Loss and DIoU Loss
- 24: **return**  $\mathcal{Y}$

---

## 5.3 Optimal Transport

As discussed in the main paper. The optimal transport is optimized in a two-stage process as proposed in [8] where we find the transport cost between the video features and prompts in the inner loop. After converging the Sinkhorn algorithm, we use the backward pass to update the learnable prompts. For the parameters, we follow the setup in [8] where  $\delta = 0.01$ ,  $\lambda = 0.1$ , and we perform 100 iterations

within the inner loop. We generate results over 4 random seeds and report the average. Further details are provided in Algorithm 2.

---

**Algorithm 2** Optimal Transport Sinkhorn Algorithm for Few-Shot TAL

---

**Input:** Untrimmed input video  $\mathcal{V}$ , pretrained model features  $f_{\text{CNN}}$ , number of prompts  $N$ , entropy parameter  $\lambda$ , maximum number of iterations  $T_{\text{in}}, T_{\text{out}}$

**Output:** Optimized prompt parameters  $\{\omega_n\}_{n=1}^N$

```

1: Initialize prompt parameters  $\{\omega_n\}_{n=1}^N$ 
2: for  $t_{\text{out}} = 1$  to  $T_{\text{out}}$  do
3:   Obtain a visual feature set  $F \in \mathbb{R}^{M \times C}$  with the visual encoder  $f_{\text{CNN}}(x_t)$ 
4:   Generate prompt feature set  $G_k \in \mathbb{R}^{N \times C}$  for each class with textual encoder
      $g(\text{label}_k, \text{ctx}_{k1}, \dots, \text{ctx}_{kn_{\text{ctx}}})$ 
5:   Calculate the cost matrix  $C_k = 1 - F^\top G_k$  for each class
6:   Calculate the OT distance with an inner loop:
7:   Initialize  $v^{(0)} = 1, \delta = 0.1, \Delta v = \infty$ 
8:   for  $t_{\text{in}} = 1$  to  $T_{\text{in}}$  do
9:     Update  $u^{(t_{\text{in}})} = u / (\exp(-C/\lambda) v^{(t_{\text{in}}-1)})$ 
10:    Update  $v^{(t_{\text{in}})} = v / (\exp(-C/\lambda)^\top u^{(t_{\text{in}})})$ 
11:    Update  $\Delta v = \sum |v^{(t_{\text{in}})} - v^{(t_{\text{in}}-1)}| / N$ 
12:    if  $\Delta v < \delta$  then
13:      Break
14:    end if
15:  end for
16:  Obtain optimal transport plan  $T_k^* = \text{diag}(u^{(t)}) \exp(-C_k/\lambda) \text{diag}(v^{(t)})$ 
17:  Calculate the OT distance  $d_{\text{OT}}(k) = \langle T_k^*, C_k \rangle$ 
18:  Calculate the classification probability  $p_{\text{OT}}(y = k|x)$  with the OT distance
19:  Update the parameters of prompts  $\{\omega_n\}_{n=1}^N$  with cross-entropy loss  $L_{\text{CE}}$ 
20: end for
21: return Optimized prompt parameters  $\{\omega_n\}_{n=1}^N$ 

```

---

## 6 Limitations

Several trade-offs must be considered when implementing optimal transport in this model architecture. Introducing  $N$  prompts for each class,  $k$  leads to more trainable parameters than single prompt training methods such as CoOP. The model’s training and inference time is also increased linearly with the number of prompts added. The effectiveness of verbose prompts and additional prompt engineering could yield improved results. Optimising these components is left to future work.

## 7 Conclusion

This paper demonstrates a novel application of Optimal Transport to few-shot temporal action localization (FS-TAL). Our results show the effectiveness of aligning multiple prompts during training over several temporal resolutions. This performance increase is due to the increase in views (via multiple prompts) and the discriminate alignment via optimal transport, thus improving performance over single prompt methods.

## References

1. Bai, Y., Wang, Y., Tong, Y., Yang, Y., Liu, Q., Liu, J.: Boundary content graph neural network for temporal action proposal generation. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVIII 16. pp. 121–137. Springer (2020) [3](#)
2. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1728–1738 (2021) [4](#)
3. Bonneel, N., Rabin, J., Peyré, G., Pfister, H.: Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision* **51**, 22–45 (2015) [5](#)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Advances in neural information processing systems* **33**, 1877–1901 (2020) [12](#)
5. Buch, S., Escorcia, V., Shen, C., Ghanem, B., Carlos Niebles, J.: Sst: Single-stream temporal action proposals. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 2911–2920 (2017) [3](#)
6. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6299–6308 (2017) [11](#)
7. Chang, S., Wang, P., Wang, F., Li, H., Feng, J.: Augmented transformer with adaptive graph for temporal action proposal generation. *arXiv preprint arXiv:2103.16024* (2021) [3](#)
8. Chen, G., Yao, W., Song, X., Li, X., Rao, Y., Zhang, K.: Prompt learning with optimal transport for vision-language models. *arXiv preprint arXiv:2210.01253* (2022) [2](#), [5](#), [9](#), [11](#), [12](#), [19](#)
9. Chen, G., Zheng, Y.D., Wang, L., Lu, T.: Dcan: improving temporal action detection via dual context aggregation. In: AAAI (2022) [3](#)
10. Chen, Y., Georgiou, T.T., Pavon, M.: Optimal transport in systems and control. *Annual Review of Control, Robotics, and Autonomous Systems* **4**, 89–113 (2021) [4](#)
11. Cheng, F., Bertasius, G.: Tallformer: Temporal action localization with long-memory transformer. *ECCV* (2022) [3](#), [11](#)
12. Cuturi, M.: Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems* **26** (2013) [4](#), [7](#), [8](#)
13. Cuturi, M., Doucet, A.: Fast computation of wasserstein barycenters. In: International conference on machine learning. pp. 685–693. PMLR (2014) [5](#)

14. Damen, D., Doughty, H., Farinella, G.M., Fidler, S., Furnari, A., Kazakos, E., Moltisanti, D., Munro, J., Perrett, T., Price, W., et al.: Scaling egocentric vision: The epic-kitchens dataset. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. pp. 720–736 (2018) [10](#), [11](#)
15. Deng, C., Chen, Q., Qin, P., Chen, D., Wu, Q.: Prompt switch: Efficient clip adaptation for text-video retrieval. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 15648–15658 (2023) [2](#), [4](#)
16. Deng, C., Wu, Q., Wu, Q., Hu, F., Lyu, F., Tan, M.: Visual grounding via accumulated attention. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) [4](#)
17. Escorcia, V., Caba Heilbron, F., Niebles, J.C., Ghanem, B.: 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. pp. 768–784. Springer (2016) [3](#)
18. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 6202–6211 (2019) [11](#)
19. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV* 16. pp. 214–229. Springer (2020) [4](#)
20. Galichon, A.: *Optimal transport methods in economics*. Princeton University Press (2018) [4](#)
21. Gong, G., Zheng, L., Mu, Y.: Scale matters: Temporal scale aggregation network for precise action localization in untrimmed videos. In: *2020 IEEE International Conference on Multimedia and Expo (ICME)*. pp. 1–6. IEEE (2020) [3](#)
22. Heilbron, F.C., Niebles, J.C., Ghanem, B.: 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*. pp. 1914–1923 (2016) [3](#)
23. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: Action recognition with a large number of classes (2014) [10](#), [11](#)
24. Ju, C., Han, T., Zheng, K., Zhang, Y., Xie, W.: Prompting visual-language models for efficient video understanding. In: *European Conference on Computer Vision*. pp. 105–124. Springer (2022) [4](#)
25. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950* (2017) [17](#)
26. Keisham, K., Jalali, A., Kim, J., Lee, M.: Multi-level alignment for few-shot temporal action localization. *Information Sciences* **650**, 119618 (2023) [14](#)
27. Li, D., Li, J., Li, H., Niebles, J.C., Hoi, S.C.: Align and prompt: Video-and-language pre-training with entity prompts. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4953–4963 (2022) [2](#), [4](#)
28. Lin, C., Li, J., Wang, Y., Tai, Y., Luo, D., Cui, Z., Wang, C., Li, J., Huang, F., Ji, R.: Fast learning of temporal action proposal via dense boundary generator. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 34, pp. 11499–11506 (2020) [3](#)
29. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y.: Learning salient boundary feature for anchor-free temporal action localization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3320–3329 (2021) [3](#)

30. Lin, T., Liu, X., Li, X., Ding, E., Wen, S.: Bmn: Boundary-matching network for temporal action proposal generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3889–3898 (2019) [3](#), [11](#)
31. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M.: Bsn: Boundary sensitive network for temporal action proposal generation. In: Proceedings of the European conference on computer vision (ECCV). pp. 3–19 (2018) [3](#)
32. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016) [3](#)
33. Liu, X., Hu, Y., Bai, S., Ding, F., Bai, X., Torr, P.H.: Multi-shot temporal event localization: a benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12596–12606 (2021) [3](#)
34. Nag, S., Xu, M., Zhu, X., Pérez-Rúa, J.M., Ghanem, B., Song, Y.Z., Xiang, T.: Multi-modal few-shot temporal action detection via vision-language meta-adaptation. arXiv preprint arXiv:2211.14905 (2022) [2](#), [4](#), [11](#), [14](#)
35. Nag, S., Zhu, X., Song, Y.Z., Xiang, T.: Post-processing temporal action detection. arXiv preprint arXiv:2211.14924 (2022) [4](#)
36. Nag, S., Zhu, X., Song, Y.Z., Xiang, T.: Zero-shot temporal action detection via vision-language prompting. In: European Conference on Computer Vision. pp. 681–697. Springer (2022) [4](#), [11](#)
37. Nag, S., Zhu, X., Xiang, T.: Few-shot temporal action localization with query adaptive transformer. arXiv preprint arXiv:2110.10552 (2021) [2](#), [11](#), [14](#)
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) [2](#), [11](#), [12](#)
39. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016) [3](#)
40. Shi, D., Cao, Q., Zhong, Y., An, S., Cheng, J., Zhu, H., Tao, D.: Temporal action localization with enhanced instant discriminability. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2023) [3](#)
41. Shi, D., Zhong, Y., Cao, Q., Ma, L., Li, J., Tao, D.: Tridet: Temporal action detection with relative boundary modeling. arXiv preprint arXiv:2303.07347 (2023) [7](#), [17](#)
42. Tan, J., Tang, J., Wang, L., Wu, G.: Relaxed transformer decoders for direct action proposal generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 13526–13535 (2021) [3](#)
43. Tang, T.N., Kim, K., Sohn, K.: Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization. arXiv preprint arXiv:2303.09055 (2023) [7](#), [8](#), [11](#)
44. Torres, L.C., Pereira, L.M., Amini, M.H.: A survey on optimal transport for machine learning: Theory and applications. arXiv preprint arXiv:2106.01963 (2021) [4](#)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017) [3](#)
46. Villani, C., et al.: Optimal transport: old and new, vol. 338. Springer (2009) [4](#)
47. Wang, L., Xiong, Y., Lin, D., Van Gool, L.: Untrimmednets for weakly supervised action recognition and detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017) [4](#), [11](#)

48. Wang, L., Yang, H., Wu, W., Yao, H., Huang, H.: Temporal action proposal generation with transformers. arXiv preprint arXiv:2105.12043 (2021) [3](#)
49. Weng, Y., Pan, Z., Han, M., Chang, X., Zhuang, B.: An efficient spatio-temporal pyramid transformer for action detection. In: ECCV (2022) [3](#), [4](#), [15](#)
50. Xu, H., Sun, X., Tzeng, E., Das, A., Saenko, K., Darrell, T.: Revisiting few-shot activity detection with class similarity control. arXiv preprint arXiv:2004.00137 (2020) [4](#)
51. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B.: G-tad: Sub-graph localization for temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10156–10165 (2020) [3](#), [11](#)
52. Yang, H., He, X., Porikli, F.: One-shot action localization by learning sequence matching network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1450–1459 (2018) [4](#)
53. Yang, L., Han, J., Zhao, T., Liu, N., Zhang, D.: Structured attention composition for temporal action localization. IEEE Transactions on Image Processing (2022) [5](#)
54. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J.: Revisiting anchor mechanisms for temporal action localization. IEEE Transactions on Image Processing **29**, 8535–8548 (2020) [3](#)
55. Yang, M., Chen, G., Zheng, Y.D., Lu, T., Wang, L.: BasicTad: an astounding rgb-only baseline for temporal action detection. arXiv preprint arXiv:2205.02717 (2022) [3](#)
56. Yang, P., Hu, V.T., Mettes, P., Snoek, C.G.: Localizing the common action among a few videos. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VII 16. pp. 505–521. Springer (2020) [14](#)
57. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C.: Graph convolutional networks for temporal action localization. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 7094–7103 (2019) [3](#), [11](#)
58. Zhang, C.L., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV. pp. 492–510. Springer (2022) [3](#), [7](#), [11](#), [17](#)
59. Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., Tian, Q.: Bottom-up temporal action localization with mutual regularization. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16. pp. 539–555. Springer (2020) [3](#)
60. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016) [2](#), [15](#)
61. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Conditional prompt learning for vision-language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16816–16825 (2022) [2](#), [4](#), [11](#), [15](#)
62. Zhou, K., Yang, J., Loy, C.C., Liu, Z.: Learning to prompt for vision-language models. International Journal of Computer Vision **130**(9), 2337–2348 (2022) [4](#), [14](#)
63. Zisserman, A., Carreira, J., Simonyan, K., Kay, W., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017) [11](#), [17](#)