Noname manuscript No.
(will be inserted by the editor)

Temporal Action Localization with Enhanced Instant Discriminability

Dingfeng Shi[1], Qiong Cao[2], Yujie Zhong[3], Shan An[2], Jian Cheng[1], Haogang Zhu[1,4], Dacheng Tao[5]

arXiv:2309.05590v1 [cs.CV] 11 Sep 2023

Received: date / Accepted: date

Abstract Temporal action detection (TAD) aims to detect all action boundaries and their corresponding categories in an untrimmed video. The unclear boundaries of actions in videos often result in imprecise predictions of action boundaries by existing methods. To resolve this issue, we propose a one-stage framework named TriDet. First, we propose a Trident-head to model the action boundary via an estimated relative probability distribution around the boundary. Then, we analyze the rank-loss problem (*i.e.* instant discriminability deterioration) in transformer-based methods and propose an efficient scalable-granularity perception (SGP) layer to mitigate this issue. To further push the limit of instant discriminability in the video backbone, we leverage the strong representation capability of pretrained large models and investigate their performance on TAD. Last, considering the adequate spatial-temporal context for classification, we design a decoupled feature pyramid network with separate feature pyramids to incorporate rich spatial context from the large model for localization. Experimental results demonstrate the robustness of TriDet and its state-of-the-art performance on multiple TAD datasets, including hierarchical (multilabel) TAD datasets.

Keywords Temporal action detection · Video understanding · Transformer · Convolutional network

1 Introduction

Temporal action detection (TAD) is a critical task in video understanding that consists of two subtasks: action recognition and action localization. In the deep learning era, the two mainstream methods, CNN-based methods and transformer-based methods, have made impressive progress in TAD. However, several unsolved problems in TAD make it a challenging task.

In object detection, the majority of objects have clear boundaries, such as outlines, which make them relatively easy to detect. However, the lack of clear boundaries is a significant issue in TAD. For instance, it can be challenging to pinpoint the exact frame that marks the boundary at the end of a long jump (Alwassel et al, 2018). This issue makes accurate localization in TAD challenging.

Existing methods address the problem from two main perspectives. Some studies (Lin et al, 2018, 2019; Zeng et al, 2019; Zhao et al, 2020; Long et al, 2019; Nag et al, 2022; Fu et al, 2023) aim to determine the boundaries of action by relying on global features, possibly missing detailed information at each instant. Meanwhile, other studies directly predict boundaries based on a single local feature (Zhang et al, 2022; Paul et al, 2018), potentially with some other features (Lin et al, 2021; Qing et al, 2021; Zhao et al, 2021), but they do not consider the relation between adjacent instants around the boundary.

To enhance localization learning, we posit that the relative response intensity of temporal features in a video can mitigate the impact of video feature complexity and increase localization accuracy. Based on this idea, we propose a one-stage action detector with a novel detection head called the Trident-head that is designed for action boundary localization. Instead of di-

∗ This work was conducted during an internship at JD.com.
✉ Corresponding authors.
✉ Qiong Cao (mathqiong2012@gmail.com),
✉ Haogang Zhu (haogangzhu@buaa.edu.cn),
✉ Jian Cheng (jian_cheng@buaa.edu.cn).
[1] Beihang University.
[2] JD.com.
[3] Meituran Inc.
[4] Zhongguancun Laboratory.
[5] University of Sydney.

rectly predicting the boundary offsets based on a single local feature, the proposed Trident-head models the action boundary via an estimated relative probability distribution of the boundary. The boundary offset is then computed based on the expected values of the neighboring bin set.

Furthermore, the transformer-based feature pyramid is utilized in several recent TAD methods (Zhang et al, 2022; Cheng and Bertasius, 2022; Weng et al, 2022), demonstrating encouraging outcomes. However, the video backbone's video features often exhibit significant similarities among snippets, which are further amplified by self-attention, leading to the rank-loss problem (*i.e.* discriminability deterioration) (Shi et al, 2023). Fortunately, the success of the previous transformer-based layers in TAD relies primarily on their macro-architecture, namely, how the normalization layer and feed-forward network (FFN) are connected, rather than the self-attention mechanism. We, therefore, propose an efficient convolutional-based layer, termed the scalable-granularity perception (SGP) layer, to alleviate the two abovementioned problems of self-attention. SGP comprises two primary branches, which serve to increase the discrimination of features in each instant and capture temporal information with different scales of receptive fields.

Additionally, most existing TAD methods utilize an action classification network (*e.g.* SlowFast(Feichtenhofer et al, 2019), I3D (Carreira and Zisserman, 2017) and TSN (Wang et al, 2018)) that is pretrained on a single dataset as a backbone. Therefore, the features extracted from those backbones are often not sufficiently distinct for boundary localization. To further push the limit of discriminability in the video backbone, we leverage the large image and video models and improve the localization accuracy for the TAD task. Namely, we utilize two types of pretrained large models: *temporal-level* (the simplified term for spatial-temporal-level) and *spatial-level* backbones, respectively.

The temporal-level backbone (*e.g.* VideoMAEv2 (Wang et al, 2023)) efficiently extracts a comprehensive representation within a specific temporal window, resulting in excellent detection performance. However, this representation is not precisely aligned with frame information, leading to potentially inaccurate localization. On the other hand, the visual context provided by the spatial-level backbone (e.g., DINOv2 (Oquab et al, 2023), MoCov2 (Chen et al, 2020b)), such as the appearance of specific objects (e.g., cigarettes in the smoking action), often determines the start and end of an action. Motivated by this, we propose a decoupled feature pyramid network (FPN) to fuse information from two backbone networks: VideoMAEv2 and DINOv2, which shows better results compared to other straightforward fusion methods. Concretely, we construct two separate feature pyramids based on the two backbones, namely, the temporal-level feature pyramid and the spatial-level feature pyramid. Then, the temporal-level feature pyramid is directly fed into the classification head, while both the temporal-level feature pyramid and spatial-level feature pyramid are combined through element-by-element summation along each level of the pyramid and fed into the localization head (*i.e.* Trident-head). By fully leveraging the benefits of both types of backbone networks, the decoupled FPN aids in enhancing localization.

Experimental results on several conventional TAD and multilabel TAD datasets show that TriDet is a state-of-the-art action detector, and extensive ablation experiments demonstrate its robustness.

The CVPR 2023 version (Shi et al, 2023) investigates the rank-loss issue of the transformer in TAD and introduces the SGP layer and Trident-head for more precise localization. The extensions of this work include the following: (1) we conduct experiments and analyze the application of temporal-level and spatial-level visually pretrained large models in TAD; (2) we propose a straightforward yet comprehensive feature pyramid network (FPN) that incorporates spatial-level context; (3) we build a comprehensive model that can adapt to multilabel TAD tasks and provide further results on two multilabel detection datasets MultiTHUMOS and Charades; (4) we conduct detailed experiments to analyze different variants of TriDet; (5) The code will be released to `https://github.com/dingfengshi/tridetplus`

2 Related Work

2.1 Temporal Action Detection

Temporal action detection (TAD) is a critical task for video understanding that aims to detect all action segments along with their boundary location and classification from untrimmed video (Lee et al, 2023, 2020). In the deep learning era, existing temporal action detection methods can be divided into two categories: two-stage methods and one-stage methods.

Two-stage methods (Xu et al, 2020; Zeng et al, 2019; Zhu et al, 2021; Sridhar et al, 2021; Qing et al, 2021; Nag et al, 2023) comprise two independent networks: a proposal generation network and a classification network. Most previous works (Lin et al, 2018, 2019, 2020; Chen et al, 2022; Escorcia et al, 2016; Liu et al, 2021) focus on the proposal generation phase. Specifically, some works (Lin et al, 2019, 2018; Chen et al, 2022) predict the probability of the action boundary and match the start and end instants densely based on the prediction score. Anchor-based methods (Lin et al, 2020; Escorcia et al, 2016) classify actions from specific anchor windows. However, two-stage methods are limited by high complexity and cannot be trained end-to-end.

One-stage methods perform localization and classification with a single network. Some previous works (Yang et al, 2020; Lin et al, 2021; Yang et al, 2022; Bhosale et al, 2023) build this hierarchical architecture based on a convolutional network (CNN). However, a performance gap remains between CNN-based and the latest TAD methods.

2.2 Object detection

Object detection is a twin task of TAD. General focal loss (Li et al, 2020) transforms bounding box regression from learning a Dirac delta distribution to a general distribution function. They utilize multiple bins to predict boundaries. However, these bins do not directly correspond to actual image information and necessitate an additional loss function to aid in convergence. Some methods (Howard et al, 2017; Chollet, 2017; Liu et al, 2022c) use depth-wise convolution to model the network structure, and some branched designs (Szegedy et al, 2017; Hu et al, 2018) show high generalization ability. These approaches are enlightening for the architecture design of TAD.

2.3 Transformer Based Method

Inspired by the great success of the transformer in the field of machine translation and object detection, some recent works (Zhang et al, 2022; Shi et al, 2022; Tan et al, 2021; Cheng and Bertasius, 2022; Liu et al, 2022b,a) adopt the attention mechanism in TAD to improve the detection performance. For example, some works(Tan et al, 2021; Shi et al, 2022; Liu et al, 2022b) detect action with the DETR-like transformer-based decoder (Carion et al, 2020), which models action segments as a set of learnable segments. Other works (Zhang et al, 2022; Cheng and Bertasius, 2022) extract a video representation with a transformer-based encoder.

However, most of these methods are based on *local* behavior. Namely, they conduct attention operations only in a local window, which introduces inductive bias similar to that of CNN but with larger computational complexity and additional limitations.

In addition, Dong et al. (Dong et al, 2021) analyze the pure self-attention operation that causes the token features to converge to a rank-1 matrix at a double exponential rate during initialization. This inspired us to analyze the issue in TAD from the perspective of feature rank. However, their analysis primarily focuses on the initialization phase and lacks an examination of specific applications and potential enhancements. We dive deeper into the rank-loss problem that emerges during the training of the TAD detector and propose an effective improvement scheme, the SGP layer, utilizing convolution.

2.4 Large-scale Pretrained Model

Recently, large-scale self-supervised pretrained models such as BERT (Devlin et al, 2018) and GPT-4 (OpenAI, 2023) have shown impressive results in the field of neural language processing (NLP). The large models trained using massive amounts of data have significantly improved various downstream tasks.

Motivated by the success seen in NLP, some researchers have begun to build visual foundation models by pretraining large-scale visual models (Bao et al, 2021; Kirillov et al, 2023; Chen et al, 2020a; Xie et al, 2022; Chen et al, 2020b), including image foundation models and video foundation models (Feichtenhofer et al, 2022; Lin et al, 2022; Wang et al, 2022). For the video foundation model, VideoMAEv2 (Wang et al, 2023) constructs a large model with billions of parameters to learn the temporal-visual representation of videos from multiple video datasets in a self-supervised manner. For the image foundation model, DINOv2 (Oquab et al, 2023) builds a large-scale curated image dataset for large ViT model training and distilling.

Although these large models have demonstrated strong performance in certain downstream tasks, their utilization and specific impact on TAD have not been thoroughly investigated. In this study, we aim to examine the impact of two types of backbones in TAD and explore methods to fully harness the potential of both backbone networks.

3 Method

3.1 Preliminaries

Problem definition. We first give a formal definition of the TAD task. Specifically, given a set of untrimmed videos $\mathcal{D} = \{\mathcal{V}_i\}_{i=1}^{n}$, we have a set of temporal visual features $X_i = \{x_t\}_{t=1}^{T_i}$ from each video $\mathcal{V}_i$, where T_i corresponds to the number of instants, and K_i segment labels $Y_i = \{s_k, e_k, c_k\}_{k=1}^{K_i}$ with the action segment start instant s_k, the end instant e_k and the corresponding action category c_k. TAD aims to detect all segments Y_i based on the input feature X_i.

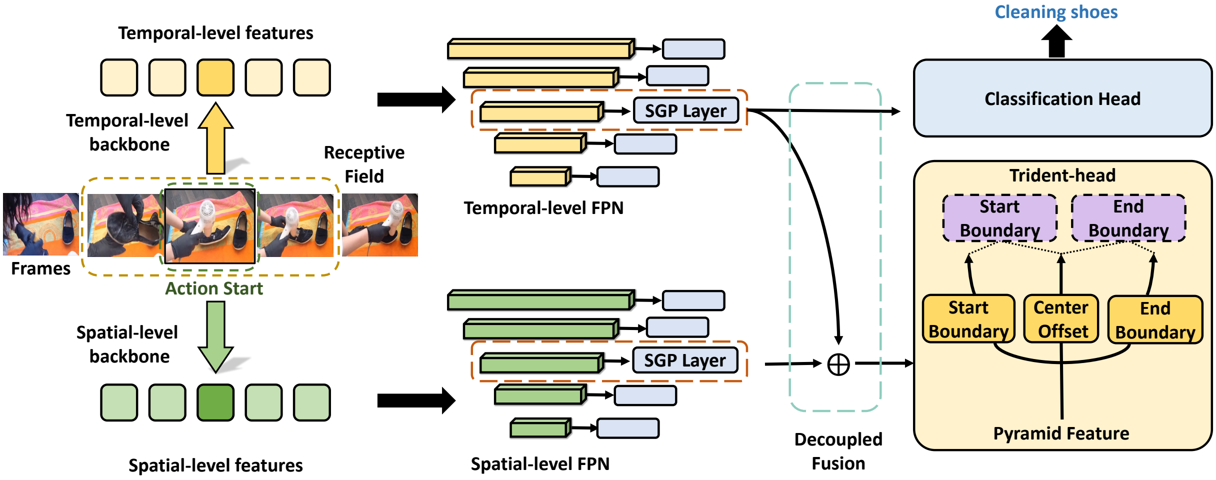


Fig. 1: Illustration of TriDet. We adopt temporal-level and spatial-level backbones to construct pyramid features with the scalable-granularity perception (SGP) layer. The temporal-level features at each level are fed into a shared-weight classification head to determine the action categories. Additionally, the Trident-head takes in a fusion of temporal-level and spatial-level features to obtain the offset for each instant, and estimates the boundary offset based on the relative distribution predicted by three branches: start boundary, end boundary and center offset.

3.2 Method Overview

Our goal is to build a simple and efficient one-stage temporal action detector. As shown in Fig. 1, the overall architecture of our detector consists of two branch feature backbones (temporal-level and/or spatial-level backbones). These backbones are accompanied by two corresponding feature pyramids, which undergo decoupled fusion and are fed into a classification head as well as a boundary-oriented Trident-head.

First, the video features are extracted using a pretrained temporal-level backbone (*e.g.* VideoMAEv2 (Wang et al, 2023) or SlowFast (Feichtenhofer et al, 2019)) or pretrained spatial-level backbone (*e.g.* DINOv2 (Oquab et al, 2023)). Next, the SGP feature pyramid for each backbone is built to tackle actions with various temporal lengths, similar to some recent TAD works (Lin et al, 2021; Zhang et al, 2022; Cheng and Bertasius, 2022). Namely, the two backbone features are iteratively downsampled, and each scale level is processed with a proposed SGP layer to enhance the interaction between features with different temporal receptive fields, resulting in the temporal-level feature pyramid and spatial-level feature pyramid. Then, the temporal-level feature pyramid is fed directly into the classification head, while both the temporal-level feature pyramid and spatial-level feature pyramid are combined through element-by- element summation along each level of the pyramid and fed into the Trident-head.

Last, action segments are detected by a designed boundary-oriented Trident-head. We elaborate on the proposed modules in the following.

3.3 Feature Pyramid with SGP Layer

The feature pyramid network (FPN). The feature pyramid is obtained by first downsampling the output features of the video backbone network several times via max-pooling (with a stride of 2). The features at each pyramid level are then processed using transformer-like layers (e.g. ActionFormer (Zhang et al, 2022)).

Current transformer-based methods for TAD tasks rely primarily on the macro-architecture of the transformer (See Section 4.5 for details), rather than the self-attention mechanisms. Specifically, SA encounters two main issues: the rank-loss problem across the temporal dimension and high computational overhead.

Limitation 1: the rank-loss problem. In (Dong et al, 2021), the authors examine the pure self-attention operation, which leads to the convergence of the token feature matrix to a rank-1 matrix at a double exponential rate during the initialization phase. This means that the feature sequences become more similar and less distinguishable with depth, which is referred to as the rank-loss problem. We have also observed the occurrence of rank loss during the training phase in the TAD task (see Section 4.6). The temporal feature sequences

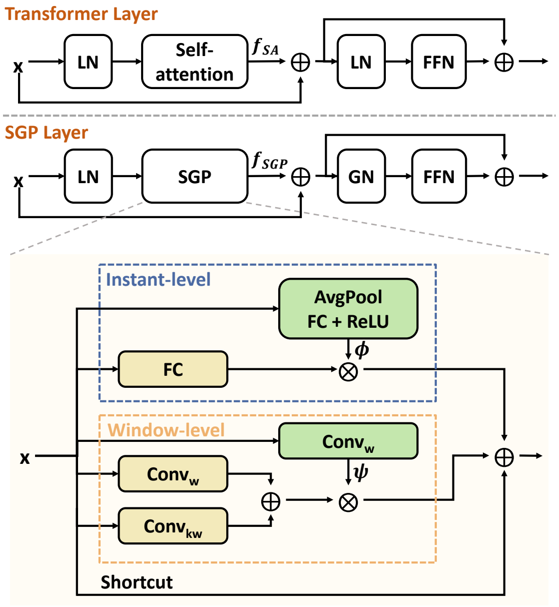


Fig. 2: Illustration of the structure of the SGP layer. We replace the self-attention and the second layer normalization (LN) with SGP and group normalization (GN), respectively.

extracted from the pretrained backbone network already show a high level of similarity, which makes it challenging to detect boundaries. However, the self-attention approach exacerbates this similarity issue and is detrimental to accurate localization.

We argue that the rank-loss problem in TAD arises because the probability matrix in self-attention (*i.e.* softmax(QK^T)) is *nonnegative* and *the sum of each row is 1*, indicating the outputs of SA are a *convex combination* of the value feature V. Considering that pure layer normalization (Ba et al, 2016) projects features onto the hypersphere in high-dimensional space, we analyze the degree of their distinguishability by studying the maximum angle between features within the instant features. We demonstrate that the maximum angle between features after the *convex combination* is less than or equal to that of the input features, resulting in increased similarity between features (as outlined in the supplementary material). A straightforward approach to mitigate this problem could be the replacement of the self-attention mechanism with a convolutional layer, given that it does not impose any constraints on the weights of its convolutional kernel. However, employing this approach leads to a performance drop (see SA-to-CNN in Section 4.5), which can be attributed to the limited fitting power of a single convolutional layer. Thus, it still necessitates innovative designs to enhance the performance of convolutional structures.

Limitation 2: high computational complexity. In addition, the dense pair-wise calculation (between instant features) in self-attention brings a high computational overhead and therefore decreases the inference speed.

Based on the above discovery, we propose an SGP layer to effectively capture multigranularity action information while suppressing the issues of rank loss and high computation complexity. The major difference between the transformer layer and the SGP layer is the replacement of the self-attention module with the fully convolutional module SGP. The successive layer normalization(Ba et al, 2016) (LN) is also changed to group normalization(Wu and He, 2018) (GN).

As shown in Fig. 2, SGP contains two main branches: an instant-level branch and a window-level branch. In the instant-level branch, we interact the features of each instant with the global average features in order to capture the overall video context. The window-level branch is designed to introduce the semantic content from a wider receptive field with a branch ψ to help dynamically focus on the features of each scale. Mathematically, given the temporal feature $X \in \mathcal{R}^{T\times D}$, the SGP can be written as:

$$f_{SGP} = \phi(X)FC(X) + \psi(X)(Conv_w(X) + Conv_{kw}(X)) + X, \tag{1}$$

where FC and $Conv_w$ denote a fully-connected layer for each instant and a 1-D depth-wise convolution layer (Chollet, 2017) over the temporal dimension with window size w, respectively. As a signature design of SGP, k is a scalable factor that captures a larger granularity of temporal information. The video-level average feature $\phi(X)$ and branch $\psi(X)$ are given as

$$\phi(X) = ReLU(FC(AvgPool(X))), \tag{2}$$
$$\psi(X) = Conv_w(X), \tag{3}$$

where $AvgPool(X) \in \mathcal{R}^{1\times D}$ denotes the average pooling operation applied to all features along the temporal dimension. Here, both $\phi(X)$ (which will be replicated to $\mathcal{R}^{T\times D}$) and $\psi(X) \in \mathcal{R}^{T\times D}$ perform element-wise multiplication with the mainstream feature.

The resultant SGP-based feature pyramid can achieve better performance than the transformer-based feature pyramid while being much more efficient.

3.4 Trident-head with Relative Boundary Modeling

Intrinsic property of action boundaries. Regarding the detection head, some existing methods directly regress the temporal length (Zhang et al, 2022) of the

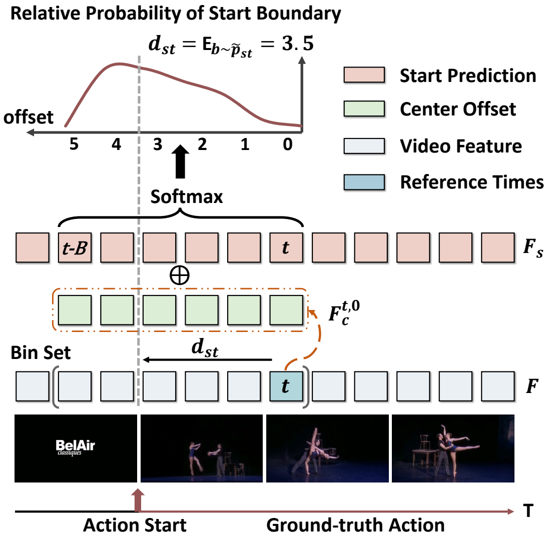


Fig. 3: The boundary localization mechanism of Trident-head. We predict the boundary response and the center offset for each instant. At instant t, the predicted boundary response in the neighboring bin set is summed element-wise with the center offset corresponding to instant t, which is further estimated as the relative boundary distribution. Finally, the offset is computed based on the expected value of the bin.

action at each instant of the feature and refine with the boundary feature(Lin et al, 2021; Qing et al, 2021), or (Lin et al, 2018, 2019; Zeng et al, 2019) simply predict an *actionness* score (indicating the probability of being an action). These simple strategies suffer from a problem in practice: imprecise boundary predictions, due to the intrinsic property of actions in videos. Namely, the boundaries of actions are typically not obvious, unlike the boundaries of objects in object detection. Intuitively, a statistical boundary localization method can reduce uncertainty and facilitate more precise boundaries.

Trident-head. In this work, we propose a boundary-oriented Trident-head to precisely locate action boundaries based on relative boundary modeling, *i.e.* considering the relation of features in a certain neighboring bin set and obtaining the relative probability of being a boundary for each instant in that set. The Trident-head consists of three components, namely, a start head, an end head, and a center-offset head, which are designed to locate the start boundary and end boundary and to capture the localization context from the center instant of the action, respectively. The Trident-head can be trained end-to-end with the detector.

Concretely, as shown in Fig. 3, given a sequence of features $F \in \mathcal{R}^{T\times D}$ output from the feature pyramid, we first obtain three feature sequences with the convolutional layer from the three branches, namely, start prediction $F_s \in \mathcal{R}^T$, end prediction $F_e \in \mathcal{R}^T$ and center offset $F_c \in \mathcal{R}^{T\times 2\times(B+1)}$, where B is the number of bins for boundary prediction, F_s and F_e characterize the response value for each instant as the starting or ending point of the action, respectively. The center offset F_c aims to estimate two conditional distributions $P(b_{st}|t)$ and $P(b_{et}|t)$, which represent the probability that each instant b_{st}/b_{et} (in start/end bin set) serves as a boundary when the instant t is the midpoint of an action. Then, we model the boundary distance by combining the outputs of the boundary head and center-offset head:

$$\widetilde{P}_{st} = Softmax(F_s^{[(t-B):t]} + F_c^{t,0}), \tag{4}$$

$$d_{st} = \mathbb{E}_{b\sim\widetilde{P}_{st}}[b] \approx \sum_{b=0}^{B}(b\widetilde{P}_{stb}), \tag{5}$$

where $F_s^{[(t-B):t]} \in \mathcal{R}^{B+1}$, $F_c^{t,0} \in \mathcal{R}^{B+1}$ are the response values in the start bin set of instant t and the center offsets predicted by instant t only, respectively, and $\widetilde{P}_{st}$ is the *relative probability*, which represents the probability of each instant being the start of an action within the bin set. Then, the distance between instant t and the start instant of action segment d_{st} is given by the expectation of the adjacent bin set. Similarly, the offset distance of the end boundary d_{et} can be obtained by

$$\widetilde{P}_{et} = Softmax(F_e^{[t:(t+B)]} + F_c^{t,1}), \tag{6}$$

$$d_{et} = \mathbb{E}_{b\sim\widetilde{P}_{et}}[b] \approx \sum_{b=0}^{B}(b\widetilde{P}_{etb}) \tag{7}$$

All heads are simply modeled in three layers of convolutional networks and share parameters at all feature pyramid levels to reduce the number of parameters.

Combination with feature pyramid. We apply the Trident-head in a predefined local bin set, which can be further improved by combining it with the feature pyramid. In this setting, features at each level of the feature pyramid share the same small number of bins B (*e.g.* 16), and the corresponding prediction for each level l can be scaled by 2^{l-1}, which can significantly help to stabilize the training process.

Formally, for an instant in the l-th feature level t^l, SGP estimates the boundary distance $\hat{d}^l_{st}$ and $\hat{d}^l_{et}$ with the Trident-head described above; then, the segments $a = (\hat{s}_t, \hat{e}_t)$ can be decoded by

$$\hat{s}_t = (t - \hat{d}^l_{st}) \times 2^{l-1}, \tag{8}$$

$$\hat{e}_t = (t + \hat{d}^l_{et}) \times 2^{l-1}. \tag{9}$$

Extension to multilabel task. It would be easy to adapt the Trident-head to a multilabel temporal action detection task by setting the outputs of the three branches to be category-dependent and assigning the positive sample positions and their corresponding regression value according to the category.

Concretely, the three branches in Trident-head are $F_s \in \mathcal{R}^{C\times T}$, $F_e \in \mathcal{R}^{C\times T}$ and $F_c \in \mathcal{R}^{C\times T\times 2\times(B+1)}$. The final offset distance for each instant $d \in \mathcal{R}^{C\times T\times 2}$ is calculated in the same way as before.

During the training phase, we employ central sampling to select classification and regression samples for each ground truth segment and allocate them to their respective instants and categories for regression. For classification, we adopt the multilabel binary objective and use focal loss (Lin et al, 2017) to optimize the classification head.

In the test, we select predicted segments based on the classification scores and use the corresponding predicted offset to decode the predicted segments.

Comparison with existing methods that have explicit boundary modeling. Many previous methods improve boundary predictions. We divide them into two broad categories: prediction based on sampling instants in segments (Lin et al, 2019; Liu et al, 2022b; Shi et al, 2022) and prediction based on a single instant. The first category predicts the boundary according to the global features of the predicted segments. These methods consider only global information instead of detailed information at each instant. The second category directly predicts the distance between an instant and its corresponding boundary based on the spatial-level feature (Lin et al, 2021; Zhang et al, 2022; Zhao et al, 2021; Qing et al, 2021). Some of these methods refine the segment with boundary features (Lin et al, 2021; Qing et al, 2021; Zhao et al, 2021).

However, they do not take the relation (*i.e.* relative probability of being a boundary) of adjacent instants into account. The proposed Trident-head differs from these two categories and shows superior performance in precise boundary localization.

3.5 Enhance TAD with Large-scale Pre-trained Models

3.5.1 Large-scale temporal-level backbone.

Traditional TAD tasks rely on temporal-level backbones to extract temporal feature sequences. However, the temporal features extracted from most of these backbones lack high distinguishability due to training from inadequate data. Fortunately, recent visual foundation models pretrained from massive amounts of data have shown impressive results in a variety of downstream tasks, inspiring us to utilize large models to enhance the performance in TAD. Therefore, we utilize VideoMAEv2 (Wang et al, 2023) as our temporal-level backbone. It is pretrained on the UnlabeledHybrid dataset and then fine-tuned on the K710 dataset.

The advantages and disadvantages. In Section 4.8.1, we analyze the advantages and disadvantages of VideoMAEv2 by comparing the detection results between VideoMAEv2 and SlowFast, a commonly used TAD backbone. Here, we present a simplified analysis.

VideoMAEv2 aims to learn a temporal-level representation from a short clip composed of multiple frames (*e.g.* 16), and the representation benefits from a large model and a large amount of training data (Wang et al, 2023). Therefore, VideoMAEv2 has a greater advantage in detecting short action segments, which contain only a small number of clips and do not require much more aggregation of temporal features.

However, VideoMAEv2 covers a short range in the temporal dimension during training (no overlap sliding with a kernel size of 2); therefore, it captures less long-range information than does SlowFast, which uses both fast and slow branches for temporal feature extraction. When applied to TAD, VideoMAEv2 lacks the ability to detect long action segments.

In addition, temporal-level backbone networks often encounter the issue of imprecise alignment between their features and the corresponding spatial information at each instant. Instead, the features serve as a generalized representation within a particular temporal window. Moreover, the boundaries of certain actions are determined by the frames in which specific objects appear, thereby resulting in imprecise localization. To enhance the precision of localization, we introduce a spatial-level backbone into the framework.

3.5.2 Large-scale spatial-level backbone.

The aim of the spatial-level backbone is to enhance the localization of action boundaries that exhibit a strong correlation with the frame context, which is ignored by the temporal-level backbone.

To incorporate frame context into the temporal feature sequence, we propose a straightforward yet comprehensive feature pyramid that involves spatial-level visual methods in the detection process, aiding in precise localization.

Concretely, we adopt the backbone of DINOv2 (Oquab et al, 2023) pretrained on the LVD-142M dataset and extract the output feature as the spatial representation for each instant. Then, we simply sample the spatial-level sequence with the same frame rate as the temporal-level sequence to match their sequence lengths. Next, we build feature pyramids with the SGP layer (w=1 for spatial-level backbone) and pooling for the temporal-level backbone and spatial-level backbone, respectively. Then, the temporal-level feature pyramid is directly fed into the classification head, while both the temporal-level feature pyramid and spatial-level feature pyramid are combined through element-by-element summation along each level of the pyramid and fed into the Trident-head.

Training and Inference Each layer l of the temporal-level feature pyramid or combined feature pyramid outputs a temporal feature $F^l \in \mathcal{R}^{(2^{l-1}T)\times D}$, which is then fed to the classification head or the Trident-head for action instance detection. The output of each instant t in feature pyramid layer l is denoted as $\hat{o}_t^l = (\hat{c}_t^l, \hat{d}_{st}^l, \hat{d}_{et}^l)$. The overall loss function is then defined as follows:

$$\begin{aligned}\mathcal{L} = &\frac{1}{N_{pos}} \sum_{l,t} \mathbb{1}_{\{c_t^l>0\}}(\sigma_{IoU}\mathcal{L}_{foc} + \mathcal{L}_{IoU}) \\ &+ \frac{1}{N_{neg}} \sum_{l,t} \mathbb{1}_{\{c_t^l=0\}}\mathcal{L}_{foc},\end{aligned} \tag{10}$$

where σ_{IoU} represents the temporal IoU between the predicted segment and the ground truth action instance and $\mathcal{L}_{foc}$ and $\mathcal{L}_{IoU}$ are the focal loss (Lin et al, 2017) and IoU loss (Rezatofighi et al, 2019), respectively. N_{pos} and N_{neg} denote the number of positive and negative samples and c_l^t is the classification label (0 for background). The term σ_{IoU} is used to reweight the classification loss at each instant, such that instants with better regression (*i.e.* of higher quality) contribute more to the training. Following previous methods (Tian et al, 2019; Zhang et al, 2020, 2022), center sampling is adopted to determine the positive samples. Namely, the instants around the center of an action instance are labeled as positive, and all others are considered negative.

Inference. At inference time, the instants with classification scores higher than threshold λ and their corresponding instances are kept. Last, Soft-NMS (Bodla et al, 2017) is applied for the deduplication of predicted instances.

4 Experiments

4.1 Datasets

We conduct experiments on six challenging datasets, including two TAD task datasets: THUMOS14 (Jiang et al, 2014) and HACS-Segment (Zhao et al, 2019) and a multilabel TAD task dataset MultiTHUMOS (Yeung et al, 2017).

TAD task datasets. THUMOS14 consists of 20 sport action classes and contains 200 and 213 untrimmed videos with 3,007 and 3,358 action segments in the training set and test set, respectively. HACS is a large-scale dataset that contains 200 classes of action, which has 37,613 videos for training, as well as 5,981 videos for test.

Multilabel TAD task datasets. MultiTHUMOS is a multilabel dataset that shares the same videos with the THUMOS14 dataset. It contains $38,690$ annotations of 65 action categories with an average of 1.5 labels per frame and 10.5 action classes per video.

Evaluation. For all these datasets, only the annotations of the training and validation sets are accessible. Following previous practice (Lin et al, 2019; Zhang et al, 2022; Cheng and Bertasius, 2022; Zeng et al, 2019), we evaluate on the validation set. We report the mean average precision (mAP) at different intersection over union (IoU) thresholds. For THUMOS14, we report IoU thresholds at [0.3:0.7:0.1]. For HACS, we report the result at the IoU threshold [0.5, 0.75, 0.95], and the average mAP is computed at [0.5:0.95:0.05]. For the multilabel datasets, we evaluate with detection-mAP instead of segmentation-mAP following the previous works (Tan et al, 2022; Tang et al, 2023; Shao et al, 2023). We report the average IoU with thresholds [0.1: 0.1: 0.9] for the MultiTHUMOS dataset.

4.2 Implementation Details

TriDet is trained end-to-end with the AdamW (Loshchilov and Hutter, 2019) optimizer. The initial learning rate is set to 10^{-4} for THUMOS14 and MultiTHUMOS, and 10^{-3} for HACS. We detach the gradient before the start boundary head and end boundary head and initialize the CNN weights of these two heads with a Gaussian distribution $\mathcal{N}(0, 0.1)$ to stabilize the training process. The learning rate is updated with the cosine annealing schedule (Loshchilov and Hutter, 2017). We train for 40, 48, and 13 epochs for THUMOS14, MultiTHUMOS and HACS (including a warmup of 20, 20 and 10 epochs).

For HACS, the number of bins B of the Trident-head is set to 14, the convolution window w is set to 11, and the scale factor k is set to 1.0. For THUMOS14, MultiTHUMOS the number of bins B of the Trident-head is set to 16, the convolution window w is set to 1, and the scale factor k is set to 1.5. We round the scaled window size and take it up to the nearest odd number for convenience.

Table 1: Comparison with the state-of-the-art methods on the HACS dataset.

Method	Venue	Backbone	0.5	0.75	0.95	Avg.
SSN (Zhao et al, 2017)	ICCV'2017	I3D	28.8	18.8	5.3	19.0
LoFi (Xu et al, 2021)	NeurIPS'2021	TSM	37.8	24.4	7.3	24.6
G-TAD (Xu et al, 2020)	CVPR'2020	I3D	41.1	27.6	8.3	27.5
TadTR (Liu et al, 2022b)	TIP'2022	I3D	47.1	32.1	10.9	32.1
BMN (Lin et al, 2019)	ICCV'2019	SlowFast	52.5	36.4	10.4	35.8
TALLFormer (Cheng and Bertasius, 2022)	ECCV'2022	Swin	55.0	36.1	11.8	36.5
TCANet (Qing et al, 2021)	CVPR'2021	SlowFast	54.1	37.2	11.3	36.8
TCANet + BMN	CVPR'2021	SlowFast	55.6	40.0	11.5	38.7
ETAD (Liu et al, 2023)	CVPRW'2023	SlowFast	55.7	39.0	13.8	38.8
TriDet	CVPR'2023	I3D	54.5	36.8	11.5	36.8
TriDet	CVPR'2023	SlowFast	56.7	39.3	11.7	38.6
TriDet	-	DINOv2	52.4	33.6	8.4	33.7
TriDet	-	VideoMAEv2	62.4	44.1	13.1	43.1
TriDet	-	Fused	**63.0**	**44.5**	**12.9**	**43.4**

4.3 Comparison with State-of-the-art Results

4.3.1 Single-label temporal action detection

HACS. In our experiment, we utilized the largest dataset, HACS, and adopted three commonly used temporal-level backbones I3D, SlowFast, and VideoMAEv2 and spatial-level backbone DINOv2 to ensure a fair and comprehensive comparison. As shown in Table 1, TriDet with VideoMAEv2 significantly outperforms the previous best method, with an average mAP margin of approximately 4.4%. This improvement is attributed to the fact that insufficient learning during training leads to inadequate discriminative temporal features, yet VideoMAEv2 effectively addresses this problem by extensively pretraining on a large dataset. This makes the learned features more distinguishable, which greatly benefits the TAD task.

In addition, through the fusion of the spatial-level backbone with DINOv2, the TriDet with fused FPN (denoted as TriDet-Fused) achieves a higher average mAP (43.4%) than does VideoMAEv2 alone (43.1%). This indicates that the spatial-level can further enhance the detection performance. We further analyze the results of the temporal-level backbone and spatial-level backbone in Section 4.8.2.

THUMOS14. We conduct the same comparison as on HACS. As Table 2 shows, TriDet also achieved state-of-the-art performance using the I3D backbone, with an average mAP of up to 69.3%, demonstrating the effectiveness of TriDet. VideoMAEv2 improves the average mAP by 0.7%, and TriDet-Fused further boosts the performance of TriDet, reaching an average mAP of 70.4%. These results are consistent with the results on HACS.

4.3.2 Multilabel temporal action detection

For the multilabel temporal action detection task, multiple levels of labels may exist at each instant. As Table 3 shows, our method achieves impressive performance on the multilabel task and significantly surpasses all previous methods on the MultiTHUMOS. In addition, we perform supplementary experiments on MultiTHUMOS to assess the influence of two-stream backbones (I3D with RGB and optical flow as input) and larger models (VideoMAEv2 + DINOv2). Clearly, incorporating both optical flow backbones and large model backbones results in noteworthy enhancement. Both optical flow and VideoMAEv2 capture the motion dynamics across consecutive frames, and the findings demonstrate that utilizing motion dynamics can effectively enhance the detection performance in the multilabel TAD task.

4.4 Ablation on the Main Components

We demonstrate the effectiveness of our proposed components in TriDet: SGP layer and Trident-head. To verify the effectiveness of our SGP layer, with the I3D backbone, we use a baseline feature pyramid used by (Lin et al, 2021; Zhang et al, 2022) to replace our SGP layer. The baseline consists of two 1D-convolutional layers and a shortcut. The window size of the convolutional layers is set to 3, and the number of channels of the intermediate features is set to the same dimension as the intermediate dimension in the FFN in our SGP layer. All other hyperparameters (*e.g.* number of the pyramid layers, etc.) are set the same as in our TriDet.

As depicted in Table 4, compared with the baseline model we implement (Row 1), the SGP layer brings a 6.2% absolute improvement in the average mAP. Sec-

Table 2: Comparison with state-of-the-art methods on THUMOS14 dataset. *: reported in VideoMAEv2 paper (Wang et al, 2023)

Method	Venue	Backbone	0.3	0.4	0.5	0.6	0.7	Avg.
BMN (Lin et al, 2019)	ICCV'2019	TSN	56.0	47.4	38.8	29.7	20.5	38.5
G-TAD (Xu et al, 2020)	CVPR'2020	TSN	54.5	47.6	40.3	30.8	23.4	39.3
A2Net (Yang et al, 2020)	TIP'2020	I3D	58.6	54.1	45.5	32.5	17.2	41.6
TCANet (Qing et al, 2021)	CVPR'2021	TSN	60.6	53.2	44.6	36.8	26.7	44.3
RTD-Net (Tan et al, 2021)	ICCV'2021	I3D	68.3	62.3	51.9	38.8	23.7	49.0
VSGN (Zhao et al, 2021)	ICCV'2021	TSN	66.7	60.4	52.4	41.0	30.4	50.2
ContextLoc (Zhu et al, 2021)	ICCV'2021	I3D	68.3	63.8	54.3	41.8	26.2	50.9
AFSD (Lin et al, 2021)	CVPR'2021	I3D	67.3	62.4	55.5	43.7	31.1	52.0
ReAct (Shi et al, 2022)	ECCV'2022	TSN	69.2	65.0	57.1	47.8	35.6	55.0
TadTR (Liu et al, 2022b)	TIP'2022	I3D	74.8	69.1	60.1	46.6	32.8	56.7
TALLFormer (Cheng and Bertasius, 2022)	ECCV'2022	Swin	76.0	-	63.2	-	34.5	59.2
ActionFormer (Zhang et al, 2022)	ECCV'2022	I3D	82.1	77.8	71.0	59.4	43.9	66.8
ActionFormer*	CVPR'2023	VideoMAEv2	84.0	79.6	73.0	63.5	47.7	69.6
TriDet	CVPR'2023	I3D	83.6	80.1	72.9	62.4	47.4	69.3
TriDet	-	DINOv2	67.9	62.2	53.2	41.8	27.7	50.6
TriDet	-	VideoMAEv2	84.8	80.0	73.3	**63.8**	48.8	70.1
TriDet	-	Fused	**85.5**	**80.7**	**73.9**	62.9	**48.9**	**70.4**

Table 3: Comparison with the state-of-the-art methods on the MultiTHUMOS datasets.

Method	Venue	Backbone	RGB	Flow	MultiTHUMOS			
					0.2	0.5	0.7	Avg
PointTAD (Tan et al, 2022)	NeurIPS'2022	I3D	√		39.7	24.9	12.0	23.5
ASL (Shao et al, 2023)	Arxiv'2023	I3D	√		42.4	27.8	13.7	25.5
TemporalMaxer (Tang et al, 2023)	Arxiv'2023	I3D	√		47.5	33.4	17.4	29.9
TriDet	-	I3D	√		49.1	34.3	17.8	30.7
TriDet	-	I3D	√	√	55.7	41.0	23.5	36.2
TriDet	-	VideoMAEv2	√		57.7	42.7	24.3	37.5
Tridet	-	Fused	√		57.6	**42.9**	**25.0**	**37.7**

Table 4: Analysis of the effectiveness of three main components on THUMOS14.

Method	SA	SGP	Trident	0.3	0.5	0.7	Avg.
1				77.3	65.2	40.0	62.1
2	√			82.1	71.0	43.9	66.8
3		√		**83.6**	71.7	45.8	68.3
4		√	√	**83.6**	**72.9**	**47.4**	**69.3**

ond, we compare the SGP with the previous state-of-the-art method, ActionFormer, which adopts the self-attention mechanism in a sliding window (Beltagy et al, 2020) with window size 7 (Row 2). Our SGP layer still achieves a 1.5% improvement in average mAP, demonstrating that the convolutional network has excellent performance in TAD. Furthermore, we compare our Trident-head with the normal spatial-level regression head, which regresses the boundary distance for each instant. Trident-head improves the average mAP by 1.0%, and the mAP improvement is more obvious in the case of high IoU threshold (*e.g.* 1.6% average mAP improvement in IoU 0.7%).

4.5 The Core Effectiveness of Transformer

As described in Section 1, we claim that the previous best transformer-based method (Zhang et al, 2022) primarily improved due to the macro-architecture of the transformer rather than the self-attention mechanism. To be self-contained, in this section, we further analyze the impact of module design on the detector.

From Transformer to CNN. As Fig. 4 shows, for comparison, we build two baseline models: a convolutional (CNN) baseline and a self-attention (SA) baseline. First, we build a CNN baseline in which the convolutional module is adopted from the previous one-stage detector (Lin et al, 2021; Zhang et al, 2022). Second, the previous state-of-the-art detector (Zhang et al, 2022) with local window self-attention (Beltagy et al, 2020) is chosen as the SA baseline. Then, to analyze the

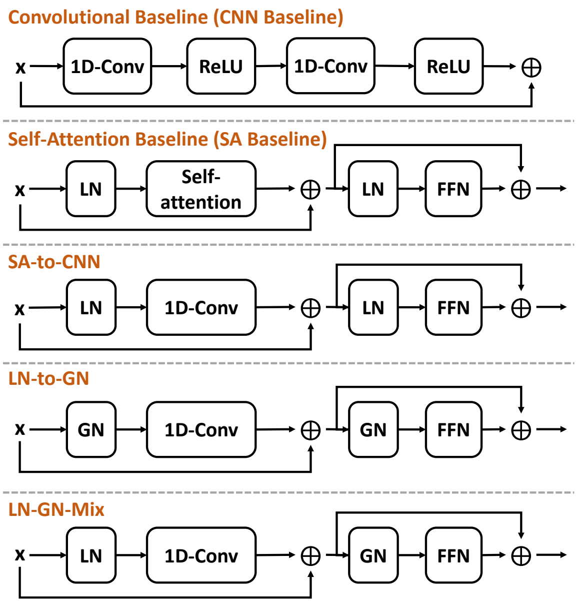


Fig. 4: Two baseline models and three different variants of the convolutional-based structure.

Table 5: The results of different variants on THUMOS14. *: with Trident-head.

Method	0.3	0.5	0.7	Avg.
CNN Baseline	77.3	65.2	40.0	62.1
SA Baseline	82.1	71.0	43.9	66.8
SA-to-CNN	80.4	67.5	42.9	64.9
LN-to-GN	80.0	68.0	42.3	64.8
LN-GN-Mix	80.8	68.8	43.6	65.7
SA-to-CNN*	81.2	68.7	43.5	65.7
LN-to-GN*	80.7	69.1	42.2	65.4
LN-GN-Mix*	81.6	69.5	42.9	66.0

importance of two common components: self-attention and normalization, in the Transformer (Vaswani et al, 2017) macro-architecture, we provide three variants of the convolutional-based structure: SA-to-CNN, LN-to-GN and LN-GN-Mix, and validate their performance on THUMOS14 with the I3D backbone.

To verify the robustness and sensitivity of Trident-head across different architectures of transformer, we conducted additional comparisons of the three aforementioned variants with and without the inclusion of Trident-head.

Results. From Table 5, we can see a large performance gap between the SA baseline and the CNN baseline (approximately 4.7% in average mAP), demonstrating that the transformer holds a large advantage for TAD tasks. Then, we conduct an ablation study with the three variants of the normal regression head.

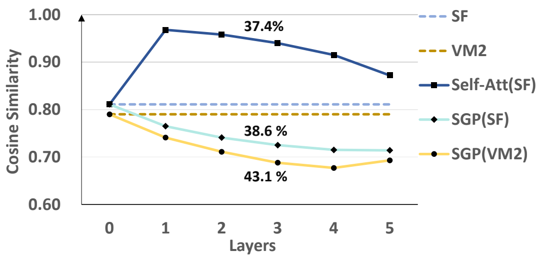


Fig. 5: The statistical average cosine similarity and average mAP for self-attention (Self-Att) and SGP layer with SlowFast (SF) and VideoMAEv2 (VM2) backbones.

We first simply replace the local self-attention with a 1D convolutional layer (SA-to-CNN) that has the same receptive field as that of (Zhang et al, 2022) (*e.g.* kernel size is 19). This change yields a dramatic performance increase in terms of the average mAP compared with the CNN baseline (approximately 2.8%) but is still behind the transformer baseline by approximately 1.9%. Next, we conduct experiments with different normalization layers (*i.e.* layer normalization (LN) (Ba et al, 2016) and group normalization (GN) (Wu and He, 2018) (*i.e.* LN-to-GN and LN-GN-Mix) and find that the hybrid structure of LN and GN (LN-GN-Mix) shows better performance compared to the SA baseline (65.7% versus 64.9%).

By combining with the Trident-head, the results consistently remain robust for all three variants (*i.e.* the performance relationship remains unchanged), demonstrating that the structural improvement remains reliable under different regression heads. Additionally, the LN-GN-Mix version achieves an average mAP of 66.0%, showcasing the potential for efficient convolutional modeling. Furthermore, convolution relaxes the restriction on the weight distributions at each moment (i.e., the sum of the weights is not necessarily 1), avoiding the rank-loss problem. Hence, these empirical results further motivate us to enhance the feature pyramid with the SGP layer.

4.6 Investigation of the Rank-Loss Problem

In this section, we validate the rank-loss problem in self-attention and the effectiveness of the SGP layer in preserving rank using the cosine similarity metric. The cosine similarity metric was chosen due to the fixed modulus property of the pure layer normalization.

Property 1 (Fixed modulus property) A pure Layer normalization $x' = LayerNorm(x)$ normalizes the temporal features $x \in \mathcal{R}^n$ to a fixed modulus $\sqrt{n-1}$

Proof Consider the data in each dimension x'_i:

$$\begin{aligned} x'_i =& \frac{x_i - mean(x)}{std(x)} \\ =& \frac{x_i - mean(x)}{\sqrt{\frac{1}{n-1}\sum_{i\in n}(x_i - mean(x))^2}}, \end{aligned} \tag{11}$$

then we have

$$\begin{aligned} ||x'||_2 =& \sqrt{\sum_{i\in n} {x'}_i^2} \\ =& \sqrt{\frac{\sum_{i\in n}(x_i - mean(x))^2}{\frac{1}{n-1}\sum_{i\in n}(x_i - mean(x))^2}} \\ =& \sqrt{n-1}. \end{aligned} \tag{12}$$

To simplify, we use cosine similarity S_c to measure the angular similarity between features at each instant and the video-level average feature:

$$S_c = \frac{1}{T}\sum_{i\in T} cos(x_i, \bar{x}) \tag{13}$$

where $\bar{x} = \frac{1}{T}\sum_{i\in T} x_i$.

We assess the average cosine similarity for the self-attention and the SGP layer with SlowFast and VideoMAEv2 backbones on the HACS dataset and present the results in Fig. 5.

First, self-attention increases the similarity of the temporal feature sequence for each layer output, surpassing even the similarity of the temporal feature sequence directly predicted by the backbone. This suggests that TAD faces the rank-loss problem due to self-attention. Conversely, our SGP layer mitigates this issue and demonstrates more discriminative power (38.6% vs 37.4% in average mAP). Furthermore, the SGP layer presents similar characteristics across various backbone networks (*i.e.* SlowFast and VideoMAEv2), thereby highlighting its robustness.

Second, the VideoMAEv2 backbone generates features with lower cosine similarity compared to the SlowFast backbone. This suggests that visual models pretrained using abundant video data produce more distinct features in the temporal dimension.

4.7 The Policies of Backbone Fusion

To explore an effective way to fuse features from temporal-level and spatial-level backbones, we conducted experiments using VideoMAEv2 and DINOv2 on THUMOS14, testing various fusion policies and fusion positions, and reporting their average mAP.

Table 6: The effectiveness of different fusion policies and fusion position.

	Policy	mAP		Policy	mAP
Fusion policy (early fusion)	add	69.7	Fusion position (add)	early fusion	69.7
	concat	69.9		before SGP	69.9
	cross-atten-T	69.2		after SGP	70.6
	cross-atten-S	52.2		FPN decouple	**71.2**
	conv-atten	**70.3**			

To recap, we decouple the FPN with the aim of improving localization precision and posit that spatial-level context is more beneficial for localization, while the classification of actions relies more on temporal-level context (motion information). To demonstrate the effectiveness of our decoupled fusion method, we conducted tests on the detection performance of multiple commonly used fusion methods, including both decoupled and coupled ones. By means of these tests, we illustrate the impact of these methods.

As shown in the left half of Table 6, for the fusion policies, we report five typical methods and conduct the fusion after the embedding network (a convolutional layer is employed to embed the output feature from two backbone networks into the same dimension, resulting in *early fusion*): (1) element-wise addition (*add*); (2) concatenation (*concat*); (3) local cross-attention (window size=5) with the temporal-level feature as value and spatial-level feature as query (*cross-atten-T*); (4) local cross-attention (window size=5) with the spatial-level feature as value and temporal-level feature as query (*cross-atten-S*); (5) predict an element-wise attention score with a convolutional layer from the concatenated features and weighted-sum of the two features (*conv-atten*).

早期融合

The *conv-atten* policy outperforms other methods, but this method fails to fully utilize the spatial-level backbone features. One possible reason is that early fusion reduces the effectiveness of the spatial-level context, as it gets destroyed by convolution after being fed into the temporal-level feature pyramid. Therefore, we further study where fusion can achieve the best result.

早期融合降低了空间级上下文的有效性，因为在送入到时间级特征金字塔后，它会被卷积破坏。因此进一步研究融合在哪里可以达到更好的效果

Specifically, three positions are considered: (1) after the embedding network (*early fusion*, which was mentioned in the previous paragraph; (2) before the SGP layer (*before SGP*), we first downsample the spatial-level feature using max-pooling and then perform element-wise summation separately for the temporal-level feature and the spatial-level feature in each pyramid level; and (3) after the SGP layer (*after SGP*), where the spatial-level features constructed by max-pooling are combined with the temporal features extracted by the SGP layer.

Upon analyzing Table 6 (right part), it was found that delaying fusion leads to a higher average mAP. This suggests that late fusion better preserves spatial-level context, thereby improving detection head prediction.

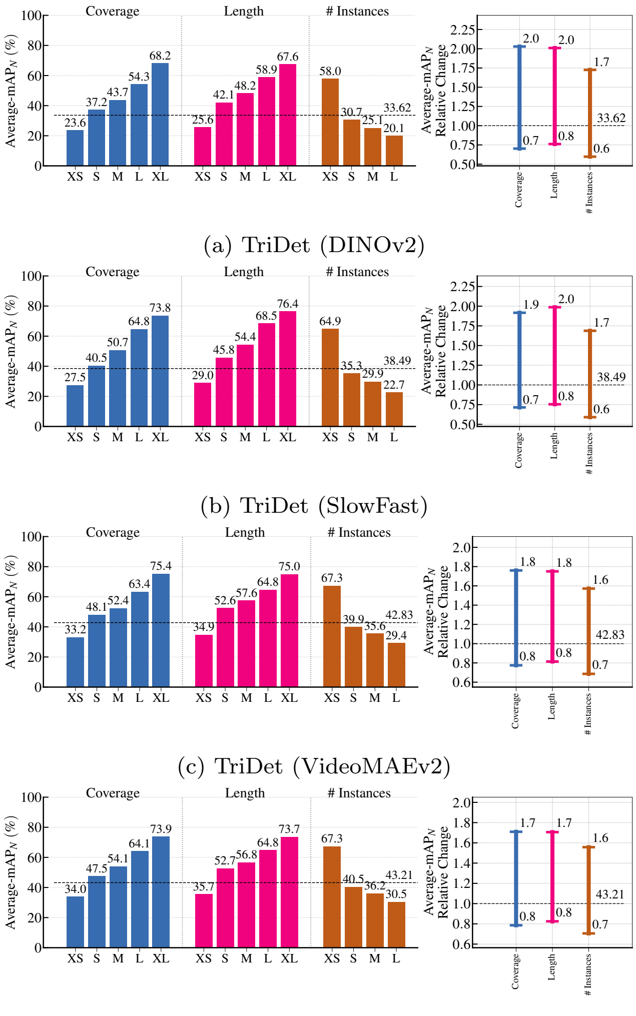
Based on this result, we further improve the *after SGP* policy by decoupling the fusion process after the SGP layer, as detailed in Section 3.5.2. Decoupling the fusion improves the results significantly compared to those of other fusion methods (71.2% vs 70.6%). This demonstrates that separating the fusion process of the spatial-level and temporal-level backbone can greatly benefit the localization process, without causing any interference with the classification process.

4.8 Error Analysis for Different Backbones

In this section, we utilize the tool provided by (Alwassel et al, 2018) to analyze the detection results for the four types of backbones on HACS, including DINOv2, SlowFast, VideoMAEv2 and the fusion of VideoMAEv2 and DINOv2. Specifically, we analyze the false positives, false negatives, and sensitivity.

The false positive analysis counts the percentage of five types of detection error in different Top-KG predictions, where G is the number of groundtruth segments. The five types of detection error are (1) background error (*i.e.* background segments are predicted as action instances), (2) localization error (*i.e.* an instance that correctly predicts the label and $0.1 \leq \sigma_{IoU} < \alpha$, where α is the preset threshold), (3) double detection error (*i.e.* accurate but repeatedly predicted instances), (4) confusion error (*i.e.* an instance that incorrectly predicts the label and $0.1 \leq \sigma_{IoU} < \alpha$) and (5) wrong label error (*i.e.* an instance that incorrectly predicts the label but $\sigma_{IoU} \geq \alpha$). In addition, the removing error impact is the improvement gained from removing the error predictions of different types.

The false negative analysis and sensitivity analysis are conducted for varying characteristics. Here, characteristics include coverage (*i.e.* normalized length), length (*i.e.* absolute length), and number of instances. The tool divides the test data into groups (denoted as XS, S, M, L, XL) based on *instances of different lengths* and *videos with different numbers of instances* and analyzes the results of each group individually. For average, the five groups are (0,0.2], (0.2,0.4], (0.4,0.6], (0.6,0.8], (0.8,1]. For length, the five groups are (0,30], (30,60], (60,120], (120,180], (180,+ inf) in seconds. For the number of instances, the four groups are (0,1], (1, 4], (4,8], (8,+inf).



(a) TriDet (DINOv2)

(b) TriDet (SlowFast)

(c) TriDet (VideoMAEv2)

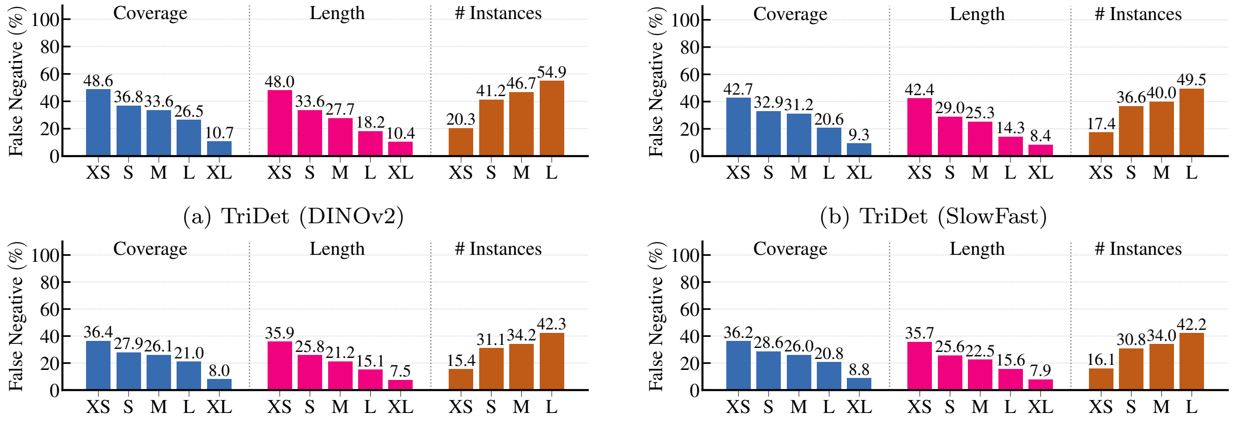
(d) TriDet-Fused

Fig. 6: Sensitivity analysis of the detection results, where mAP_N is the normalized mAP with the average number N of ground truth segments per class (Alwassel et al, 2018).

4.8.1 Comparison of VideoMAEv2 and SlowFast

In this section, we conduct experiments to analyse the pros and cons for VideoMAEv2 in Section 3.5.1 comparing with SlowFast. SlowFast is a powerful backbone that contains slow and fast paths to capture context at different scales, and is trained on the Kinect dataset. VideoMAEv2 is a temporal-level backbone that captures temporal context with a short scale (window size = 16), and is trained on a large amount of unlabeled video data and fine-tuned on the Kinect dataset.

In Fig. 6(b) and Fig. 6(c), we observe that VideoMAEv2 has a significant improvement in accuracy for short action segments (XS, S, M in the pink bar). However, in the case of long action segments (L and XL



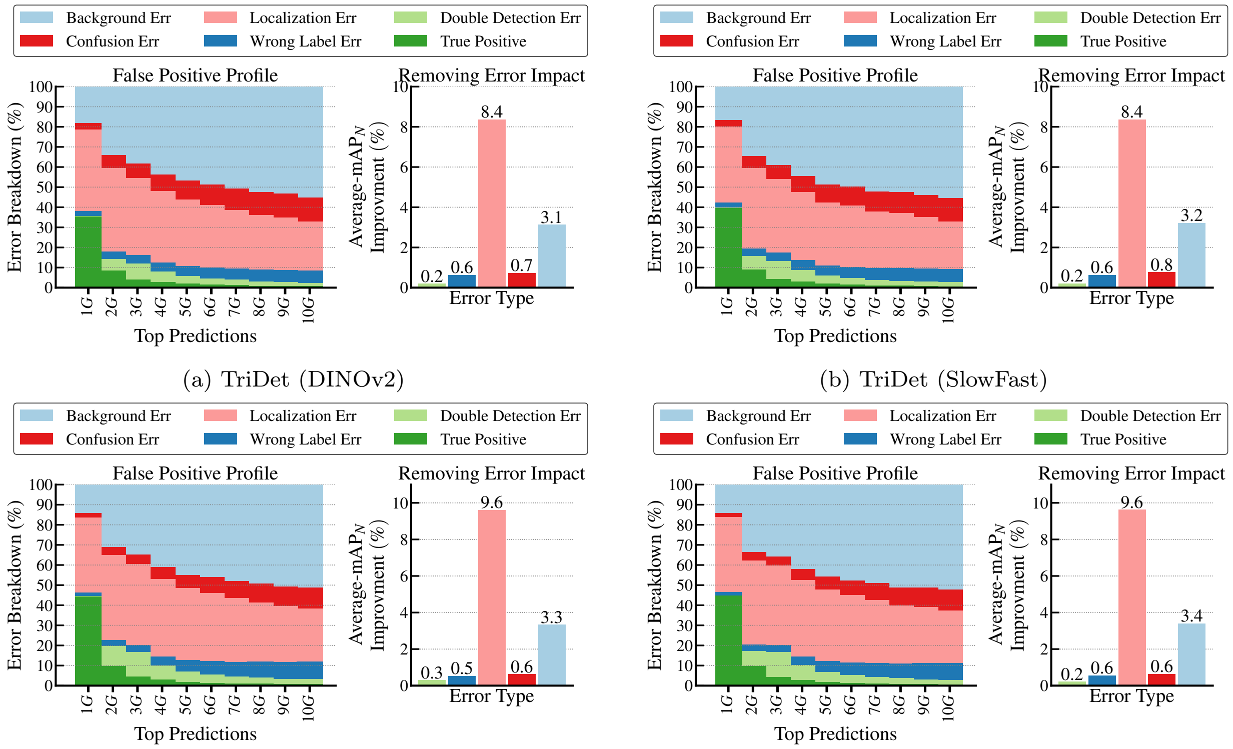
(a) TriDet (DINOv2)

(b) TriDet (SlowFast)

(c) TriDet (VideoMAEv2)

(d) TriDet-Fused

Fig. 7: The false negative analysis of the detection results, which counts the percentage of several common types of detection error in different Top-KG prediction groups, where G is the number of groundtruth segments.



(a) TriDet (DINOv2)

(b) TriDet (SlowFast)

(c) TriDet (VideoMAEv2)

(d) TriDet-Fused

Fig. 8: False positive analysis of the detection results, which counts the percentage of several common types of detection error in different Top-KG prediction groups, where G is the number of groundtruth segments.

in the pink bar), SlowFast outperforms VideoMAEv2. It suggests that the long temporal context captured by the SlowFast is beneficial for detecting long action segments. On the other hand, since HACS contains a larger proportion of short instances, VideoMAEv2 achieves better overall results.

In Fig. 7(b) and Fig. 7(c), the false negative rate of VideoMAEv2 is lower than SlowFast in all characteristics. However, in Fig. 8(b) and Fig. 8(c), we also observe an increase in the removing error impact of *localization error* and *background error* for VideoMAEv2 compared to SlowFast. The results above indicate that VideoMAEv2 enhances the overall mAP by increasing the detection rate of action segments. However, it also results in more redundant and imprecise predictions.

4.8.2 Comparison for DINOv2 and VideoMAEv2

In this section, we investigate the effectiveness and differences of temporal-level and spatial-level pre-trained backbones with VideoMAEv2 and DINOv2 in TAD. DINOv2 and VideoMAEv2 have been pre-trained on large-scale image and video datasets, respectively.

In Fig. 6(a) and Fig. 6(c), we can see the only DINOv2 backbone can still achieve promising results on the HACS dataset (average mAP 33.6%). However, the temporal-level context of VideoMAEv2 outperforms the spatial-level context of DINOv2 for action segments with different lengths. This aligns with our expectations, as the single spatial-level context lacks motion information and struggles to fully represent the action.

Additionally, in Fig. 8(a) and Fig. 8(c), we analyze the false positives for the two models. We can observe that the most significant types of false positive segments for the two models are *localization error*. However, the impact of *localization error* is higher for VideoMAEv2 compared to DINOv2, indicating that VideoMAEv2 has a greater potential to benefit from more accurate localization, despite having better overall performance (43.1% vs 33.7%). The results show that VideoMAEv2 alone still has room for improvement in localization, and there is a need to use spatial-level information to help improve it further.

4.8.3 Comparison for VideoMAEv2 and the fused model of DINOv2 and VideoMAEv2

In this section, we will analyze the effectiveness of the fused model (TriDet-Fused) by comparing it with the VideoMAEv2-only model.

In Fig. 6(c) and Fig. 6(d), we can observe that the average mAP for videos with more than one segment (*i.e.* the S, M, L items in the brown bar) increases. The accuracy of predicting short-length segments (*i.e.* XS, S in the pink bar) has also improved, though there is slightly less accuracy noticed for extremely long segments (*i.e.* XL in the pink bar).

Moreover, in Fig. 7(c) and Fig. 7(d), we can observe that the segments missed by TriDet using only VideoMAEv2 are mainly small ones (*i.e.* XS, S in the pink bar). However, this problem can be mitigated by combining VideoMAEv2 with the DINOv2, as demonstrated by the decrease in the false negative rates(*i.e.* 35.9% vs 35.7% and 25.8% vs 25.6%, for XS and S respectively).

The results indicate that the DINOv2 improves TriDet's ability to detect multiple short segments from videos. However, for extremely long videos, the detector struggles to incorporate too much spatial-level context and still relies on the temporal-level backbone. This is because DINOv2 is trained on the image datasets, which makes it difficult to capture the motion dynamics in long videos. The results suggest that there is still room for improvement in the method.

In addition, in Fig. 8(c) and Fig. 8(d), after removing the imprecise localization prediction (*i.e.* localization error), the average mAP improves by 9.6%, indicating that even though TriDet achieved state-of-the-art performance, inaccurate localization is still the main problem. Moreover, background error is also an important issue. Thus, how to suppress meaningless predictions remains an open question. The improvement in removing error impact (*e.g.* backgroud error, wrong label error) also suggests that the introduction of DINOv2 pushes the upper limits of TriDet's capabilities.

4.9 Qualitative Analysis

In Fig. 9, we present the visualization of the detection results on the HACS validation set.Clearly, TriDet can accurately predict the start and end instants of the action segments. Moreover, for action segments that involve specific objects such as cigarettes and shoe brushes, TriDet-Fused shows better performance, indicating that decoupled fusion can enhance the accuracy of boundary prediction even further.

5 Conclusion

In this paper, we present TriDet, a one-stage convolutional-based framework. To enhance localization learning, we propose a Trident-head to model the action boundary using an estimated relative probability distribution around the boundary. We also address the rank-loss problem commonly found in transformer-based methods by introducing an efficient SGP layer. Additionally, we leverage pretrained large models to improve the discriminability in the video backbone and present a decoupled FPN that can further boost the detection accuracy. We evaluate our method on THUMOS14, HACS, Multi-THUMOS, and Charades datasets, showcasing its high generalization capability and state-of-the-art performance. We conduct extensive ablation studies to validate the effectiveness of our approach.

Acknowledgements This work was supported by the National Key Research and Development Program of China under 2021YFB2700304, the National Natural Science Foundation of China under Grant No.61971017, No.U21A20523 and No. L222152, the National Key Research and Development Program of China under 2021ZD0140407 and the Spe-

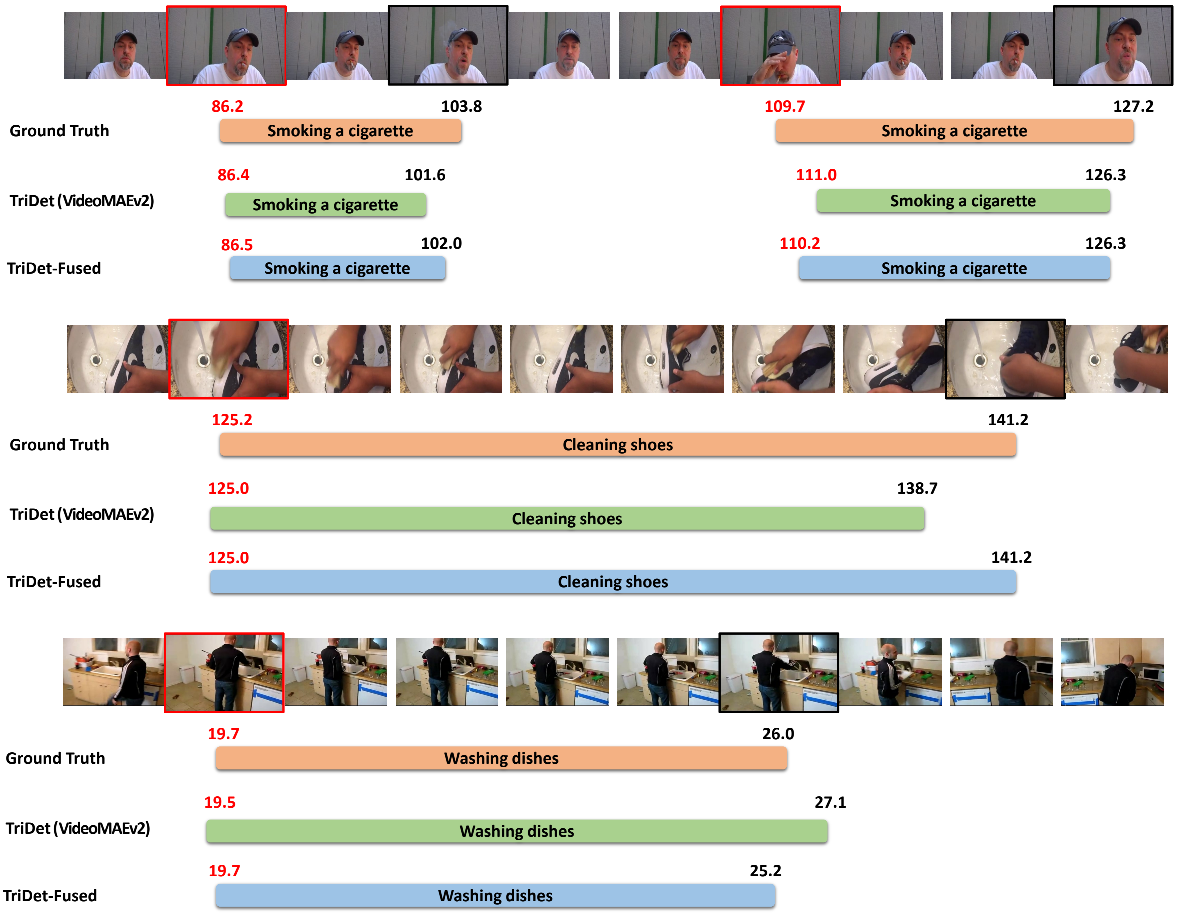


Fig. 9: Visualization results of the HACS validation set. The start and end timestamps (in seconds) of the actions are highlighted in red and black text, respectively.

cial Project for Innovation in Next-Generation Electronic Information Technology under Grant No.20310105D.

A The rank-loss problem in Transformer.

In (Dong et al, 2021), the authors discuss how the pure self-attention operation causes the input feature to converge to a rank-1 matrix at a double exponential rate, while MLP and residual connections can only partially slow this convergence. We have observed this phenomenon not only during initialization but also during training, which is disastrous for TAD tasks. This is because the video feature sequences extracted by pretrained action recognition networks are often highly similar (see Fig. 5), which further aggravates the rank-loss problem and makes the features at each instant indistinguishable, resulting in inaccurate detection of action.

We posit that the core reason for this issue lies in the softmax function used in self-attention. Namely, the probability matrix ($i.e.$ softmax(QK^T)) is *nonnegative* and *the sum of each row is 1*, indicating the outputs of SA are *convex combination* for the value feature V. We demonstrate that the largest angle between any two features in $V' = SA(V)$ is always less than or equal to the largest angle between features in V.

Definition 1 (Convex Combination) Given a set of points $S = \{x_1, x_2..., x_n\}$, a convex combination is a point of the form $\sum_n a_n x_n$, where $a_n \geq 0$ and $\sum_n a_n = 1$.

Definition 2 (Convex Hull) The convex hull H of a given set of points S is identical to the set of all their convex combinations. A convex hull is a convex set.

Property 2 (Extreme point) An extreme point p is a point in the set that does not lie on any open line segment between any other two points of the same set. For a point set S and its convex hull H, we have $p \in S$.

Lemma 1 *Consider the case of a convex hull that does not contain the origin. Let $a, b \in \mathbb{R}^n$ and let S be the convex hull formed by them. Then, the angle between any two position vectors of points in S is less than or equal to the angle between the position vectors of the extreme points $\vec{a}$ and $\vec{b}$.*

Proof Consider the objective function

$$f(x) = \cos(\vec{x}, \vec{y}) = \frac{\langle \vec{x}, \vec{y} \rangle}{\|\vec{x}\|_2 \, \|\vec{y}\|_2}, \tag{14}$$

where $\vec{x}, \vec{y}$ are the position vectors of two points x_1, x_2 within the convex hull S (a line segment with extreme points a and b). The angle between two vectors is invariant with respect to the magnitude of the vectors; thus, for simplicity, we define $\vec{x} = \vec{a} + x\vec{b}$, $\vec{y} = \vec{a} + y\vec{b}$, where $x, y \in [0, +\infty)$. Moreover, we have

$$\begin{aligned} f'(x) = \|\vec{x}\|_2^{-3} \, \|\vec{y}\|_2^{-1} \times \\ [\langle \vec{b}, \vec{y} \rangle ||\vec{a} + x\vec{b}||_2^2 - (||\vec{b}||_2^2 x + \langle \vec{a}, \vec{b} \rangle) \langle \vec{a} + x\vec{b}, \vec{y} \rangle] \end{aligned} \tag{15}$$

We consider

$$\begin{aligned} g(x) =& \langle \vec{b}, \vec{y} \rangle ||\vec{a} + x\vec{b}||_2^2 - (||\vec{b}||_2^2 x + \langle \vec{a}, \vec{b} \rangle) \langle \vec{a} + x\vec{b}, \vec{y} \rangle \\ =& \langle \vec{b}, \vec{y} \rangle (||\vec{a}||_2^2 + 2\langle \vec{a}, \vec{b} \rangle x + ||\vec{b}||_2^2 x^2) - [\langle \vec{b}, \vec{y} \rangle ||b||_2^2 x^2 \\ &+ (\langle \vec{a}, \vec{b} \rangle ||b||_2^2 + \langle \vec{a}, \vec{b} \rangle \langle \vec{b}, \vec{y} \rangle) x + \langle \vec{a}, \vec{y} \rangle \langle \vec{a}, \vec{b} \rangle] \\ =& (\langle \vec{a}, \vec{b} \rangle \langle \vec{b}, \vec{y} \rangle - \langle \vec{a}, \vec{y} \rangle \langle \vec{b}, \vec{b} \rangle) x + \langle \vec{a}, \vec{a} \rangle \langle \vec{b}, \vec{y} \rangle - \langle \vec{a}, \vec{y} \rangle \langle \vec{a}, \vec{b} \rangle. \end{aligned} \tag{16}$$

Substituting $\vec{y} = \vec{a} + y\vec{b}$ into the above equation, we have

$$\begin{aligned} g(x) =& (\langle \vec{a}, \vec{b} \rangle \langle \vec{b}, \vec{a} + y\vec{b} \rangle - \langle \vec{a}, \vec{a} + y\vec{b} \rangle \langle \vec{b}, \vec{b} \rangle) x + \\ & \langle \vec{a}, \vec{a} \rangle \langle \vec{b}, \vec{a} + y\vec{b} \rangle - \langle \vec{a}, \vec{a} + y\vec{b} \rangle \langle \vec{a}, \vec{b} \rangle \\ =& [\langle \vec{a}, \vec{b} \rangle (\langle \vec{a}, \vec{b} \rangle + y \langle \vec{b}, \vec{b} \rangle) - (\langle \vec{a}, \vec{a} \rangle + y \langle \vec{a}, \vec{b} \rangle) \langle \vec{b}, \vec{b} \rangle] x + \\ & [\langle \vec{a}, \vec{a} \rangle (\langle \vec{a}, \vec{b} \rangle + y \langle \vec{b}, \vec{b} \rangle) - (\langle \vec{a}, \vec{a} \rangle + y \langle \vec{a}, \vec{b} \rangle) \langle \vec{a}, \vec{b} \rangle] \\ =& (||\langle \vec{a}, \vec{b} \rangle||_2^2 - ||\vec{a}||_2^2 ||\vec{b}||_2^2) x + (||\vec{a}||_2^2 ||\vec{b}||_2^2 - ||\langle \vec{a}, \vec{b} \rangle||_2^2) y \\ =& (||\langle \vec{a}, \vec{b} \rangle||_2^2 - ||\vec{a}||_2^2 ||\vec{b}||_2^2)(x - y). \end{aligned} \tag{17}$$

According to the Cauchy-Schwartz inequality, we can obtain

$$||\langle \vec{a}, \vec{b} \rangle||_2^2 - ||\vec{a}||_2^2 ||\vec{b}||_2^2 \leq 0 \tag{18}$$

Then, we have

$$g(x) \begin{cases} > 0 & x < y \\ = 0 & x = y \\ < 0 & x > y. \end{cases} \tag{19}$$

Thus, for any position vector $\vec{y}$, when $x = 0$ or $x \to \infty$ (Equivalent to $\vec{x} = \vec{a}$ or $\vec{x} = \vec{b}$), the angle formed between $\vec{y}$ and $\vec{x}$ is maximum.

Without loss of generality, given a specific $\vec{y}$, if its maximum vector $\vec{x} = \vec{a}$, we can then set $\vec{y}$ to $\vec{a}$ and find its maximum vector again, which yields

$$\theta(\vec{x}, \vec{y}) \leq \theta(\vec{a}, \vec{y}) \leq \theta(\vec{b}, \vec{a})$$

The proof is completed.

Theorem 1 *Consider the case of a convex hull that does not contain the origin. Let $X = \{x_1, x_2, \ldots, x_k\}$ be a set of points, and let S be its convex hull. Then, the maximum angle between the position vectors of any two points in S is formed by the position vectors of two extreme points of S.*

Proof Assume that this case holds when k.

When $k = 2$, based on Lemma 1, the maximum angle is formed by the extreme points $\vec{x_1}$ and $\vec{x_2}$.

When $k \geq 3$, we can sort the elements of X such that for a point y in S, $\vec{x_k}$ maximizes the angle $\theta(\vec{y}, \vec{x_k})$. Furthermore, the points x in S are of the form:

$$\begin{aligned} & \lambda_1 \vec{x_1} + \lambda_2 \vec{x_2} + ... + \lambda_k \vec{x_k} \\ =& (\lambda_1 + ... + \lambda_{k-1})(\frac{\lambda_1 \vec{x_1}}{\lambda_1 + ... + \lambda_{k-1}} + ... + \frac{\lambda_{k-1} \vec{x_{k-1}}}{\lambda_1 + ... + \lambda_{k-1}}) \\ & + \lambda_k \vec{x_k}, \end{aligned} \tag{20}$$

where $(\frac{\lambda_1 \vec{x_1}}{\lambda_1 + \ldots + \lambda_{n-1}} + \ldots + \frac{\lambda_{k-1} \vec{x_{k-1}}}{\lambda_1 + \ldots + \lambda_{k-1}})$ is a position vector of a point located within the convex hull induced by $\{x_1, x_2, ..., x_{k-1}\}$. Through Lemma 1 and by definition, we can obtain

$$\theta(\vec{x}, \vec{y}) \leq \theta(\vec{x_k}, \vec{y}) \tag{21}$$

For any two points x and y in a convex hull S, by setting $\vec{y} = \vec{x_k}$ and using the above inequality twice, without loss of generality, we can assume that the vector $\vec{x_1}$ makes the largest angle with $\vec{x_k}$. Then, we can obtain

$$\theta(\vec{x}, \vec{y}) \leq \theta(\vec{x_k}, \vec{y}) \leq \theta(\vec{x_1}, \vec{x_k}) \tag{22}$$

By definition, $\theta(\vec{x_1}, \vec{x_k})$ is no greater than the maximum angle formed by any other two basis vectors.

The proof is completed.

Corollary 1 *When the convex hull of the input set V does not contain the origin, the largest angle between any two features after self-attention $V' = SA(V)$ is always less than or equal to the largest angle between features in V.*

Table 7: Comparison with the state-of-the-art methods on EPIC-KITCHEN dataset with the SlowFast backbone. *V.* and *N.* denote the *verb* and *noun* sub-tasks, respectively.

Subset	Method	0.1	0.2	0.3	0.4	0.5	Avg.
V.	BMN (Lin et al, 2019)	10.8	8.8	8.4	7.1	5.6	8.4
	G-TAD (Xu et al, 2020)	12.1	11.0	9.4	8.1	6.5	9.4
	ActionFormer (Zhang et al, 2022)	26.6	25.4	24.2	22.3	19.1	23.5
	TriDet	**28.6**	**27.4**	**26.1**	**24.2**	**20.8**	**25.4**
N.	BMN (Lin et al, 2019)	10.3	8.3	6.2	4.5	3.4	6.5
	G-TAD (Xu et al, 2020)	11.0	10.0	8.6	7.0	5.4	8.4
	ActionFormer (Zhang et al, 2022)	25.2	24.1	22.7	20.5	17.0	21.9
	TriDet	**27.4**	**26.3**	**24.6**	**22.2**	**18.3**	**23.8**

Remark 1 In the temporal action detection (TAD) task, the temporal feature sequences extracted by the pre-trained video classification backbone often exhibit high similarity and pure layer normalization (Ba et al, 2016) projects the input features onto the hypersphere in the high-dimensional space. Consequently, the convex hull induced by these features often does not encompass the origin. As a result, the self-attention operation causes the input features to become more similar, reducing the distinction between temporal features and hindering the performance of the TAD task.

B Additional Experimental Results

To further validate the robustness of TriDet, we conduct additional experiments on two single-label datasets: ActivityNet-1.3(Caba Heilbron et al, 2015) and EPIC-KITCHEN 100 (Damen et al, 2022), and a multilabel dataset Charades (Sigurdsson et al, 2016).

EPIC-KITCHEN 100 is a large-scale dataset of first-person vision that has two subtasks: *noun* localization (*e.g.* door) and *verb* localization (*e.g.* open the door). It contains 495 and 138 videos with 67,217 and 9,668 action segments for training and test, respectively. The number of action classes for *noun* and *verb* are 300 and 97. ActivityNet shares 200 classes of action with the HACS dataset and contains 10,024 videos for training, as well as 4,926 videos for test. Charades is a large-scale common household activities dataset that contains 7,985 and 1,863 videos for training and test, with 49,809 and 16,691 action segments, respectively.

For EPIC-KITCHEN, we report IoU thresholds at [0.1:0.5:0.1]. For ActivityNet, we report the result at the IoU threshold [0.5, 0.75, 0.95], and the average mAP is computed at [0.5:0.95:0.05]. We report the average IoU with thresholds [0.1: 0.1: 0.9] for the Charades datasets.

The initial learning rate is set to 10^{-4} for Charades and EPIC-KITCHEN, and 10^{-3} for ActivityNet. We train for 9, 23, 19 and 15 for Charades, EPIC-KITCHEN *verb*, EPIC-KITCHEN *noun*, ActivityNet (including a warmup of 5, 5, 5 and 10 epochs).

For ActivityNet, the number of bins B of the Trident-head is set to 12, the convolution window w is set to 15 and the scale factor k is set to 1.3. For Charades, and EPIC-KITCHEN, the number of bins B of the Trident-head is set to 16, the convolution window w is set to 1, and the scale factor k is set to 5 for Charades and 1.5 for EPIC-KITCHEN.

We show their results in Table 7, Table 8 and Table 9, respectively. With the same backbone network, TriDet achieves State-of-the-art performance on these datasets, demonstrating its robustness.

References

Alwassel H, Heilbron FC, Escorcia V, Ghanem B (2018) Diagnosing error in temporal action detectors. In: Eur. Conf. Comput. Vis.

Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv preprint arXiv:160706450

Bao H, Dong L, Piao S, Wei F (2021) Beit: Bert pre-training of image transformers. arXiv preprint arXiv:210608254

Beltagy I, Peters ME, Cohan A (2020) Longformer: The long-document transformer. arXiv preprint arXiv:200405150

Bhosale S, Nag S, Kanojia D, Deng J, Zhu X (2023) Diffsed: Sound event detection with denoising diffusion. arXiv preprint arXiv:230807293

Bodla N, Singh B, Chellappa R, Davis LS (2017) Soft-nms–improving object detection with one line of code. In: Int. Conf. Comput. Vis.

Caba Heilbron F, Escorcia V, Ghanem B, Carlos Niebles J (2015) Activitynet: A large-scale video benchmark for human activity understanding. In: IEEE Conf. Comput. Vis. Pattern Recog.

Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S (2020) End-to-end object detection with transformers. In: Eur. Conf. Comput. Vis., Springer

Table 8: Comparison with the state-of-the-art methods on ActivityNet-1.3 dataset.

Method	Backbone	0.5	0.75	0.95	Avg.
PGCN (Zeng et al, 2019)	I3D	48.3	33.2	3.3	31.1
ReAct (Shi et al, 2022)	TSN	49.6	33.0	8.6	32.6
BMN (Lin et al, 2019)	TSN	50.1	34.8	8.3	33.9
G-TAD (Xu et al, 2020)	TSN	50.4	34.6	9.0	34.1
AFSD (Lin et al, 2021)	I3D	52.4	35.2	6.5	34.3
TadTR (Liu et al, 2022b)	TSN	51.3	35.0	9.5	34.6
TadTR (Liu et al, 2022b)	R(2+1)D	53.6	37.5	10.5	36.8
VSGN (Zhao et al, 2021)	I3D	52.3	35.2	8.3	34.7
PBRNet (Liu and Wang, 2020)	I3D	54.0	35.0	9.0	35.0
TCANet+BMN (Qing et al, 2021)	TSN	52.3	36.7	6.9	35.5
TCANet+BMN (Qing et al, 2021)	SlowFast	54.3	**39.1**	**8.4**	**37.6**
TALLFormer (Cheng and Bertasius, 2022)	Swin	54.1	36.2	7.9	35.6
ActionFormer (Zhang et al, 2022)	R(2+1)D	**54.7**	37.8	**8.4**	36.6
TriDet	R(2+1)D	**54.7**	**38.0**	**8.4**	**36.8**

Table 9: Comparison with the state-of-the-art methods on the Charades datasets.

Method	Backbone	Charades			
		0.2	0.5	0.7	Avg
PointTAD (Tan et al, 2022)	I3D	17.5	13.5	9.1	12.1
ASL (Shao et al, 2023)	I3D	24.5	16.5	9.4	15.4
TriDet	I3D	**27.1**	**20.4**	**13.2**	**18.4**

Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. In: IEEE Conf. Comput. Vis. Pattern Recog.

Chen G, Zheng YD, Wang L, Lu T (2022) Dcan: improving temporal action detection via dual context aggregation. In: AAAI

Chen T, Kornblith S, Norouzi M, Hinton G (2020a) A simple framework for contrastive learning of visual representations. In: Int. Conf. Machine Learning, PMLR

Chen X, Fan H, Girshick R, He K (2020b) Improved baselines with momentum contrastive learning. arXiv preprint arXiv:200304297

Cheng F, Bertasius G (2022) Tallformer: Temporal action localization with long-memory transformer. Eur Conf Comput Vis

Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: IEEE Conf. Comput. Vis. Pattern Recog.

Damen D, Doughty H, Farinella GM, , Furnari A, Ma J, Kazakos E, Moltisanti D, Munro J, Perrett T, Price W, Wray M (2022) Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. Int J Comput Vis

Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805

Dong Y, Cordonnier JB, Loukas A (2021) Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In: Int. Conf. Machine Learning

Escorcia V, Caba Heilbron F, Niebles JC, Ghanem B (2016) Daps: Deep action proposals for action understanding. In: Eur. Conf. Comput. Vis.

Feichtenhofer C, Fan H, Malik J, He K (2019) Slowfast networks for video recognition. In: Int. Conf. Comput. Vis.

Feichtenhofer C, Li Y, He K, et al (2022) Masked autoencoders as spatiotemporal learners. Advances in neural information processing systems 35:35946–35958

Fu J, Gao J, Xu C (2023) Semantic and temporal contextual correlation learning for weakly-supervised temporal action localization. IEEE Transactions on Pattern Analysis and Machine Intelligence

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. IEEE Conf Comput Vis Pattern Recog

Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: IEEE Conf. Comput. Vis. Pattern Recog.

Jiang YG, Liu J, Roshan Zamir A, Toderici G, Laptev I, Shah M, Sukthankar R (2014) THUMOS challenge: Action recognition with a large number of classes

Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, et al (2023) Segment anything. arXiv preprint arXiv:230402643

Lee P, Uh Y, Byun H (2020) Background suppression network for weakly-supervised temporal action localization. In: Proceedings of the AAAI conference on artificial intelligence, pp 11320–11327

Lee P, Kim T, Shim M, Wee D, Byun H (2023) Decomposed cross-modal distillation for rgb-based temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 2373–2383

Li X, Wang W, Wu L, Chen S, Hu X, Li J, Tang J, Yang J (2020) Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. Adv Neural Inform Process Syst

Lin C, Li J, Wang Y, Tai Y, Luo D, Cui Z, Wang C, Li J, Huang F, Ji R (2020) Fast learning of temporal action proposal via dense boundary generator. In: AAAI

Lin C, Xu C, Luo D, Wang Y, Tai Y, Wang C, Li J, Huang F, Fu Y (2021) Learning salient boundary feature for anchor-free temporal action localization. In: IEEE Conf. Comput. Vis. Pattern Recog.

Lin T, Zhao X, Su H, Wang C, Yang M (2018) Bsn: Boundary sensitive network for temporal action proposal generation. In: Eur. Conf. Comput. Vis.

Lin T, Liu X, Li X, Ding E, Wen S (2019) Bmn: Boundary-matching network for temporal action proposal generation. In: Int. Conf. Comput. Vis.

Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: Int. Conf. Comput. Vis.

Lin Z, Geng S, Zhang R, Gao P, de Melo G, Wang X, Dai J, Qiao Y, Li H (2022) Frozen clip models are efficient video learners. In: European Conference on Computer Vision, Springer, pp 388–404

Liu Q, Wang Z (2020) Progressive boundary refinement network for temporal action detection. In: Proceedings of the AAAI Conference on Artificial Intelligence

Liu S, Xu M, Zhao C, Zhao X, Ghanem B (2023) Etad: Training action detection end to end on a laptop. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 4524–4533

Liu X, Hu Y, Bai S, Ding F, Bai X, Torr PH (2021) Multi-shot temporal event localization: a benchmark. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 12596–12606

Liu X, Bai S, Bai X (2022a) An empirical study of end-to-end temporal action detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 20010–20019

Liu X, Wang Q, Hu Y, Tang X, Zhang S, Bai S, Bai X (2022b) End-to-end temporal action detection with transformer. IEEE Trans Image Process

Liu Z, Mao H, Wu CY, Feichtenhofer C, Darrell T, Xie S (2022c) A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition

Long F, Yao T, Qiu Z, Tian X, Luo J, Mei T (2019) Gaussian temporal awareness networks for action localization. In: IEEE Conf. Comput. Vis. Pattern Recog.

Loshchilov I, Hutter F (2017) SGDR: Stochastic gradient descent with warm restarts. In: Int. Conf. Learn. Represent.

Loshchilov I, Hutter F (2019) Decoupled weight decay regularization. In: Int. Conf. Learn. Represent.

Nag S, Xu M, Zhu X, Pérez-Rúa JM, Ghanem B, Song YZ, Xiang T (2022) Multi-modal few-shot temporal action detection via vision-language meta-adaptation. arXiv preprint arXiv:221114905

Nag S, Zhu X, Deng J, Song YZ, Xiang T (2023) Difftad: Temporal action detection with proposal denoising diffusion. arXiv preprint arXiv:230314863

OpenAI (2023) Gpt-4 technical report. 2303.08774

Oquab M, Darcet T, Moutakanni T, Vo HV, Szafraniec M, Khalidov V, Fernandez P, Haziza D, Massa F, El-Nouby A, Howes R, Huang PY, Xu H, Sharma V, Li SW, Galuba W, Rabbat M, Assran M, Ballas N, Synnaeve G, Misra I, Jegou H, Mairal J, Labatut P, Joulin A, Bojanowski P (2023) Dinov2: Learning robust visual features without supervision

Paul S, Roy S, Roy-Chowdhury AK (2018) W-talc: Weakly-supervised temporal activity localization and classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 563–579

Qing Z, Su H, Gan W, Wang D, Wu W, Wang X, Qiao Y, Yan J, Gao C, Sang N (2021) Temporal context aggregation network for temporal action proposal refinement. In: IEEE Conf. Comput. Vis. Pattern Recog.

Rezatofighi H, Tsoi N, Gwak J, Sadeghian A, Reid I, Savarese S (2019) Generalized intersection over union: A metric and a loss for bounding box regression. In: IEEE Conf. Comput. Vis. Pattern Recog.

Shao J, Wang X, Quan R, Zheng J, Yang J, Yang Y (2023) Action sensitivity learning for temporal action localization. arXiv preprint arXiv:230515701

Shi D, Zhong Y, Cao Q, Zhang J, Ma L, Li J, Tao D (2022) React: Temporal action detection with relational queries. In: Eur. Conf. Comput. Vis.

Shi D, Zhong Y, Cao Q, Ma L, Li J, Tao D (2023) Tridet: Temporal action detection with relative boundary modeling. In: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 18857–18866

Sigurdsson GA, Varol G, Wang X, Farhadi A, Laptev I, Gupta A (2016) Hollywood in homes: Crowdsourcing data collection for activity understanding. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, Springer, pp 510–526

Sridhar D, Quader N, Muralidharan S, Li Y, Dai P, Lu J (2021) Class semantics-based attention for action detection. In: Int. Conf. Comput. Vis.

Szegedy C, Ioffe S, Vanhoucke V, Alemi A (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI

Tan J, Tang J, Wang L, Wu G (2021) Relaxed transformer decoders for direct action proposal generation. In: Int. Conf. Comput. Vis.

Tan J, Zhao X, Shi X, Kang B, Wang L (2022) Pointtad: Multi-label temporal action detection with learnable query points. Advances in Neural Information Processing Systems 35:15268–15280

Tang TN, Kim K, Sohn K (2023) Temporalmaxer: Maximize temporal context with only max pooling for temporal action localization. arXiv preprint arXiv:230309055

Tian Z, Shen C, Chen H, He T (2019) Fcos: Fully convolutional one-stage object detection. In: Int. Conf. Comput. Vis.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. Adv Neural Inform Process Syst 30

Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L (2018) Temporal segment networks for action recognition in videos. IEEE Trans Pattern Anal Mach Intell

Wang L, Huang B, Zhao Z, Tong Z, He Y, Wang Y, Wang Y, Qiao Y (2023) Videomae v2: Scaling video masked autoencoders with dual masking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 14549–14560

Wang R, Chen D, Wu Z, Chen Y, Dai X, Liu M, Jiang YG, Zhou L, Yuan L (2022) Bevt: Bert pretraining of video transformers. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 14733–14743

Weng Y, Pan Z, Han M, Chang X, Zhuang B (2022) An efficient spatio-temporal pyramid transformer for action detection. In: Eur. Conf. Comput. Vis.

Wu Y, He K (2018) Group normalization. In: Eur. Conf. Comput. Vis.

Xie Z, Zhang Z, Cao Y, Lin Y, Bao J, Yao Z, Dai Q, Hu H (2022) Simmim: A simple framework for masked image modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9653–9663

Xu M, Zhao C, Rojas DS, Thabet A, Ghanem B (2020) G-tad: Sub-graph localization for temporal action detection. In: IEEE Conf. Comput. Vis. Pattern Recog.

Xu M, Perez Rua JM, Zhu X, Ghanem B, Martinez B (2021) Low-fidelity video encoder optimization for temporal action localization. Adv Neural Inform Process Syst

Yang L, Peng H, Zhang D, Fu J, Han J (2020) Revisiting anchor mechanisms for temporal action localization. IEEE Trans Image Process

Yang M, Chen G, Zheng YD, Lu T, Wang L (2022) Basictad: an astounding rgb-only baseline for temporal action detection. arXiv preprint arXiv:220502717

Yeung S, Russakovsky O, Jin N, Andriluka M, Mori G, Fei-Fei L (2017) Every moment counts: Dense detailed labeling of actions in complex videos. International Journal of Computer Vision

Zeng R, Huang W, Tan M, Rong Y, Zhao P, Huang J, Gan C (2019) Graph convolutional networks for temporal action localization. In: Int. Conf. Comput. Vis.

Zhang C, Wu J, Li Y (2022) Actionformer: Localizing moments of actions with transformers. In: Eur. Conf. Comput. Vis.

Zhang S, Chi C, Yao Y, Lei Z, Li SZ (2020) Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: IEEE Conf. Comput. Vis. Pattern Recog.

Zhao C, Thabet AK, Ghanem B (2021) Video self-stitching graph network for temporal action localization. In: Int. Conf. Comput. Vis.

Zhao H, Yan Z, Torresani L, Torralba A (2019) Hacs: Human action clips and segments dataset for recognition and temporal localization. arXiv preprint arXiv:171209374

Zhao P, Xie L, Ju C, Zhang Y, Wang Y, Tian Q (2020) Bottom-up temporal action localization with mutual regularization. In: Eur. Conf. Comput. Vis.

Zhao Y, Xiong Y, Wang L, Wu Z, Tang X, Lin D (2017) Temporal action detection with structured segment networks. In: ICCV

Zhu Z, Tang W, Wang L, Zheng N, Hua G (2021) Enriching local and global contexts for temporal action localization. In: Int. Conf. Comput. Vis.