TE-TAD: Towards Full End-to-End Temporal Action Detection via Time-Aligned Coordinate Expression

Supplementary Material

A. Overview

In this material, we mainly provide three aspects: (1) training details, (2) implementation details, and (3) additional analysis.

B. Extended Training and Inference

As referred to in Sec. 3.5, we describe a more detailed training loss, following the standard bipartite matching loss [4] denoted as Eq. (8). The matched permutation indices, $\hat{\pi}$, play a pivotal role in this process, identifying the best match for each query to a ground truth instance or a "no action" class ($\varnothing$) in case of non-matching. The matched indices are computed as follows:

$$\hat{\pi} = \arg\min \sum_{i=1}^{N_q} \mathcal{L}_{match}(\mathcal{A}_i, \hat{\mathcal{A}}_{\pi(i)}). \tag{A}$$

The matched indices represent optimal one-to-one matching between the predicted proposals and the ground truth targets using the bipartite matching algorithm [12]. These matched pairs are utilized in the matching loss Eq. (8) and formally defined as follows:

$$\begin{aligned}\mathcal{L}_{match}(\mathcal{A}_i, \hat{\mathcal{A}}_{\sigma(i)}) = &\lambda_{cls}\mathcal{L}_{cls}(c_i, \hat{p}^{(L_D)}_{\pi(i)}) + \\ &\mathbb{1}_{[c_i \neq \varnothing]}[\lambda_{reg}\mathcal{L}_{reg}(b_i, \hat{b}^{(L_D)}_{\pi(i)})],\end{aligned} \tag{B}$$

where both b_i and $\hat{b}_{\pi(i)}$ contain start and end timestamps of ground truth predicted segments. For the classification loss $\mathcal{L}_{cls}$, we use the focal loss [17], which effectively addresses the class imbalance issue. For the regression loss, our model utilizes DIoU [36] ($\mathcal{L}_{diou}$) and log-ratio distance for the width. DIoU evaluates the relative center distance and GIoU [24] once, while the log-ratio compares the widths relatively. The regression loss $\mathcal{L}_{reg}$ is formally defined as follows:

$$\mathcal{L}_{reg} = \mathcal{L}_{diou}(b_i, \hat{b}^{(L_D)}_{\pi(i)}) + \log\left(d_i / \hat{d}^{(L_D)}_{\pi(i)}\right), \tag{C}$$

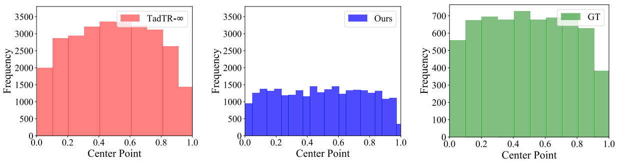
where d_i is the length of the ground truth calculated by $(e_i - s_i)/2$. The log term represents the log-ratio of the widths, comparing the ground truth width d_i with the predicted width $\hat{d}^{(L_D)}_{\pi(i)}$. This modified regression loss term ensures that the relative scale of the ground truth and predicted segments are in the time-aligned coordinate expression.

C. Implementation Details

Environment All experiments are conducted using PyTorch 2.0.1 on a single NVIDIA A6000 GPU.

Training Our TE-TAD is trained end-to-end with the AdamW [22] optimizer. The initial learning rate is set to 0.0001, and the weight decay factor is 0.05 for all datasets. Training durations are specified per dataset: THUMOS14 undergoes 150 epochs, EpicKitchens 80, and ActivityNet v1.3, 16. Learning rate decay is implemented towards the latter part of training: epoch 140 for THUMOS14, 70 for EpicKitchens, and 14 for ActivityNet v1.3, with the decay factor being 0.1. Batch sizes are set to 8 for THUMOS14, 32 for ActivityNet v1.3, and 4 for EpicKitchens.

Architecture The number of hidden dimensions in the transformer is 256 for THUMOS14 and 512 for both ActivityNet v1.3 and EpicKitchens. The number of multi-scale levels, L, is configured as 8 for THUMOS14 and 6 for both ActivityNet v1.3 and EpicKitchens. The model architecture includes four encoder layers and six ($L_D = 6$) decoder layers. For the AQS, the length of each sector, T_{sector}, is set to 128 for THUMOS14 and 64 for both ActivityNet v1.3 and EpicKitchens. Additionally, the number of queries for each sector is set to 50 for all datasets, with a cap of 3000 max queries per video to maintain computational efficiency.



(a) TadTR-∞ (b) Ours (c) GT

Figure A. Distribution of matching instability across video instances: (a) shows TadTR-∞ where higher frequency at the center of the videos indicates instability due to normalized coordinates; (b) presents Ours with a more uniform distribution and similar to the distribution of the ground truth, suggesting less bias to coordinate expression; and (c) shows the number of ground truth (GT) instances.

Content Embedding	Position Embedding	mAP@AVG	
		RAW	w/NMS
Sine Embedding [37]		65.0	65.9
	✓	64.1	66.2
Learnable Query [33]		63.0	64.8
	✓	65.1	67.2
Encoder Memory		64.7	66.1
	✓	64.2	66.2
Gaussian Random		64.4	65.9
	✓	64.6	66.8
Zero Init		65.0	66.4
	✓	66.1	67.9

Table A. Comparative analysis of different query proposal methods on THUMOS14 for various content and position embedding techniques, with and without the adoption of NMS.

Others We adopt an exponential moving average similar to those used in [27, 32] to mitigate overfitting. Following the previous implementation [27], we employ a slight Gaussian noise to the input features. The noise scales of standard deviation are set to 0.25 for THUMOS14 and EpicKitchens and 0.75 for ActivityNet v1.3.

D. Additional Analysis

Effect of Query Proposal Methods In the realm of query-based detection, the method of initializing queries for the decoder plays a pivotal role in the model's ability to accurately localize and classify actions. Building upon the foundations laid by two-stage query-based methods such as Deformable-DETR [37] and DINO [33], we explore various query proposal techniques and their influence on THUMOS14. Table A presents a comparative analysis, revealing the mAP@AVG for different content and position embedding strategies, both in raw form and when supplemented by NMS. This examination allows us to discern the efficacy of each method and its contribution to the robustness and precision of action detection.

Analysis of Coordinate Expression Bias In our experiments, we aim to investigate where matching instabilities predominantly occur within the video timeline. To conduct this analysis, we count the changed matching permutations between proposals and ground truth across the entire training ((a) and (b)) and the number of ground truth instances (c) on THUMOS14. To show the relative locations, we plot the center point of proposals and ground truths, and we normalize timeline values between 0 and 1 to represent relative locations within the video. The distribution patterns depicted in Fig. A elucidate the differences in stability and bias between our method and TadTR-∞. Our approach exhibits a more uniform distribution of the center points across the timeline, indicating less instability and bias towards the center of the video timeline compared to TadTR-∞. In contrast, using normalized coordinates in TadTR-∞ contributes to its instability, as seen in the skewed distribution towards the center points of the video timeline. Especially, this skewed distribution at the center points of the video timeline is affected by the non-uniform differentiation of the *sigmoid* function. Consequently, by aligning closely with the ground truth distribution, our method shows its effective and unbiased temporal action detection, indicating the robustness of our time-aligned coordinate expression.