DiGIT: Multi-Dilated Gated Encoder and Central-Adjacent Region Integrated Decoder for Temporal Action Detection Transformer

Supplementary Material

A. Implementation Details

Table B provides an overview of the hyperparameter settings used for each dataset: THUMOS14 [14], ActivityNet v1.3 [10], and HACS-Segment [50]. The hyperparameter settings are divided into four main categories: general model configuration, query selection, encoder settings, and training configuration. Additionally, this section describes how the video snippets were processed and the hardware configurations used in our experiments.

General Settings The general model configuration defines parameters for the architecture, including model dimension, feedforward network dimension, number of encoder and decoder layers, and attention settings. For THUMOS14, we use a model dimension (D) of 512, while a smaller value of 256 is applied to ActivityNet v1.3 and HACS-Segment, reflecting their lower temporal complexity. The feedforward network (FFN) dimension (D_h) is set to 2048 across all datasets to ensure sufficient capacity. Both the encoder (L_E) and decoder (L_D) consist of 6 layers. For self-attention and deformable attention, 8 attention heads are used, with 4 sampling points per head in the deformable attention mechanism. Dropout is disabled for all datasets.

Query Selection THUMOS14 [14] contains videos with a wide range of action durations, requiring a flexible query selection strategy. For this dataset, we adopt a top-k selection approach, where k is set to 50% of the total number of multi-scale features across all levels $\sum_{l=1}^{L} T_l$. This approach is consistent with the adaptive query selection (AQS) [15], which also leverages a variable number of queries based on the video features. In contrast, for ActivityNet v1.3 [10] and HACS-Segment [50], we use a fixed number of 100 queries per video. This is because these datasets consist of videos with more uniform durations than the variable-length videos in THUMOS14.

Training Settings We use AdamW [30] as the optimizer across all datasets. For THUMOS14, a learning rate of 0.00005 is used, which is lower than the 0.0001 applied to ActivityNet v1.3 and HACS-Segment, reflecting the higher complexity of THUMOS14. A weight decay of 0.05 is used for ActivityNet v1.3, while a smaller value of 0.0001 is applied to THUMOS14 and HACS-Segment to balance regularization and learning stability. Gradient clipping is applied to all datasets with a maximum norm of 0.1, ensuring numerical stability during training. Batch sizes are set to 4 for THUMOS14 and 16 for both ActivityNet v1.3 and HACS-Segment, reflecting differences in model dimensionality and dataset characteristics.

Loss Coefficients For the matching loss $\mathcal{L}_{\text{match}}$, the classification loss (Focal Loss [24]) has a coefficient of 2 for THUMOS14, compared to 1 for ActivityNet v1.3 and HACS-Segment, due to the high class overlap in THUMOS14. The GIoU [33] loss coefficient is consistently set to 2 across all datasets. Log-width loss is omitted for THUMOS14 but included for ActivityNet v1.3 and HACS-Segment with a coefficient of 1, as precise action width estimation is more critical in these datasets.

Feature Extraction We follow the preprocessing strategies in prior works [6, 15, 47] to extract video features. For THUMOS14 [14], I3D [5] and InternVideo2 [41] features are extracted using 16-frame snippets with a stride of 4 frames. For ActivityNet v1.3 [10] and HACS-Segment [50], InternVideo2 [41] features are extracted using 16-frame snippets with a stride of 8 frames. For ActivityNet v1.3 [10], R2+1D [38] features are extracted using 16-frame non-overlapping snippets, as follows in TSP [2].

Hardware Configuration All experiments were conducted using a single NVIDIA A100 GPU.

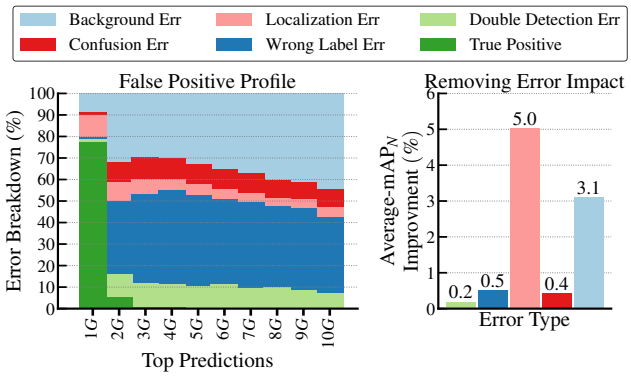
B. Additional Experiments

Analysis on Computational Complexity Table A shows the comparison of computational complexity between state-of-the-art methods and our method with InternVideo2 [41] features on THUMOS14 [14]. For a fair comparison, inference times are calculated per video for all models, and all results are measured on an NVIDIA A100 GPU. Our method demonstrates competitive inference speed for the detection phase, comparable to anchor-free detectors such as ActionFormer [47] and ActionMamba [6]. Moreover, DiGIT significantly reduces post-processing time compared to anchor-free detectors. This efficiency is achieved as query-based detectors leverage their set-prediction mechanism, which ideally eliminates the need for post-processing steps like NMS. Compared to TE-TAD and TadTR, DiGIT exhibits slight increases in memory usage and processing time due to its enhanced model capacity.

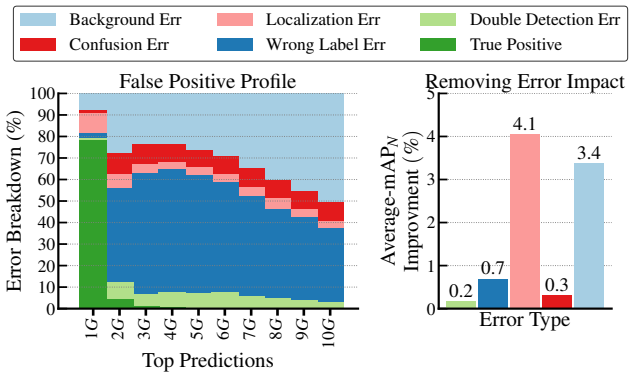
DETAD [1] Analysis Fig. A provides a DETAD [1] analysis on false positive errors for various models, including TadTR [27], TadTR enhanced with our MDGE and CAID, TE-TAD [15], and our DiGIT. The analysis divide the error types into categories such as background errors, localization errors, confusion errors, and wrong label errors. Our method demonstrates a significant reduction in false positive localization errors compared to other models, indicating its ability to correctly localize action boundaries.

Head Type	Method	Inference Time (ms)			Peak Memory (GB)
		Detector	PostProcess	Total	
Anchor-free	ActionFormer [47]	42.47±45.40	31.75±23.36	74.22±50.60	3.04
	ActionMamba [6]	28.76±43.44	31.75±24.27	60.51±49.15	2.88
Query-based	TadTR [27]	92.07±93.30	17.39±10.00	109.64±103.92	1.24
	TadTR [27] + Ours	93.79±97.71	18.29±10.33	111.22±108.68	1.73
	TE-TAD [15]	27.71±3.72	5.61±16.20	33.32±17.55	2.48
	DiGIT	31.27±5.27	8.08±18.50	39.35±19.24	3.52

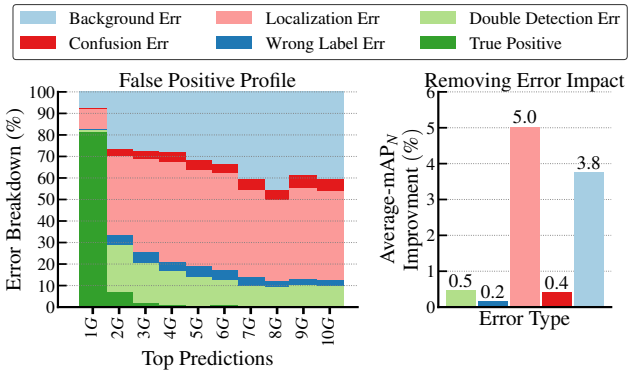
Table A. **Analysis of computational complexity with InternVideo2 features on THUMOS14.** The table compares inference times for the detection phase, post-processing phase, and total runtime for state-of-the-art anchor-free and query-based methods.



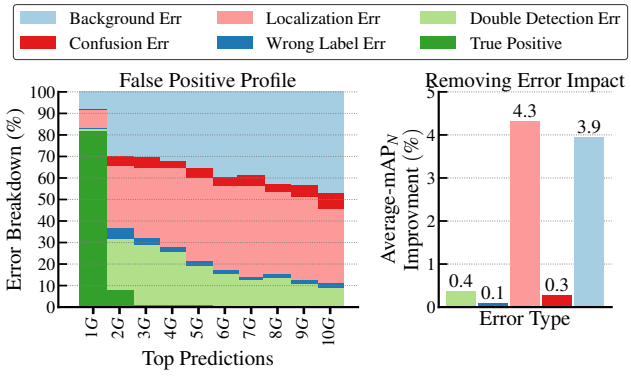
(a) TadTR [27]



(b) TadTR [27] + Ours



(c) TE-TAD [15]



(d) DiGIT

Figure A. **False Positive Analysis.** Each row compares DETAD error analysis for various models: (a) TadTR [27], (b) TadTR with our MDGE and CAID, (c) TE-TAD [15], and (d) DiGIT.

Dataset	THUMOS14 [14]	ActivityNet v1.3 [10]	HACS-Segment [50]
General Setting			
Model Dim. (D)	512	256	256
FFN Dim. (D_h)	2048	2048	2048
Num. Enc. Layer (L_E)	6	6	6
Num. Dec. Layer (L_D)	6	6	6
Num. Self-Attn. Head	8	8	8
Num. Deform. Attn. Head (the number of m)	8	8	8
Num. Deform. Attn. Point (the number of p)	4	4	4
Dropout	0	0	0
Query Selection Setting			
Query Selection Type	Ratio	Fix Number	Fix Number
Top-k Selection Ratio	0.5	-	-
Max Num. Queries (max N_q)	900	-	-
Num. Queries (N_q)	-	100	100
MDGE Settings			
Num. Dilated Conv. (N_d)	2	2	2
Kernel Size	11	11	11
Training Setting			
Optimizer	AdamW [30]	AdamW [30]	AdamW [30]
Learning Rate	0.00005	0.0001	0.0001
Weight Decay	0.0001	0.05	0.0001
Batch Size	4	16	16
Gradient Cliping	0.1	0.1	0.1
Loss Coefficient for $\mathcal{L}_{match}$			
Classification Loss (Focal Loss [24])	2	1	1
GIoU [33] Loss	2	2	2
Log-Width Loss	0	1	1

Table B. **Hyperparameter Settings across THUMOS14, ActivityNet v1.3, and HACS-Segment.** The table presents model parameters, query selection criteria, encoder settings, and training configurations for each dataset. "Model Dim." and "FFN Dim." refer to the dimensionality of model and feedforward network layers, respectively. "Num. Deform. Attn. Head" and "Num. Deform. Attn. Point" represents the number of attention heads and sampling points in the deformable attention. "Num. DilatedConv." and "Kernel Size" specify the configurations in MDGE. The query selection method is adapted per dataset, with top-k selection used for THUMOS14 and fixed query counts for ActivityNet v1.3 and HACS-Segment.