Context-Enhanced Memory-Refined Transformer for Online Action Detection

Supplementary Material

A. Diagnising Context Modeling for OAD

Existing methods [2, 5, 6] suffer from a training-inference discrepancy, causing short-term context imbalance and a non-causal leakage during anticipation, resulting in learning biases. Fig. 1 shows the learning biases present in existing works from a performance perspective.

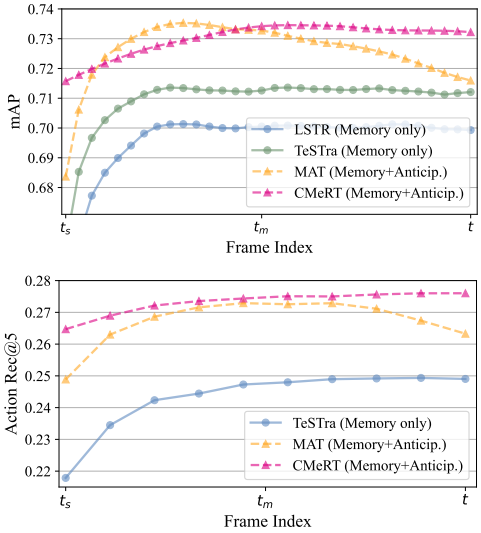

Figure 1. Frame performance within the short-term memory on THUMOS'14(top) and EK100(bottom).

First, we observe that early frames in the short-term memory are poorly learned, resulting in significantly lower performance. These poorly represented frames serve as low-quality samples, impairing the learning of classifier to effectively predict the latest frame. In contrast, CMeRT improves performance for early frames, though a performance gap still remains compared to the latest frame. The remaining gap is due to the use of a shorter near-past context, which limits the amount of context available to earlier frames compared to the latest one. Our empirical findings demonstrate that shorter near-past contexts are more beneficial, as they act as a form of data augmentation by exposing frames to less context. Naive approaches [2, 5, 6] that omit near-past context can also be seen as a form of data augmentation. However, they over-augment the data, introducing poor training samples that hamper the learning process.

Second, the performance curve of the anticipation-based method MAT [2] confirms the presence of non-causal leakage, as it shows significantly higher performance for intermediate frames compared to the latest frame. CMeRT however, effectively mitigate this leakage and learning bias, prioritizing the learning of the latest frame.

B. Experiments

Hyperparameters. The hyperparameters used for each dataset are summarized in Tab. 1.

Table 1. Hyperparameters for different experimental settings.

	THUMOS'14	CrossTask	EK100
batch size	32	32	32
epoch	12	12	12
warmup	8	5	10
learning rate	2e-4	7e-5	7e-5
weight decay	5e-5	1e-5	1e-4

MAT-rw and MAT-stream. We implement MAT-rw and MAT-stream based on the state-of-the-art memory-based model MAT [2] to evaluate standard approaches for addressing the training-inference discrepancy.

In MAT-rw, we assign a higher weight to the loss of the latest frame to mitigate the learning bias towards intermediate frames. Specifically, the weight is set to 1.2 for THUMOS'14 and 3.0 for CrossTask and EK100.

In MAT-stream, only the latest frame in the short-term memory is used for training, while other short-term frames are discarded to align with the inference. we modify the sliding window sampling by setting the stride to 1, ensuring all video frames are used for training. However, this increases the training set size compared to using a stride equal to the short-term memory length, resulting in more training samples and updates per epoch than the standard MAT. To mitigate this, we adjust the batch size to match the number of updates per epoch as in MAT [2].

C. Advancing OAD

DinoV2 Features. We use the Dinov2 ViT-g/14 model [1] to extract advanced RGB features for THUMOS'14 and CrossTask. We replace only the RGB features while other features, such as optical flow, remain unchanged. For THUMOS'14, following [5], we extract video frames at a rate of 24 FPS and divide the video into chunks of 6 frames, using the intermediate frame of each chunk for RGB feature extraction. The feature extraction is performed at the chunk level, meaning evaluation occurs every 0.25 seconds. The feature encoding process for CrossTask is similar to THUMOS"14, except that the chunk size is increased to 24 frames to align with the existing feature set.

While the advanced feature extractor improves performance, it also increases the computational burden. Following [5], we report the runtime for end-to-end online inference on THUMOS'14, including two-stream feature extraction in Tab. 2. Specifically, DenseFlow [4] is used to compute

optical flow, while RGB features are extracted using either ResNet52 [3] or the DinoV2 model. The results in Tab. 2 align with prior works [2, 5], confirming that optical flow remains the primary speed bottleneck. Compared to optical flow feature, the runtime for DinoV2 RGB feature extraction remains manageable. However, the DinoV2 model inference can be further accelerated through techniques such as model distillation, model weights quantization or conversion to Optimized formats, like TorchScript and ONNX. Model inference optimization is already a well-established practice in the industry, providing significant opportunities to leverage more advanced features while maintaining efficiency.

Table 2. Efficiency analysis of feature extraction on THUMOS'14. The performance is reported in frames per second(FPS)

Optical Flow		RGB	
Computation	Extraction	ResNet52	DinoV2
8.6	47.6	69.0	13.9

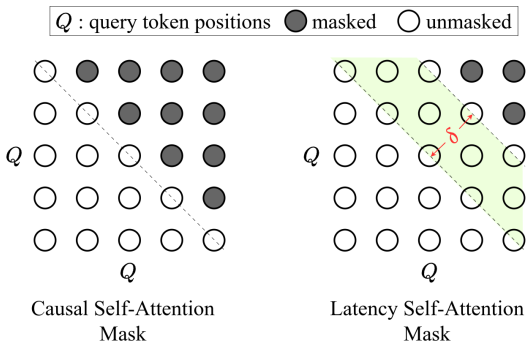

Figure 2. Self-attention masking to control query interactions.

OAD with latency For applications where delays are acceptable or post-prediction refinement is required, it is valuable to explore the advantages of incorporating limited future information into online action detection. To explore this, we introduce a future latency parameter, δ, and propose the first OAD baseline with future latency. Specifically, we construct base models based on Testra [6], MAT [2], and our model CMeRT by replacing the causal mask in short-term self-attention with a new latency mask, as shown in Fig. 2. This new mask allows each short-term frame to additionally access the near-future information up to a limit of δ.

We evaluate the new OAD with latency setting using various base models and latency settings, with results in Fig. 3. Incorporating future latency improves performance across all models. Even a small latency, *e.g.* $\delta = 0.5$ can lead to greater improvements, with further gains expected as the latency increases. CMeRT consistently outperforms others by a large margin, demonstrating its robustness.

D. Qualitative Results

Fig. 4 and Fig. 5 show some qualitative results for THUMOS'14 and CrossTask, respectively. The bar charts present

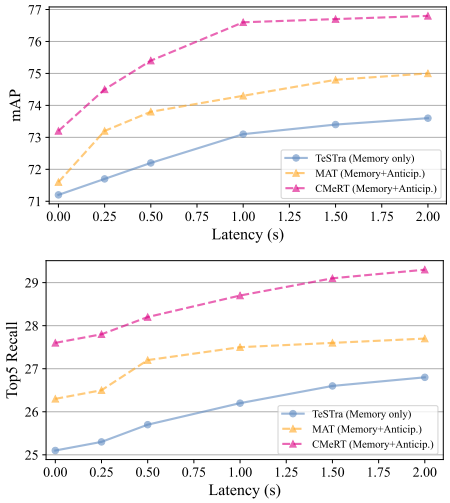

Figure 3. OAD performance under varied future latency on THUMOS'14(top) and EK100(bottom).

a comparison between the ground truth and the predictions from MAT [2] and our method CMeRT. The curve plots display the confidence in identifying the current true action. The results highlight that CMeRT effectively reduce the misclassification between background and foreground actions. Additionally, it improves the distinction between similar actions *(PoleVault vs. HighJump)*. However, it struggles with short actions *(Whisk mixture & add coffee)* or small subjects in similar backgrounds *(SoccerPenalty)*.

E. Extra Ablation Studies

Query configuration in the long-term compressor: We test on four query configurations (stage1-stage2): 16-16, 16-32, 32-32, and 32-64 on THUMOS'14. The mAP is 72.8%, 73.2%, 72.9%, and 72.8%, respectively. The results suggest that intermediate configurations are optimal, as excessive queries introduce noise and redundancy, while too few causes the loss of valuable information.

Short over long future: We designed the long-term memory (t_l to t_s) to generate a near-future (t_s to $t_s + T_f$) that overlaps and extends beyond the short memory to serve a pseudo-future for all short-term frames. Experimentally, generating a short near-future is favored over a longer one, as longer pseudo-futures are more challenging and costly, leading to degraded quality (Fig. 6). Even using the true future, performance saturates beyond a certain length (Fig. 7), which justifies our use of short-future generation.

Long-short division: We evaluate the impact of long-short term division on performance. As shown in Fig. 8, excessive long-term memory introduces noise, while insufficient long-term causes information loss. The short-term length has minimal impact if sufficient long-term is provided. Besides, near-future generation is less impacted by the division, since it always predicts the future following the long-term.

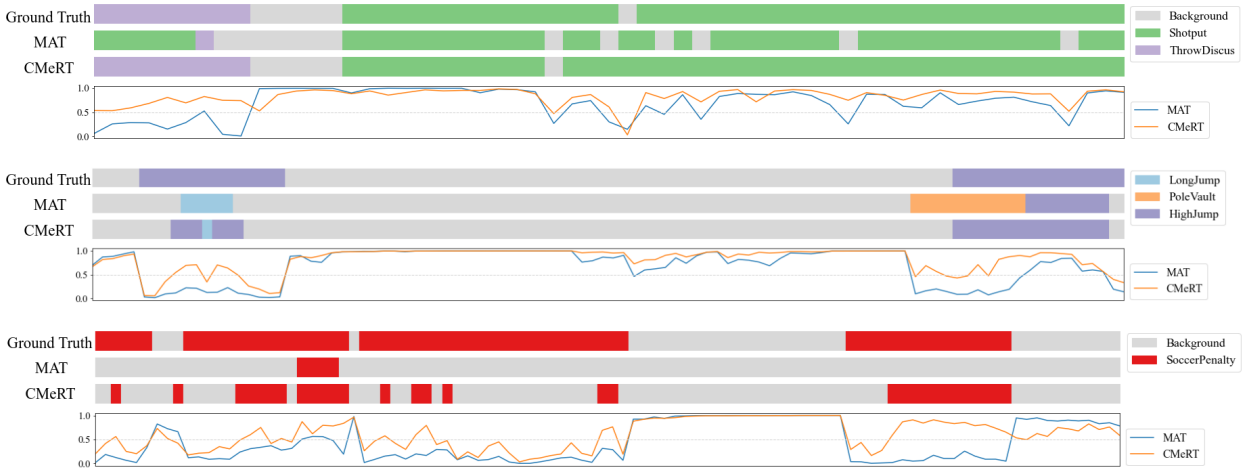

Figure 4. Quality results on THUMOS'14 - bar charts show predictions; curve plots for confidence of the true action.

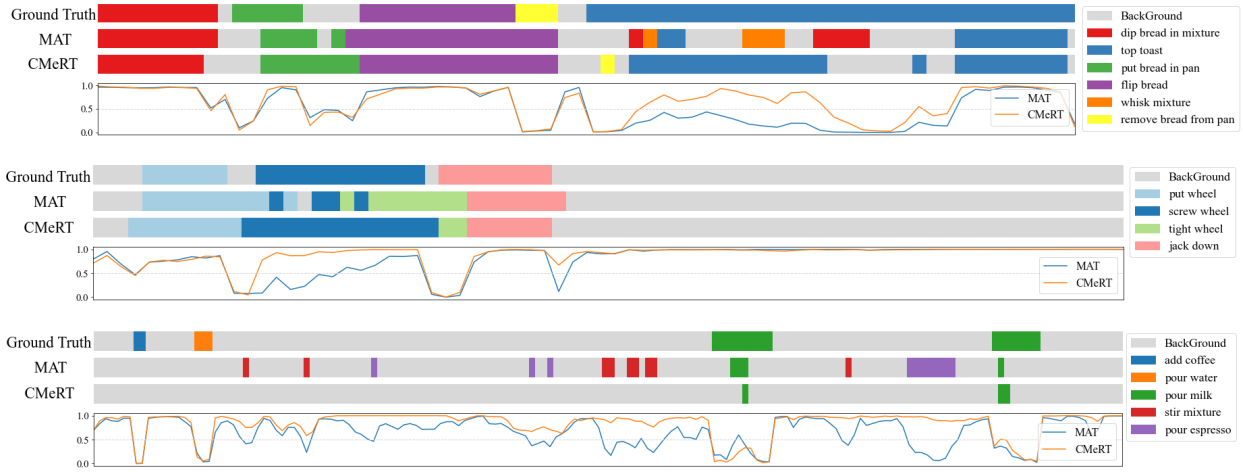

Figure 5. Quality results on CrossTask: top - Make French Toast, middle - Change a Tire, bottom - Make a Latte

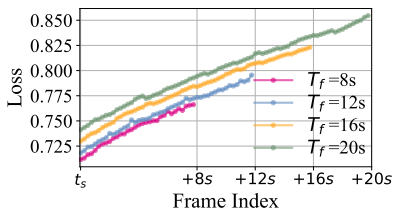

Figure 6. Extended future generation reduces quality (on THUMOS'14).

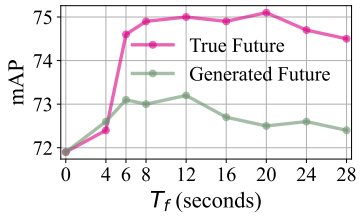

Figure 7. Distant future not helpful (on THUMOS'14).

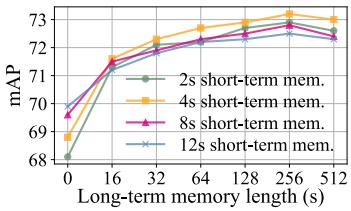

Figure 8. Impact of long-short division on THUMOS'14.

References

[1] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[2] Jiahao Wang, Guo Chen, Yifei Huang, Limin Wang, and Tong Lu. Memory-and-anticipation transformer for online action understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13824–13835, 2023. 1, 2

[3] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[4] Shiguang Wang, Zhizhong Li, Yue Zhao, Yuanjun Xiong, Limin Wang, and Dahua Lin. Denseflow, 2020. https://github.com/open-mmlab/denseflow. 1

[5] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34:1086–1099, 2021. 1, 2

[6] Yue Zhao and Philipp Krähenbühl. Real-time online video detection with temporal smoothing transformers. In *European Conference on Computer Vision*, pages 485–502. Springer, 2022. 1, 2