Asynchronous Temporal Modeling with Two-Agent Framework for Streaming Dense Video Captioning

Supplementary Material

A. Preliminary Experiments Details

Our analysis of threshold-based streaming models revealed two critical issues: (1) No universal threshold works optimally across all videos. (2) The acceptable threshold range is extremely narrow, with small deviations causing drastic behavioral changes: lower thresholds result in silence, higher thresholds cause excessive captioning. This instability highlights Threshold-Gated Discrepancy (TGD), where minor threshold variations lead to inconsistent captioning. Figure 6 illustrates these findings through probability distributions and heatmaps. `Takusen` addresses TGD by adopting an event-driven framework with Silence Token Prediction that adapts dynamically to video content while maintaining temporal coherence.

B. More Analysis of Fixed Decoding Points

Our experiments on ActivityNet Captions explored fixed decoding points (FDPs) impact without an Oracle agent. Table 5 and Figure 8 illustrate these findings. Performance peaks at 3 FDPs (CIDEr: 30.4, F1-score: 47.0). While higher FDPs improve recall (32.3→44.2), they significantly reduce precision (50.2→16.8 at 8 FDPs), causing over-segmentation and reduced caption coherence. These findings reinforce the importance of our Oracle model. Without Oracle guidance, 3 FDPs provides the best recall-precision trade-off, while further increases degrade performance, highlighting the necessity of event-driven decoding over fixed-time strategies.

C. Additional Visualization Results

Figure 7 presents additional YouCook2 results, confirming `Takusen`'s robust performance across different video domains. `Takusen` successfully captures complex human-object interactions while maintaining precise temporal alignment. VideoLLaMA3-7B shows limitations in event boundary alignment, often generating overly short captions that lack contextual depth and failing to track multi-step actions smoothly.

D. Prompts for Streaming Interaction

We explored two prompt templates for Oracle-Listener interaction during streaming training as well as inference: *(1) Pure Oracle-Guided:* Oracle explicitly dictates event boundaries, providing stable supervision that enhances event alignment and reduces unnecessary captioning. *(2) Oracle-Guided with Auto-Forecasting:* Listener autonomously determines event boundaries after initial Oracle instructions. Our experiments revealed TGD with auto-forecasting: instability in event boundary detection causing either over-silence (low thresholds) or over-talking (high thresholds). We adopted the Pure Oracle-Guided approach in our final implementation for stable captioning and better event alignment.

Pure Oracle-guided v.s. Mixed Prompts

Pure Oracle-Guided:
Oracle: Please describe events in the video, determining start/end times yourself.
Listener: I will determine event times and describe them.
Oracle: Event 0 starts now.
Listener: Let me track Event 0.
Oracle: Event 0 has ended. Describe it.
Listener: Event 0: A camera pans over a snowy area leading to a man snowboarding down a mountain.
...

Oracle-Guided with Auto-Forecasting:
Oracle: Please describe events in the video, determining start/end times yourself.
Listener: I will determine event times and describe them.
Listener: Event 0 starts now.
Listener: Let me track Event 0.
Listener: Event 0 has ended. Describe it.
Listener: Event 0: A camera pans over a snowy area leading to a man snowboarding down a mountain.
...

E. Efficiency Analysis

`Takusen` has 0.418B trainable parameters and processes video at 2 FPS, supporting videos up to 30 minutes. Despite employing two models, `Takusen` is more efficient than single-model alternatives. We compare against TRACE-uni [13], a representative single-model LMM-based method, measuring FLOPs per sample. `Takusen` requires 29.1 TFLOPs compared with TRACE-uni's 39.2 TFLOPs, while achieving higher CIDEr (43.7 vs. 29.2 on ActivityNet Captions). This efficiency stems from the Oracle's sparse 1 FPS sampling with a compact 2B model, which offloads temporal localization from the Listener's denser 2 FPS processing with an 8B model. After receiving Oracle prompts, the Listener takes only 222.5ms to generate

captions, demonstrating real-time capability.

F. Latency and Causal Processing

We clarify that both the Oracle and Listener process frames *causally* without accessing future content. At any time t_c, the Oracle has analyzed frames up to $t_c^{(O)}$ while the Listener has processed up to $t_c^{(L)}$, where $t_c^{(L)} \leq t_c^{(O)} \leq t_c$. The term "see further into the future" refers to the Oracle's faster processing rate rather than actual future access. The temporal gap $\Delta = t_c^{(O)} - t_c^{(L)}$ averages 1.2s on ActivityNet Captions and 0.8s on YouCook2. This gap arises because the Oracle processes fewer frames per second with a smaller model, enabling faster processing per unit of video time. This represents an architectural efficiency advantage rather than a violation of the streaming constraint.

G. Cross-Dataset Generalization

To evaluate how well `Takusen` generalizes across domains, we conduct cross-dataset experiments between ActivityNet Captions (diverse everyday activities) and YouCook2 (procedural cooking steps). When trained on ActivityNet Captions and tested on YouCook2, `Takusen` achieves a CIDEr of 28.4, METEOR of 5.8, and F1 of 29.2. Conversely, training on YouCook2 and testing on ActivityNet Captions yields CIDEr 31.6, METEOR 7.8, and F1 42.1. Despite significant domain differences, `Takusen` retains over 70% of in-domain performance in both directions, demonstrating that the Oracle-Listener framework learns transferable temporal reasoning patterns rather than overfitting to dataset-specific statistics. Joint training on both datasets achieves near-optimal performance on both test sets (CIDEr 42.9 on ActivityNet Captions and 39.1 on YouCook2), confirming that the framework handles diverse video types without catastrophic forgetting.

H. FDP Selection and Robustness

The optimal number of fixed decoding points (FDPs) correlates with dataset-level event density: ActivityNet Captions averages 3.7 events per video (optimal at 3 FDPs), while YouCook2 averages 7.8 events per video (optimal at 10 FDPs). A simple heuristic of setting FDPs to average video length divided by average event length provides a practical guideline. Importantly, FDP selection is robust: varying the count by ± 2 from the optimum changes CIDEr by less than 1.5 points. In contrast, threshold-based methods exhibit 5–10 point CIDEr swings with threshold changes smaller than 0.01 (as illustrated in Figure 2). Since FDPs are dataset-level constants rather than per-video parameters, they are significantly more practical to tune than instance-specific thresholds.

I. Discussion on Memory and Architecture

Memory vs. Caption Triggering. Memory management and caption triggering are orthogonal challenges in streaming video captioning. Our primary contribution addresses *when* to generate captions by solving the TGD problem through Oracle-Listener collaboration, rather than proposing a new memory mechanism. Existing streaming methods [84] also employ growing context; the distinction lies in our event-driven triggering mechanism. Practically, our 128K context window covers all benchmark videos (up to 30 minutes at 2 FPS). For longer videos, a sliding window approach could maintain performance with minimal degradation. Memory compression techniques could integrate with our framework as orthogonal future work.

Architecture vs. Base Model Strength. The performance gains of `Takusen` stem from the Oracle-Listener architecture rather than a stronger base model. As shown in Table 3, CM^2 alone achieves CIDEr 33.1, which is lower than Streaming GIT's 41.2. The full `Takusen` system achieves 43.7, representing a +10.6 CIDEr gain over CM^2 alone. Removing the Oracle and relying solely on FDPs causes a substantial drop from 43.7 to 30.4, further isolating the Oracle's contribution to the overall performance.

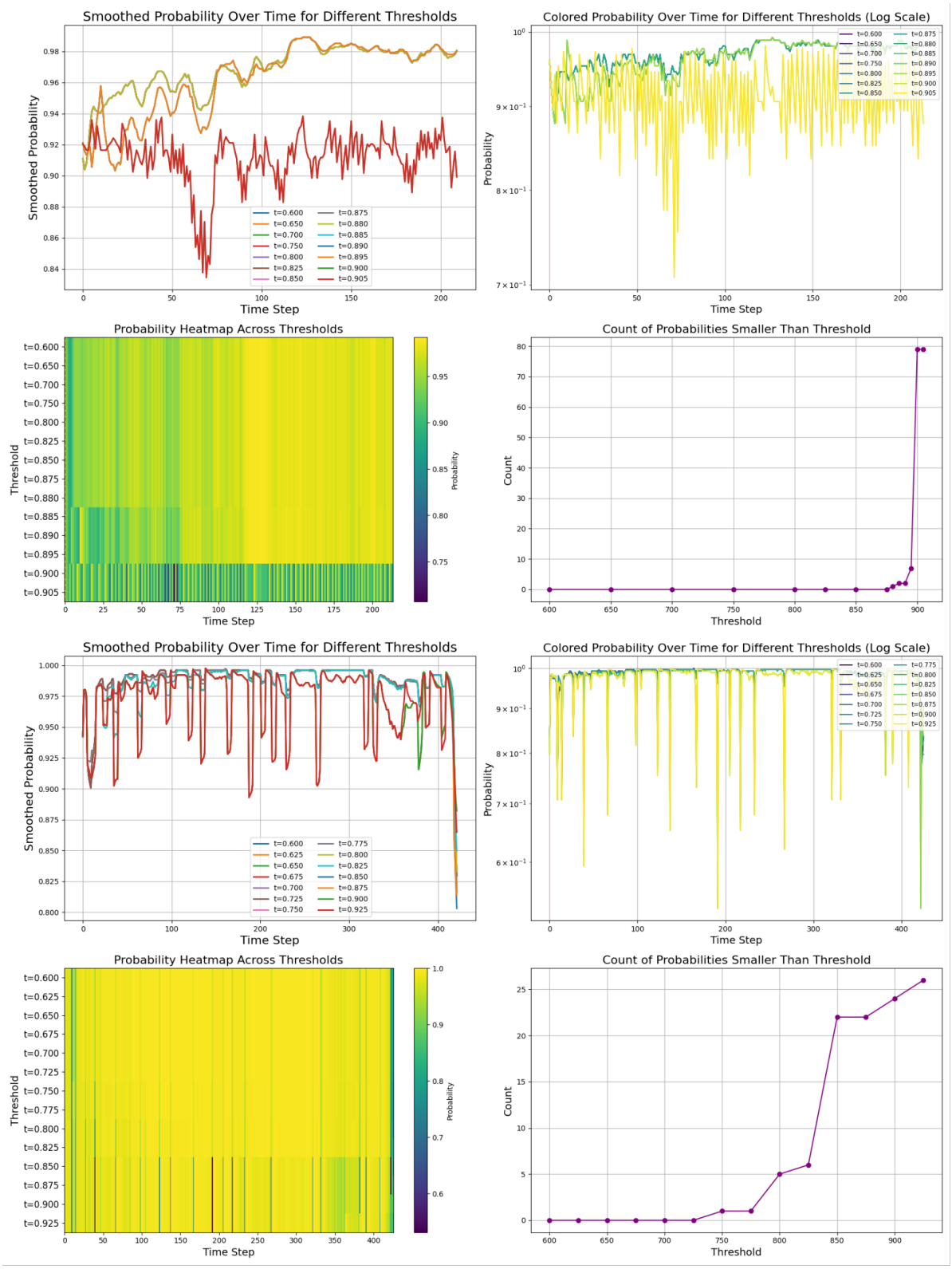

Figure 6. Analysis of probability-based thresholding, illustrating instability where small threshold variations lead to abrupt changes in output behavior.

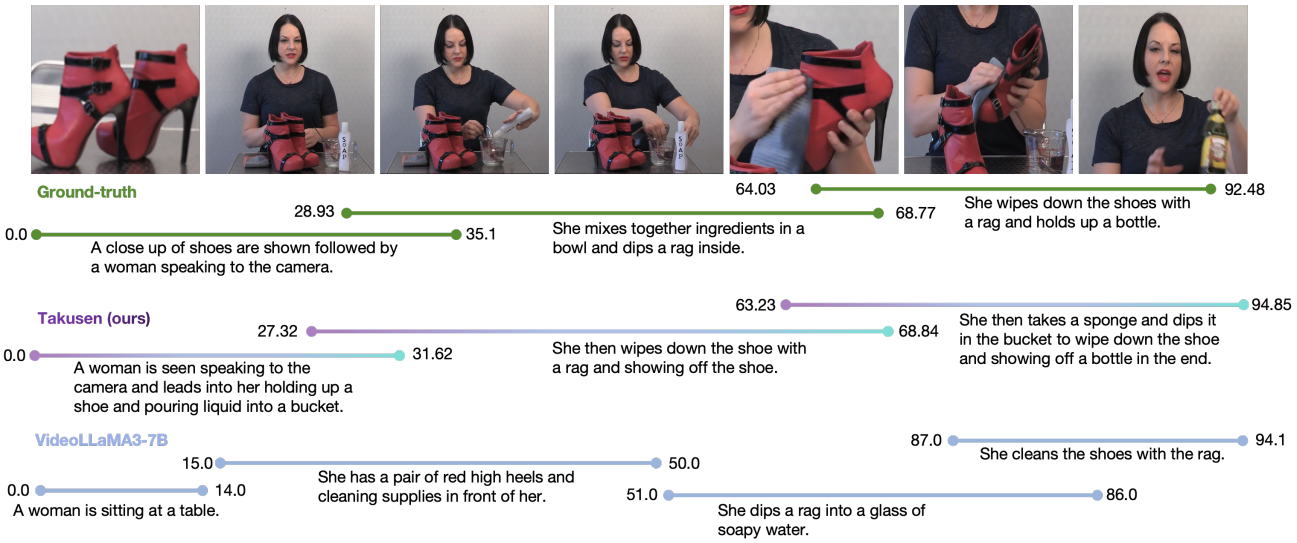

Figure 7. More Visualization: Comparison of our `Takusen` with ground-truth and VideoLLaMA3-7B.

# FDPs	CIDEr	SODA_c	METEOR	BLEU-4	ROUGE-L	Precision	Recall	F1
2	23.5	4.7	5.5	1.6	9.6	44.5	32.3	37.4
3	30.4	6.0	6.7	1.6	11.3	50.2	44.2	47.0
4	23.7	5.2	5.7	1.2	8.6	39.0	41.3	40.1
5	16.9	4.6	5.0	0.8	7.0	31.4	37.0	33.8
6	13.3	3.7	4.1	0.7	5.4	25.4	33.4	28.6
7	11.7	3.3	3.7	0.7	4.6	21.3	30.7	24.9
8	8.9	2.8	3.1	0.4	3.5	16.8	26.6	20.3
Avg.	18.3	4.3	4.8	1.0	7.1	32.7	35.1	33.2

Table 5. Analysis on the number of decoding points (#FDPs) on ActivityNet Captions dataset, where there is no Oracle model.

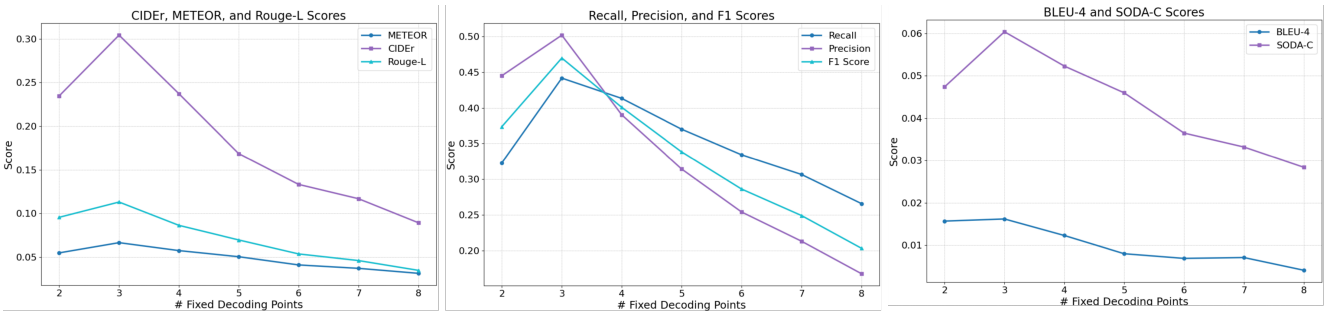

Figure 8. Impact of fixed decoding points on ActivityNet Captions metrics. Performance peaks at 3 FDPs; additional FDPs reduce CIDEr, METEOR, and precision due to redundant captioning.