Breaking the Second Barrier: Sub-Second Timestamped Omni-Modal Captioning

Zhihe Yang[1,2,*] **Xin Li**[1,*] **Hao Tan**[1] **Zhao Zhong**[1] **Liefeng Bo**[1] **Yunjian Xu**[2,†]
[1]Tencent Hunyuan, Guangdong, China.
[2]The Chinese University of Hong Kong, Hong Kong SAR, China.

Abstract

Omni-modal caption models equipped with precise timestamp grounding are crucial for fine-grained video understanding and temporally controllable video generation. However, existing open-source models, and even proprietary systems, struggle to provide fine-grained timestamp grounding for short video captioning. To bridge this gap, we present **Deci-Omni-Captioner**, an advanced open-source omni-modal captioning framework designed for sub-second timestamp precision. During the Supervised Fine-Tuning (SFT) stage, we leverage Optical Character Recognition (OCR) and Automatic Speech Recognition (ASR) models to strictly calibrate visual text and spoken dialogue boundaries, offering highly deterministic temporal anchors to endow the model with sub-second timestamp grounding capabilities. Furthermore, we curate a specialized dataset for multi-reward Reinforcement Learning (RL) and propose Span-Specific Credit Assignment (SSCA). Unlike conventional global advantage normalization, which dilutes reward signals across weakly coupled multimodal descriptions, SSCA calculates advantages independently for different structural spans, effectively isolating penalties and rewards across semantic and temporal dimensions. Extensive experiments demonstrate that Deci-Omni-Captioner establishes state-of-the-art (SOTA) performance among open-source models on multiple semantic comprehension benchmarks (e.g., AVUT, UGC, VCapsBench). On our challenging Deci-Timestamp Bench, it substantially outperforms leading proprietary models (Gemini 2.5 and 3.0 series) on Dialogue and Subtitle grounding while remaining competitive on visual-event localization.

1 Introduction

The rapid emergence of video generation models (e.g., Seedance [1, 2, 3], Kling [4], Gen [5, 6], HunyuanVideo [7, 8], Veo [9], SORA [10], and Wan [11]) has brought significant attention to the necessity of high-quality training data. To obtain video generation models with strict audio-visual consistency, omni-modal caption models play a crucial role. During the pre-training phase of generative models, billions of videos typically require annotation by caption models. These video-caption pairs are subsequently utilized to endow the generation models with precise semantic instruction-following and comprehension capabilities.

To enable temporal control in video generation, caption models must provide precise timestamp grounding, which is a capability currently absent in open-source omni-modal models. Furthermore, while capturing highly dynamic details requires processing videos at higher frame rates (e.g., 10 FPS), existing models are largely restricted to 2 FPS and paradoxically exhibit degraded comprehension when evaluated at these unseen higher rates, as illustrated in Fig. 1(a). Although some closed-source models (e.g., Gemini series models [12, 13]) can successfully understand high-FPS videos, their

*Equal Contributions.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

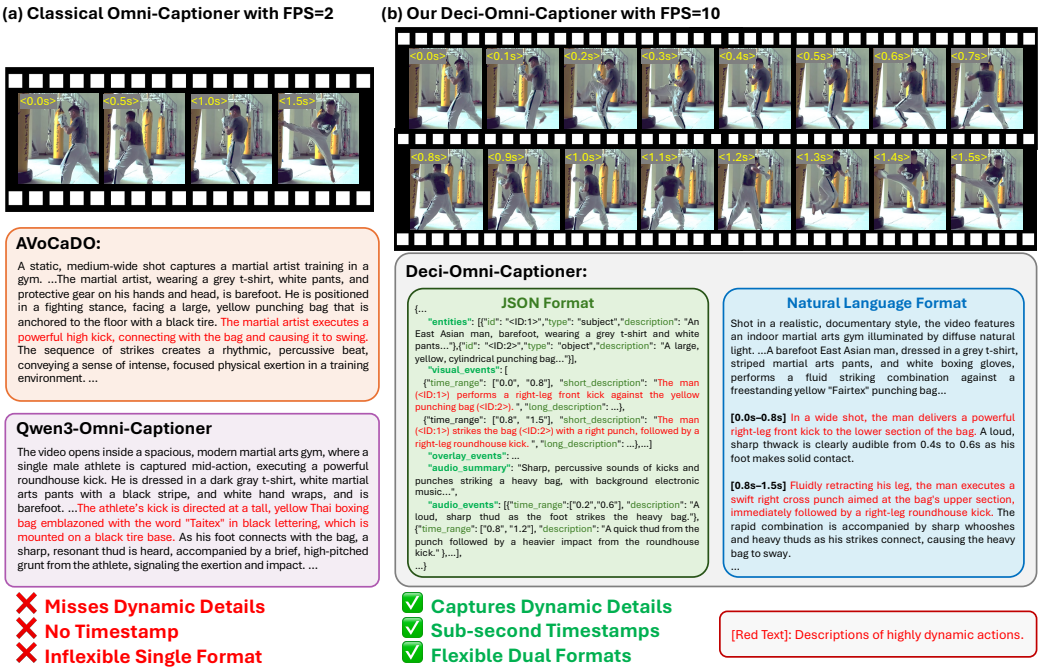


Figure 1: Qualitative comparison of video captioning capabilities. (a) Classical Omni-Captioners are limited to processing at 2 frames per second (FPS) (performing worse at unseen higher frame rates like 10 FPS), resulting in missed dynamic details. Furthermore, these models inherently lack timestamp grounding and are restricted to an inflexible single-format output. (b) Our Deci-Omni-Captioner processes video at 10 FPS to successfully capture highly dynamic details with sub-second timestamp accuracy. Additionally, it offers flexible dual-format outputs (JSON and Natural Language (NL)).

timestamp accuracy remains limited, typically with errors around 0.5s. This margin of error is still insufficient for the video generation domain, where clip lengths mostly range from 5 to 15 seconds.

To alleviate these issues, we propose **Deci-Omni-Captioner**, fine-tuned on a specialized dataset tailored for short videos comprising 110K videos for SFT and 2K videos for RL. Notably, Deci-Omni-Captioner not only achieves SOTA performance across numerous multimodal comprehension benchmarks among open-source models, but also surpasses Gemini in timestamp accuracy. Moreover, our model offers flexible dual-format outputs (structured JSON and NL), allowing users to easily parse and assemble captions for specific requirements, as shown in Fig. 1(b).

During the SFT data construction phase, we introduce a novel timestamp alignment mechanism to achieve sub-second precision for boundary-explicit text and speech. Specifically, we employ OCR models on high-FPS frames to strictly calibrate the temporal boundaries of visual text, and utilize ASR models on audio tracks to align the exact start and end times of spoken dialogue. Experimental results demonstrate that this explicit temporal anchoring not only significantly enhances the model's timestamp grounding capabilities for text and speech, but also improves its overall temporal understanding, enabling zero-shot generalization to temporally ambiguous visual events.

To enhance semantic comprehension and further improve timestamp grounding of the model, we construct a high-quality multi-reward dataset for the RL stage. To address the reward dilution problem inherent in global sequence-level optimization, we propose a novel Span-Specific Credit Assignment (SSCA) mechanism. By distributing reward credits independently across different structural spans based on their specific reward dimensions, SSCA effectively isolates penalties and rewards. Empirical verification confirms that this fine-grained supervision increases the training efficiency of Group Sequence Policy Optimization (GSPO) [14].

Overall, our contributions are summarized as follows: (1) We open-source **Deci-Omni-Captioner**, an advanced omni-modal captioning model with sub-second timestamp grounding capabilities. It achieves SOTA performance among open-source models on multiple semantic benchmarks and

surpasses leading proprietary systems (e.g., Gemini) in timestamp precision. (2) We introduce a scalable data construction pipeline that leverages OCR and ASR for explicit temporal anchoring, enabling the creation of high-quality video captioning datasets with strict timestamp alignment. (3) We propose SSCA, a novel fine-grained advantage normalization mechanism designed to resolve the reward dilution problem and enhance the RL training efficiency of caption models under multi-reward supervision.

2 Related Works

Omni-Modal Video Understanding and Timestamp Grounding. Recent advancements in open-source omni-modal foundation models, such as Qwen2.5-Omni [15] and Qwen3-Omni [16], have catalyzed the development of stronger audio-visual captioners. Building upon these foundations, representative open-source omni-modal caption models have emerged, such as Video-SALMONN2+7B [17], UGC-VideoCaptioner [18], AVoCaDO [19], and Omni-Captioner [20]. However, these models do not explicitly support fine-grained timestamp prediction in the generated captions. In parallel, powerful purely visual captioning models have been built on strong vision-language (VL) backbones such as Qwen2.5-VL [21] and Qwen3-VL [22]. Alongside these, representative caption-oriented methods such as Vid2Seq [23], Streaming Dense Video Captioning [24], Video ReCap [25], and TA-Prompting [26] have been proposed to improve dense description and timestamp grounding. However, these methods do not model audio and usually provide only coarse event-level timing rather than reliable fine-grained timestamps. Proprietary omni-modal models such as Gemini [12] are currently the closest available option, yet they remain closed-source, costly for billion-scale annotation, and still limited in timestamp precision. Consequently, open-source omni-modal caption models with reliable fine-grained timestamps are urgently needed.

RL-based Alignment for Caption Models. Recent caption alignment has progressed from offline preference learning to online RL. Early caption-oriented post-training mainly followed the offline paradigm, where preference pairs are constructed in advance and optimized with objectives in the style of Direct Preference Optimization (DPO). Representative examples of this paradigm include MrDPO [27, 17] and AVC-DPO [28], alongside broader efforts in DPO-based video-language alignment [29]. More recent work increasingly turns to online RL, since it often yields stronger improvements than offline preference learning in caption quality and generalization [19, 30, 31, 32, 33, 34]. Existing online methods mainly follow two directions. One line relies on judge-style rewards, where a Large Language Model (LLM) or Vision-Language Model (VLM) directly evaluates the generated caption, as in VLRM [30] and parts of VideoCap-R1 [31]. The other line adopts more verifiable rewards, such as VQA-based supervision or decomposed completeness/correctness objectives, as in CapRL [32] and CCCaption [33]. For both directions, the core difficulty lies in reward assignment. Unlike math or code, where tokens are tightly coupled through a single reasoning trajectory and outcome-level rewards are often sufficient [35, 36], caption quality is distributed across relatively weakly coupled visual and auditory factors, and different spans may vary independently across reward dimensions. Nevertheless, existing sequence-level objectives such as GRPO [37], DAPO [38], GSPO [14], and SAPO [39] still assign a shared advantage to all tokens in a caption, which limits learning efficiency under multi-reward supervision. CapRL further shows that simply increasing the number of rewards does not automatically translate into proportional training gains [32]. These observations suggest the need for RL methods that are specifically designed for caption models and support finer-grained credit assignment.

3 Deci-Omni-Captioner

3.1 Model Architecture

Deci-Omni-Captioner is built upon the Qwen3-Omni [16] architecture. The original Qwen3-Omni employs Time-aligned Multimodal Rotary Position Embedding (TM-RoPE) [15, 16] to align temporal information across modalities. While effective for general multimodal synchronization, this implicit encoding of temporal order through position embeddings lacks a direct symbolic representation of wall-clock time. Consequently, it leads to significant errors in absolute time alignment for short videos requiring sub-second localization.

To achieve precise timestamp grounding, there are two intuitive approaches to explicitly expose absolute time to the language model. The first is to render timestamps directly onto video frames and rely on OCR to recover the corresponding time. The second is to prepend explicit timestamp tokens before each visual or audio segment, thereby converting timestamp alignment into a textual conditioning problem. As shown in Fig. 2, we adopt the second design as our main architectural modification, as it is significantly more flexible and does not require modifying the original video frames. This explicit token-based strategy aligns with the approaches increasingly adopted by recent VL models (e.g., Qwen3-VL [22] and Keye-VL 1.5 [40]). Our empirical ablation (Table 3) further confirms its superiority, consistent with the findings of TimeLens [41].

3.2 Data Construction

3.2.1 SFT Data Pipeline

Our SFT dataset consists of 110K high-quality videos. The comprehensive data distribution across multiple dimensions is detailed in Appendix A.1.1 (Fig. 6). As illustrated in Fig. 3(a), our data construction pipeline is systematically divided into five steps:

Step 1: Coarse Caption Generation. We utilize Gemini-2.5-Pro to generate initial captions in a structured JSON format. To ensure high quality, we employ a two-stage prompting strategy: we first prompt the model to generate a visual caption, which is subsequently injected as context to guide the generation of the audio caption. This context-aware approach yields significantly more detailed and accurate audio descriptions (see Appendix A.1.2 for exact prompts). However, due to the inherent limitations of Gemini-2.5-Pro in precise timestamp grounding, this step only produces raw captions with coarse timestamps.

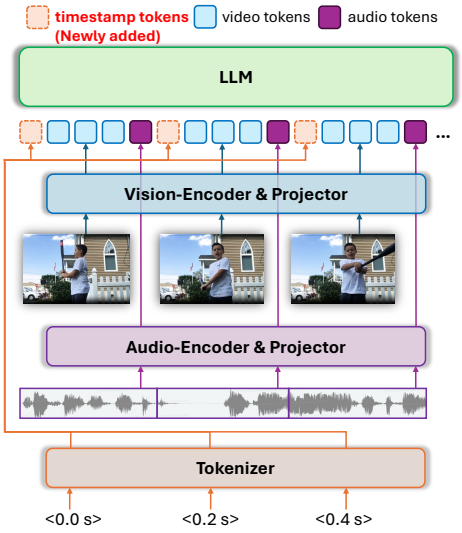


Figure 2: Based on the Qwen3-Omni framework, the Deci-Omni-Captioner improves timestamp grounding by inserting absolute timestamp tokens directly preceding the associated visual and audio tokens.

Step 2: Visual Timestamp Alignment via OCR. To accurately align the timestamps of visual text in the raw captions, we extract individual video frames and process them with the HunyuanOCR [42] model. This provides precise appearance and disappearance timestamps for on-screen text elements, such as subtitles, watermarks, and stylized fonts.

Step 3: Audio Timestamp Alignment via ASR. For videos containing human speech, we employ the Doubao-Seed-ASR-2.0 [43] model to extract accurate timestamps for the auditory content. It natively supports high-precision word-level timestamp generation during its decoding process, enabling us to precisely align the temporal boundaries of spoken dialogue in the raw captions.

Step 4: Timestamp Injection and Refinement. We then use Gemini-2.5-Pro to inject the exact timestamps obtained from the OCR and ASR steps back into the raw captions from Step 1. As demonstrated in our experiments (Table 4), this explicit anchoring mechanism significantly improves the overall timestamp precision. Notably, the model effectively zero-shot generalizes this precise temporal understanding to other general visual and audio events in the video.

Step 5: Quality Filtering. Finally, we apply a rigorous filtering pipeline to the refined data. We filter out samples based on length constraints (removing excessively short or long captions), self-repetition (eliminating repeated text), and JSON parsing errors (discarding unparsable outputs). After a final distribution-based sampling, we obtain the ultimate 110K high-quality samples for SFT.

3.2.2 RL Data Pipeline

To further elevate the model's performance in both semantic accuracy and timestamp grounding, we construct a highly refined RL dataset comprising 2K carefully annotated videos, as illustrated in Fig. 3(b). The construction of this RL dataset proceeds through the following three steps:

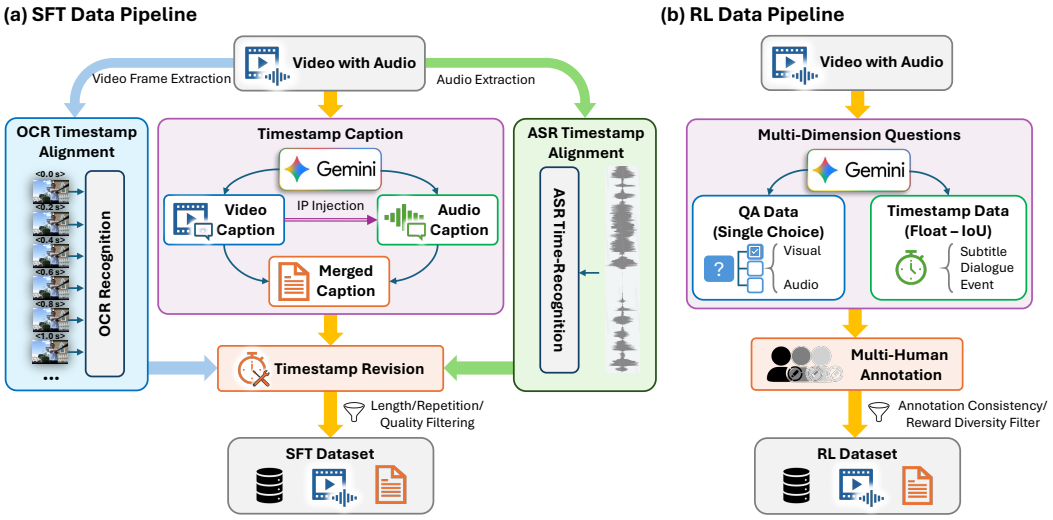


Figure 3: The data construction pipelines for Deci-Omni-Captioner. (a) SFT Data Pipeline: Initial captions from Gemini are refined via an explicit anchoring mechanism, using OCR and ASR to strictly calibrate visual and audio timestamps. (b) RL Data Pipeline: For each video, Gemini synthesizes multiple single-choice QA pairs and timestamp-centric queries, which are then annotated by multiple human annotators and rigorously filtered for consistency.

Step 1: Multi-Dimensional Query Generation. We use Gemini-2.5-Pro to automatically extract two distinct types of data based on the video content. For comprehensive semantic understanding, we generate single-choice question-answering (QA) pairs, creating one visual-centric and one audio-centric QA pair per video. For timestamp grounding, rather than generating explicit questions, we instruct the model to extract short descriptive texts for three atomic-level elements per video: a specific subtitle string, an isolated spoken sentence, and a distinct visual action event. These raw textual descriptions serve as the targets for the subsequent grounding task.

Step 2: Independent Human Annotation. The synthesized single-choice QA pairs and the extracted temporal descriptions are then distributed to three professional human annotators. For the temporal descriptions, the annotators are tasked with labeling the exact start and end times for each described element. To ensure strict objectivity, the three annotators work completely independently and are blinded to each other's labeling results throughout the process.

Step 3: Stringent Consensus Filtering and Reward Verification. We apply strict retention criteria based on inter-annotator agreement. Semantic QA pairs are retained only if all three annotators achieve absolute consensus. For timestamp annotations, samples are preserved only if the maximum temporal discrepancy among annotators falls below predefined thresholds. Recognizing the varying degrees of temporal ambiguity across modalities, we set distinct tolerance thresholds (e.g., stricter for text/speech, relaxed for behavioral events), as detailed in Appendix A.1.3 (Fig. 7).

To ensure the RL signal is highly discriminative, we further verify these samples using our SFT-trained model. Specifically, the model generates 8 candidate captions per video via temperature sampling. These captions are then provided as the sole textual context to an evaluator model (Doubao-Seed-2.0-Pro), which acts as a proxy to answer the QA pairs and extract timestamps. We compute a binary reward for semantic QA and a temporal Intersection over Union (tIoU) reward for timestamp tasks by comparing the evaluator's output against the human ground truth. We filter out trivial or unsolvable samples where candidate captions yield identical rewards across all 8 runs. This rigorous pipeline yields a final RL dataset of 1K semantic QA samples (1 visual and 1 audio query each) and 1K timestamp samples (3 queries each for subtitles, dialogue, and events).

3.3 RL with Span-Specific Credit Assignment

As discussed in Sec. 2, while online RL is increasingly adopted for multimodal alignment [30, 31, 32], the core difficulty lies in multi-reward credit assignment. Unlike math or code where a single

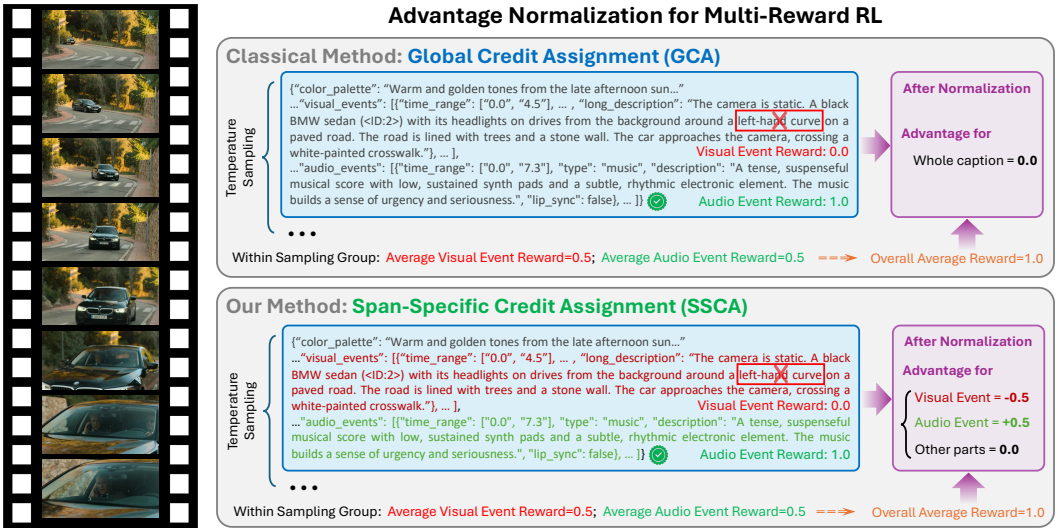


Figure 4: Advantage Normalization strategies in Multi-Reward RL. The Classical Method (GCA) distributes a uniform advantage across all tokens, causing local errors to dilute the overall reward. In contrast, our SSCA calculates advantages independently for distinct structural spans, isolating local penalties from rewards and significantly improving the resolution of the RL training signal.

reasoning trajectory naturally couples all tokens to an outcome-level reward [35], caption generation produces weakly coupled segments for distinct visual and auditory events. This loose dependency means different spans within the same caption can vary independently in quality.

Conventional sequence-level objectives, such as GRPO [37], ignore this independence by calculating a global advantage for the entire response. Given a set of G responses $\{y_1, y_2, \ldots, y_G\}$ sampled for a prompt x, the classical global advantage $A_{\text{global}}(y_i)$ is computed by normalizing the sum of its K individual rewards:

$$A_{\text{global}}(y_i) = \frac{\sum_{k=1}^{K} r_k(x, y_i) - \mu}{\sigma}, \tag{1}$$

where μ and σ are the mean and standard deviation of the summed rewards. Consequently, as shown in Fig. 4 (Classical Method), a single local error (e.g., a hallucinated visual description) pulls down the global score, unfairly penalizing correct segments elsewhere in the same caption.

To address this, we introduce SSCA, a fine-grained advantage normalization strategy. Our core insight is that each reward evaluating a distinct semantic or temporal dimension can be traced back to its specific textual span $S_k(y_i)$ within the generated caption y_i. Instead of aggregating rewards, we perform normalization independently for each reward k, computing a span-specific advantage $A_k(y_i)$ for its corresponding tokens:

$$A_k(y_i) = \frac{r_k(x, y_i) - \mu_k}{\sigma_k}, \quad \text{applied to tokens } j \in S_k(y_i), \tag{2}$$

where μ_k and σ_k are the mean and standard deviation of the k-th reward within the current group.

For the remaining "background" tokens not covered by any specific reward span $S_k(y_i)$, we assign a default global advantage $A_{\text{global}}(y_i)$, identical to the classical method. Thus, the advantage $A^{(j)}(y_i)$ for the j-th token in y_i is formulated as:

$$A^{(j)}(y_i) = \begin{cases} A_k(y_i) & \text{if } j \in S_k(y_i) \text{ for some } k, \\ A_{\text{global}}(y_i) & \text{otherwise.} \end{cases} \tag{3}$$

As illustrated in Fig. 4 (Our Method), this approach isolates the credit and penalty signals. A penalty for a visual hallucination is strictly applied to the visual span, while an accurate audio description simultaneously receives a positive advantage. Our experiments (Fig. 5) confirm that this span-specific supervision significantly improves the resolution of the learning signal.

Implementation and scope. SSCA is applied to structured JSON rollouts, with each sample activating either the semantic-QA or timestamp reward group. Visual and audio QA rewards map to the complete `visual_events` and `audio_events` sections; Subtitle, Dialogue, and Event rewards map to the single queried event in `overlay_events`, `audio_events`, and `visual_events`, respectively, as identified by the evaluator. Character boundaries are converted to token boundaries before assigning $A_k(y_i)$. Because the schema sections are disjoint, attributed spans do not overlap. Absent, malformed, or unmatched targets receive no span attribution, so their tokens and all other background tokens fall back to $A_{\text{global}}(y_i)$ rather than being masked. SSCA changes only the per-token advantage tensor, leaving the policy objective's importance-sampling and clipping terms unchanged; it is compatible in principle with GRPO [37], DAPO [38], and GSPO [14], but is validated here only with GSPO. Appendix B.4 reports token coverage, and Table 12 shows that JSON-trained SSCA gains transfer to NL inference.

4 Experimental Results

4.1 Experimental Setup

Baselines. To comprehensively evaluate our Deci-Omni-Captioner, we select a diverse set of proprietary and open-source models for comparison. For proprietary models, we include Google's latest multimodal suite: Gemini-2.5-Flash, Gemini-2.5-Pro [12], Gemini-3.0-Flash, and Gemini-3.1-Pro [13]. For timestamp evaluation, we test both decimal-second and clock-style (`MM:SS.ss`) timestamp representations and report each Gemini model using the format that yields stronger grounding performance, with all other settings fixed (Appendix B.1). For open-source VL models, we compare against MiMo-VL-7B-RL [44], Qwen2.5-VL-7B-Instruct [21], Qwen3-VL-8B-Instruct, and Qwen3-VL-30B-A3B-Instruct [22]. Although these models lack native audio perception, they are included because they typically exhibit significantly stronger timestamp grounding capabilities than current omni-modal models. To assess joint audio-visual timestamp alignment, we evaluate against SOTA open-source omni models, including Qwen2.5-Omni-7B [15], AVoCaDO [19], UGC-VideoCaptioner [18], Video-SALMONN2+7B [17], as well as Qwen3-Omni-Instruct and Qwen3-Omni-Captioner [16].

Benchmarks. Existing temporal grounding benchmarks primarily target long-form videos and are not well-suited for fine-grained, sub-second alignment. To address this, we construct and release the first **Deci-Timestamp Bench**. This benchmark comprises 400 videos and 747 timestamp annotations. It systematically assesses models across three atomic-level dimensions: Subtitle (267 annotations), Dialogue (256 annotations), and Event (224 annotations). All test samples in the Deci-Timestamp Bench have undergone the same rigorous human annotation and consensus filtering process as our RL data pipeline to ensure high quality and objectivity. Furthermore, to evaluate semantic understanding performance, we benchmark models across four public test sets. These include three omni-modal benchmarks for joint audio-visual understanding: AVUT [45], DailyOmni [46], and UGC [18]; and one VL benchmark: VCapsBench [47], which focuses exclusively on vision-language captioning quality. Since our Captioner is specifically designed to facilitate short-form video generation, we filter the original public benchmarks to retain only videos shorter than 35.0 seconds. Detailed information regarding the evaluation procedures and metrics is provided in Appendix A.3.1.

4.2 Experimental Results

Timestamp Grounding Performance. To evaluate timestamp grounding, we first prompt each model to generate a comprehensive timestamped caption for a given video (see Appendix A.3.2 for exact prompts). We then use an LLM to extract the predicted time span $P = [s_p, e_p]$ corresponding to a specific target description from the Deci-Timestamp Bench. By comparing this prediction against the human-annotated ground truth span $G = [s_g, e_g]$, we calculate the temporal Intersection over Union as $\text{tIoU} = \frac{|P \cap G|}{|P \cup G|}$. If a model fails to generate the target description entirely, its tIoU is strictly penalized to 0. We also report Recall at threshold τ (R1@τ), which measures the percentage of predictions where $\text{tIoU} \geq \tau$ (for $\tau \in \{0.5, 0.7\}$). Furthermore, to assess absolute temporal precision, we compute the mean absolute time error $\Delta t = \frac{|s_p - s_g| + |e_p - e_g|}{2}$ (in seconds), calculated exclusively over non-zero-overlap targets ($\text{tIoU} > 0$). Since assigning zero to an unmatched target couples target coverage with temporal localization, Appendix B.2 additionally reports per-track match rates and

Table 1: Timestamp grounding performance on the Deci-Timestamp Bench across the Event, Dialogue, and Subtitle dimensions. **Best** and second-best results among open-source models are highlighted in bold and underlined, respectively. Dialogue metrics are not reported for VL-only models due to their lack of audio perception.

Model	FPS / TPF	Event				Dialogue				Subtitle			
		R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓
Proprietary Models													
Gemini-2.5-Flash	10/-	33.5	19.6	32.0	1.40	53.9	35.5	47.5	0.60	34.8	18.7	30.5	0.60
Gemini-2.5-Pro	10/-	54.5	37.1	51.9	1.44	77.3	58.6	68.1	0.57	70.8	41.2	58.4	0.41
Gemini-3.0-Flash	10/-	38.4	19.2	36.7	1.32	52.7	31.6	46.9	0.74	46.8	24.3	40.8	0.54
Gemini-3.1-Pro	10/-	45.5	26.8	43.1	1.46	52.3	31.2	46.2	0.94	52.8	26.6	46.3	0.50
Open-source VL Models													
MiMo-VL-7B-RL	10/200	30.8	16.1	32.4	2.03	-	-	-	-	29.2	12.7	33.2	1.59
Qwen2.5-VL-7B-Instruct	10/200	21.0	10.3	22.8	2.92	-	-	-	-	3.4	1.1	9.3	5.02
Qwen3-VL-8B-Instruct	10/200	47.3	29.9	43.0	1.24	-	-	-	-	52.8	34.5	48.0	1.69
Qwen3-VL-30B-A3B-Instruct	10/200	55.6	37.5	50.8	1.31	-	-	-	-	62.2	37.8	54.1	0.78
Open-source Omni Models													
Qwen2.5-Omni-7B	2/500	21.4	8.0	21.9	1.89	8.6	4.3	15.8	2.52	7.9	3.4	14.5	2.06
AVoCaDO	2/500	12.1	6.2	12.1	2.22	5.5	2.0	8.7	1.94	6.4	1.9	9.4	1.19
UGC-VideoCaptioner	2/500	22.3	12.1	21.5	4.64	8.2	3.1	12.7	3.05	4.5	2.2	9.8	2.80
Video-SALMONN2+7B	2/500	19.6	11.6	22.4	4.50	5.1	1.6	11.5	4.28	3.0	0.7	8.8	4.30
Qwen3-Omni-Instruct	2/500	37.1	19.6	36.4	1.91	18.8	10.5	27.7	2.69	23.6	9.4	27.3	1.99
Qwen3-Omni-Captioner	2/500	33.9	20.1	34.5	1.63	20.7	8.6	28.1	2.77	24.7	8.2	28.2	2.13
Deci-Omni-Captioner (SFT)	10/200	55.8	40.6	50.4	1.26	88.3	75.0	78.4	0.26	92.5	82.0	80.4	0.11
Deci-Omni-Captioner (RL)	10/200	**56.7**	**42.9**	**51.1**	**1.22**	**94.9**	**89.8**	**85.5**	**0.12**	**93.3**	**87.3**	**83.1**	**0.10**

Format note. For a format-fair comparison, Gemini-2.5-Pro uses decimal-second timestamps, while Gemini-2.5-Flash, Gemini-3.0-Flash, and Gemini-3.1-Pro use the clock-style `MM:SS.ss` format. Appendix B.1 reports both formats.

recomputes tIoU and Δt after conditioning on non-zero-overlap targets. For a fair comparison, we configure all baseline open-source Omni models at their default training resolution of 2 FPS with 500 tokens per frame (TPF), as higher frame rates cause severe performance degradation (detailed in Appendix B.7). In contrast, we evaluate all VL models and our Deci-Omni-Captioner at 10 FPS with 200 TPF, allowing them to fully utilize their fine-grained temporal perception.

As shown in Table 1, Deci-Omni-Captioner achieves the strongest overall performance among open-source models across all three tracks, with RL post-training further enhancing timestamp precision. On the *Dialogue* and *Subtitle* tasks, our model achieves near-perfect alignment, yielding a minimal error of $\Delta t \approx 0.1$s and R1@0.7 scores of $\sim$90% and $\sim$87%, respectively. Against proprietary models, it establishes a clear lead on these two boundary-explicit tracks and remains competitive on *Event*: compared with Gemini-2.5-Pro, it attains lower Event boundary error and higher R1@0.5 and R1@0.7, while their Event tIoU values are close (51.1 vs. 51.9). Under the non-zero-overlap protocol, it retains the highest tIoU among open-source models on all three tracks.

Overall performance on *Event* remains lower across all models because physical behaviors have semantically ambiguous transitions, unlike the more deterministic boundaries of speech and text. This observation is also consistent with the higher inter-annotator variance observed during human labeling (cf. Appendix A.1.3).

Semantic Captioning Performance. Following the methodology in AVoCaDO [19], we repurpose the AVUT and DailyOmni VQA tasks to evaluate general semantic understanding and caption quality. Specifically, each model first generates a video caption, which is then provided as the sole input to an evaluator LLM to answer the original VQA question. UGC and VCapsBench are evaluated as standard captioning benchmarks. Importantly, since many leading open-source models construct their training data by distilling from Gemini [19, 18], using Gemini as a judge would inherently favor its own derivatives. We therefore use Doubao-Seed-2.0-Pro as the evaluator for the reported results, and verify ranking stability with Claude-Sonnet-4.6 and GPT-5.5 on AVUT, DailyOmni, and UGC (Appendix B.3). Furthermore, unlike the timestamp evaluation, we infer all baseline models using their official recommended prompts to maximize semantic performance. All other environmental settings (e.g., FPS and TPF) remain identical to the timestamp experiments.

As shown in Table 2, the Deci-Omni-Captioner (SFT) already establishes a highly competitive baseline, demonstrating superior or comparable performance against SOTA open-source Omni models. With our span-specific RL post-training, the Deci-Omni-Captioner (RL) further enhances semantic richness and accuracy. Specifically, it achieves the highest overall scores among open-source models on AVUT, UGC, and VCapsBench, while delivering highly competitive results on DailyOmni (second only to AVoCaDO). More impressively, our model bridges the gap with top-tier

Table 2: Semantic captioning performance on public benchmarks. For VCapsBench IR, lower is better (↓); all other metrics are percentages where higher is better.

Model	Omni-Bench								VL-Bench		
	AVUT			DailyOmni	UGC				VCapsBench		
	gemini	human	Overall	Overall	Visual	Audio	Details	Overall	AR	IR ↓	CR
Proprietary Models											
Gemini-2.5-Flash	51.44	57.11	53.11	62.71	78.19	67.13	65.26	70.19	68.47	14.35	79.94
Gemini-2.5-Pro	73.43	72.76	73.24	75.65	82.78	77.50	74.00	78.09	68.32	14.00	79.43
Gemini-3.0-Flash	58.38	66.87	60.87	69.49	80.54	72.16	70.47	74.38	66.90	15.18	78.88
Gemini-3.1-Pro	59.48	63.01	60.51	67.49	75.30	69.27	64.47	69.67	59.90	14.90	70.39
Open-source Omni Models											
Qwen2.5-Omni-7B	40.61	44.51	41.76	55.01	65.42	41.71	42.86	49.99	33.49	16.87	40.29
AVoCaDO	62.18	60.57	61.71	**70.42**	71.21	56.88	50.29	59.46	53.87	16.86	64.80
UGC-VideoCaptioner	43.49	45.93	44.21	54.70	66.41	47.74	43.55	52.56	44.74	16.51	53.59
Video-SALMONN2+7B	52.12	53.25	52.45	61.63	67.23	49.02	44.42	53.55	41.61	16.86	50.05
Qwen3-Omni-Instruct	42.72	50.41	44.98	55.93	70.32	44.57	46.67	53.85	46.51	**14.98**	54.71
Qwen3-Omni-Captioner	46.19	54.27	48.57	58.86	69.40	42.94	46.56	52.97	53.61	16.28	64.04
Deci-Omni-Captioner (SFT)	65.84	67.29	66.26	66.41	78.39	74.00	66.54	72.97	63.69	15.21	75.11
Deci-Omni-Captioner (RL)	**67.77**	**68.50**	**67.98**	69.03	**78.65**	**74.30**	**67.36**	**73.43**	**64.03**	15.50	**75.78**

proprietary systems, consistently outperforming Gemini-2.5-Flash, Gemini-3.0-Flash, and Gemini-3.1-Pro across multiple subsets, and demonstrating comprehensive audio-visual understanding that rivals Gemini-2.5-Pro.

4.3 Ablation Studies

Impact of Adding Timestamp Tokens. As discussed in Section 3.1, incorporating absolute time is critical for sub-second grounding. Table 3 ablates different architectural strategies for temporal injection. The "No Change" baseline relies solely on the backbone's implicit TM-RoPE, yielding the poorest grounding accuracy. Rendering timestamps directly onto video frames ("Add Timestamp on Frames") offers only marginal gains, as it is bottlenecked by the visual encoder's OCR limits and added visual noise. In contrast, prepending explicit timestamp tokens to each sensory segment ("Add Timestamp Tokens") reframes alignment as a direct textual conditioning problem. This strategy achieves the lowest Δt and highest tIoU across all dimensions, confirming its superiority for precise multimodal synchronization.

Table 3: Ablation of explicit absolute time injection strategies.

Model	Event		Dialogue		Subtitle	
	tIoU↑	Δt (s)↓	tIoU↑	Δt (s)↓	tIoU↑	Δt (s)↓
Add Timestamp Tokens	50.4	1.26	78.4	0.26	80.4	0.11
Add Timestamp on Frames	47.9	1.23	74.3	0.32	72.5	0.20
No Change	44.5	1.41	76.1	0.31	59.4	0.32

Impact of Timestamp Alignment Mechanism. To understand the contribution of our visual and audio timestamp calibration, we train identical base models on different SFT data variants (Table 4). Without explicit calibration ("Not Aligned"), the model relies solely on Gemini's raw predictions, yielding the highest absolute time errors and lowest tIoU scores. Incorporating ASR calibration ("ASR Aligned") exclusively enhances Dialogue localization, raising its tIoU from 65.9% to 78.8% without affecting Subtitle performance. Conversely, introducing only OCR calibration ("OCR Aligned") significantly boosts Subtitle tIoU from 65.3% to 79.7%, but leaves Dialogue grounding weak. When both mechanisms are combined ("ASR + OCR Aligned"), the model seamlessly integrates high-precision temporal anchoring for both text and speech. Furthermore, this dual-alignment synergy propagates to general action grounding, pushing Event tIoU to its peak at 50.4%. This demonstrates that precise, deterministic anchors facilitate better global temporal interpolation for blurry behavioral events.

Table 4: Ablation of explicit timestamp alignment during the SFT data construction phase.

Model	Event		Dialogue		Subtitle	
	tIoU↑	Δt (s)↓	tIoU↑	Δt (s)↓	tIoU↑	Δt (s)↓
ASR + OCR Aligned	50.4	1.26	78.4	0.26	80.4	0.11
OCR Aligned	48.9	1.33	68.3	0.36	79.7	0.13
ASR Aligned	49.6	1.29	78.8	0.23	63.1	0.36
Not Aligned	47.9	1.34	65.9	0.41	65.3	0.32

Impact of Credit Assignment Mechanism. To validate SSCA's effectiveness in resolving reward dilution, we compare Deci-Omni-Captioner fine-tuned with SSCA (+SSCA-GSPO) against the Classical Method (+GCA-GSPO). As shown in Fig. 5, SSCA consistently outperforms GCA across both domains. On semantic benchmarks, SSCA's fine-grained credit assignment prevents an accurate visual description from being unfairly penalized by an audio hallucination (or vice versa) within the same caption, yielding superior scores (e.g., 67.98% vs. 66.37% on AVUT). Similarly, on timestamp benchmarks, SSCA isolates the rewards for distinct temporal elements. This prevents a poor event

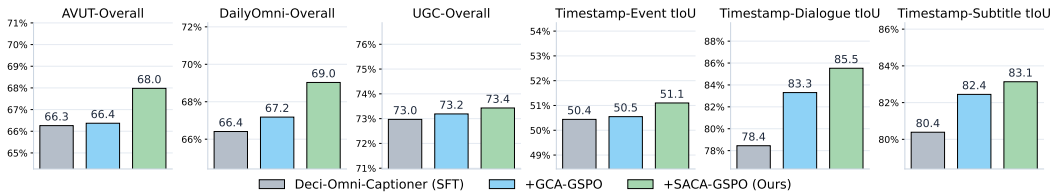


Figure 5: Performance comparison of advantage calculation methods under the GSPO framework. Our SSCA consistently outperforms the Classical Method (GCA) across both semantic benchmarks (left) and timestamp grounding dimensions (right).

localization from degrading an accurate text or speech timestamp, boosting tIoU for Subtitle and Dialogue to 83.14% and 85.52% (vs. 82.45% and 83.30% for GCA). Even on the challenging Event task, SSCA maintains a clear advantage (51.10% vs. 50.55%). This confirms that isolating span-specific penalties significantly enhances overall omni-modal captioning performance.

5 Conclusion

In this paper, we introduced **Deci-Omni-Captioner**, a novel open-source omni-modal captioning framework that addresses the lack of fine-grained timestamp grounding in existing models. By incorporating explicit timestamp tokens and processing high-FPS inputs during SFT, we leveraged OCR and ASR to establish deterministic temporal anchors for speech and on-screen text while improving localization of semantically ambiguous visual events. Furthermore, to overcome the reward dilution problem inherent in global sequence-level optimization, we proposed SSCA for the RL stage. SSCA isolates penalties and rewards across distinct structural spans, ensuring highly efficient and discriminative multi-reward learning. Extensive experiments demonstrate that Deci-Omni-Captioner establishes new SOTA performance among open-source models on multiple semantic benchmarks. On Deci-Timestamp Bench, it achieves tIoU scores of 85.5% and 83.1% with Δt of 0.12s and 0.10s on Dialogue and Subtitle, respectively, while attaining competitive Event grounding with 51.1% tIoU and 1.22s Δt. With its flexible dual-format outputs, Deci-Omni-Captioner provides a powerful solution for large-scale annotation, paving the way for more precise and controllable video generation.

References

[1] Yu Gao, Haoyuan Guo, Tuyen Hoang, Weilin Huang, Lu Jiang, Fangyuan Kong, Huixia Li, Jiashi Li, Liang Li, Xiaojie Li, et al. Seedance 1.0: Exploring the boundaries of video generation models. arXiv preprint arXiv:2506.09113, 2025.

[2] Team Seedance, Heyi Chen, Siyan Chen, Xin Chen, Yanfei Chen, Ying Chen, Zhuo Chen, Feng Cheng, Tianheng Cheng, Xinqi Cheng, et al. Seedance 1.5 pro: A native audio-visual joint generation foundation model. arXiv preprint arXiv:2512.13507, 2025.

[3] Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. arXiv preprint arXiv:2604.14148, 2026.

[4] Kling Team, Jialu Chen, Yuanzheng Ci, Xiangyu Du, Zipeng Feng, Kun Gai, Sainan Guo, Feng Han, Jingbin He, Kang He, et al. Kling-omni technical report. arXiv preprint arXiv:2512.16776, 2025.

[5] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 7346–7356, October 2023.

[6] Runway. Introducing runway Gen-4.5: A new frontier for video generation. `https://runwayml.com/research/introducing-runway-gen-4.5`, 2025.

[7] Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. arXiv preprint arXiv:2412.03603, 2024.

[8] Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. arXiv preprint arXiv:2511.18870, 2025.

[9] Google DeepMind. Veo. `https://deepmind.google/models/veo/`, 2026.

[10] OpenAI. Sora 2 system card. `https://openai.com/index/sora-2-system-card/`, 2025.

[11] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314, 2025.

[12] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261, 2025.

[13] Google DeepMind. Gemini 3. `https://deepmind.google/models/gemini/`, 2026.

[14] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. arXiv preprint arXiv:2507.18071, 2025.

[15] Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. arXiv preprint arXiv:2503.20215, 2025.

[16] Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-omni technical report. arXiv preprint arXiv:2509.17765, 2025.

[17] Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zejun Ma, and Chao Zhang. video-SALMONN 2: Caption-enhanced audio-visual large language models. arXiv preprint arXiv:2506.15220, 2025.

[18] Peiran Wu, Yunze Liu, Zhengdong Zhu, Enmin Zhou, and Junxiao Shen. UGC-VideoCaptioner: An omni UGC video detail caption model and new benchmarks. arXiv preprint arXiv:2507.11336, 2025.

[19] Xinlong Chen, Yue Ding, Weihong Lin, Jingyun Hua, Linli Yao, Yang Shi, Bozhou Li, Qiang Liu, Yuanxing Zhang, Pengfei Wan, et al. AVoCaDO: An audiovisual video captioner driven by temporal orchestration. In The Fourteenth International Conference on Learning Representations, 2026.

[20] Ziyang Ma, Ruiyang Xu, Zhenghao Xing, Yunfei Chu, Yuxuan Wang, Jinzheng He, Jin Xu, Pheng-Ann Heng, Kai Yu, Junyang Lin, et al. Omni-captioner: Data pipeline, models, and benchmark for omni detailed perception. In The Fourteenth International Conference on Learning Representations, 2026.

[21] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923, 2025.

[22] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. arXiv preprint arXiv:2511.21631, 2025.

[23] Antoine Yang, Arsha Nagrani, Paul Hongsuck Seo, Antoine Miech, Jordi Pont-Tuset, Ivan Laptev, Josef Sivic, and Cordelia Schmid. Vid2Seq: Large-scale pretraining of a visual language model for dense video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10714–10726, 2023.

[24] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18243–18252, 2024.

[25] Md Mohaiminul Islam, Ngan Ho, Xitong Yang, Tushar Nagarajan, Lorenzo Torresani, and Gedas Bertasius. Video recap: Recursive captioning of hour-long videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 18198–18208, 2024.

[26] Wei-Yuan Cheng, Kai-Po Chang, Chi-Pin Huang, Fu-En Yang, and Yu-Chiang Frank Wang. TA-prompting: Enhancing video large language models for dense video captioning via temporal anchors. arXiv preprint arXiv:2601.02908, 2026.

[27] Changli Tang, Yixuan Li, Yudong Yang, Jimin Zhuang, Guangzhi Sun, Wei Li, Zujun Ma, and Chao Zhang. Enhancing multimodal LLM for detailed and accurate video captioning using multi-round preference optimization. arXiv preprint arXiv:2410.06682, 2024.

[28] Jiyang Tang, Hengyi Li, Yifan Du, and Wayne Xin Zhao. AVC-DPO: Aligned video captioning via direct preference optimization. arXiv preprint arXiv:2507.01492, 2025.

[29] Ruohong Zhang, Liangke Gui, Zhiqing Sun, Yihao Feng, Keyang Xu, Yuanhan Zhang, Di Fu, Chunyuan Li, Alexander G Hauptmann, Yonatan Bisk, et al. Direct preference optimization of video large multimodal models from language model reward. In Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 694–717, 2025.

[30] Maksim Dzabraev, Alexander Kunitsyn, and Andrei Ivaniuta. VLRM: Vision-language models act as reward models for image captioning. arXiv preprint arXiv:2404.01911, 2024.

[31] Desen Meng, Rui Huang, Zhilin Dai, Xinhao Li, Yifan Xu, Jun Zhang, Zhenpeng Huang, Meng Zhang, Lingshu Zhang, Yi Liu, et al. Videocap-R1: Enhancing mllms for video captioning via structured thinking. arXiv preprint arXiv:2506.01725, 2025.

[32] Long Xing, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jianze Liang, Qidong Huang, Jiaqi Wang, Feng Wu, and Dahua Lin. CapRL: Stimulating dense image caption capabilities via reinforcement learning. arXiv preprint arXiv:2509.22647, 2025.

[33] Zhijiang Tang, Linhua Wang, Jiaxin Qi, Weihao Jiang, Peng Hou, Anxiang Zeng, and Jianqiang Huang. Cccaption: Dual-reward reinforcement learning for complete and correct image captioning. arXiv preprint arXiv:2602.21655, 2026.

[34] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In International Conference on Machine Learning, pages 10818–10838. PMLR, 2025.

[35] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025.

[36] Yanlin Wang, Yanli Wang, Daya Guo, Jiachi Chen, Ruikai Zhang, Yuchi Ma, and Zibin Zheng. Rlcoder: Reinforcement learning for repository-level code completion. In 2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE), pages 1140–1152. IEEE, 2025.

[37] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.

[38] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In The Thirty-ninth Annual Conference on Neural Information Processing Systems, 2026.

[39] Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. arXiv preprint arXiv:2511.20347, 2025.

[40] Biao Yang, Bin Wen, Boyang Ding, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai Keye-VL 1.5 technical report. arXiv preprint arXiv:2509.01563, 2025.

[41] Jun Zhang, Teng Wang, Yuying Ge, Yixiao Ge, Xinhao Li, Ying Shan, and Limin Wang. Timelens: Rethinking video temporal grounding with multimodal LLMs. arXiv preprint arXiv:2512.14698, 2025.

[42] Hunyuan Vision Team, Pengyuan Lyu, Xingyu Wan, Gengluo Li, Shangpin Peng, Weinong Wang, Liang Wu, Huawen Shen, Yu Zhou, Canhui Tang, et al. HunyuanOCR technical report. arXiv preprint arXiv:2511.19575, 2025.

[43] Ye Bai, Jingping Chen, Jitong Chen, Wei Chen, Zhuo Chen, Chuang Ding, Linhao Dong, Qianqian Dong, Yujiao Du, Kepan Gao, et al. Seed-ASR: Understanding diverse speech and contexts with LLM-based speech recognition. arXiv preprint arXiv:2407.04675, 2024.

[44] Xiaomi MiMo Team. MiMo-VL technical report. arXiv preprint arXiv:2506.03569, 2025.

[45] Yudong Yang, Jimin Zhuang, Guangzhi Sun, Changli Tang, Yixuan Li, Peihan Li, Yifan Jiang, Wei Li, Zejun Ma, and Chao Zhang. Audio-centric video understanding benchmark without text shortcut. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 6580–6598, 2025.

[46] Ziwei Zhou, Rui Wang, Zuxuan Wu, and Yu-Gang Jiang. Daily-Omni: Towards audio-visual reasoning with temporal alignment across modalities. arXiv preprint arXiv:2505.17862, 2025.

[47] Shi-Xue Zhang, Hongfa Wang, Duojun Huang, Xin Li, Xiaobin Zhu, and Xu-Cheng Yin. Vcapsbench: A large-scale fine-grained benchmark for video caption quality evaluation. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 12726–12734, 2026.

[48] Yuze Zhao, Jintao Huang, Jinghan Hu, Xingjun Wang, Yunlin Mao, Daoze Zhang, Zeyinzi Jiang, Zhikai Wu, Baole Ai, Ang Wang, Wenmeng Zhou, and Yingda Chen. SWIFT: A scalable lightweight infrastructure for fine-tuning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 39, pages 29733–29735, 2025.

A Implementation Details

A.1 Training Data Construction

A.1.1 Data Distribution

Our 110K videos for SFT are curated from an internal proprietary collection of high-quality video clips, encompassing diverse sources such as movies, TV series, vlogs, and short-form videos. The original dataset comprised approximately 270K raw videos. As illustrated in the quality filtering stage of Fig. 3(a), these samples underwent a rigorous screening process to eliminate those with parsing errors, semantic repetitions, or inappropriate lengths. We subsequently performed distribution-aware sampling on the remaining candidates to derive the final 110K high-quality dataset.

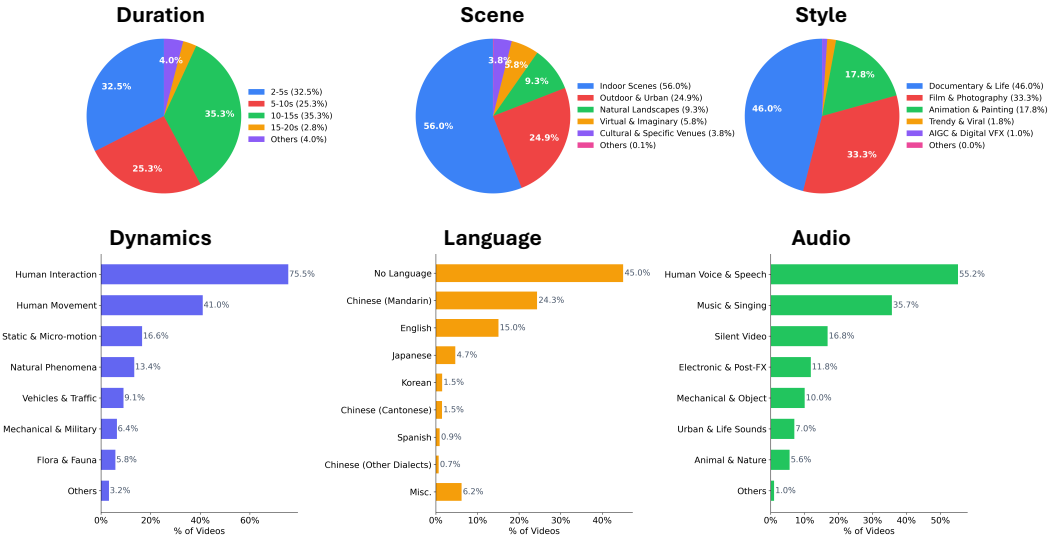


Figure 6: Data distribution of the 110K videos for SFT across multiple dimensions. We systematically analyze the distribution of video **Duration** (predominantly 2-15s, tailored for video generation), visual **Scene** (environmental settings), production **Style** (visual genres), and subject **Dynamics** (types of intra-frame motion and action). For the audio track, we report the distribution of spoken **Language** and overarching **Audio** categories (e.g., speech, music, acoustic events).

Since Deci-Omni-Captioner is primarily tailored to serve as the annotation foundation for video generation models, which typically produce clips ranging from 5 to 15 seconds, we intentionally biased our sampling to ensure that the vast majority of our captions correspond to short video durations, predominantly concentrated in the 2-15s range. Furthermore, to ensure comprehensive omni-modal coverage, we meticulously balanced the data distribution across multiple visual and auditory dimensions. On the visual front, the dataset covers a wide spectrum of environmental scenes, production styles, and subject dynamics. On the auditory front, it encompasses diverse spoken languages and a broad array of acoustic events. The detailed distribution of the final 110K videos for SFT is visualized in Fig. 6.

A.1.2 Prompt Templates for Coarse Caption Generation

To generate the initial high-quality, structured JSON captions for our 110K videos for SFT, we employ a two-stage prompting strategy using Gemini-2.5-Pro. This two-stage strategy effectively reduces the cognitive load on the model and ensures that the auditory description accurately references the visual context.

In the first stage, we input the raw video alongside the visual prompt to instruct the model to parse the visual stream into a structured JSON containing a detailed entity registry, a visual action track, and an overlay track. In the second stage, we feed the video, the audio prompt, and the previously generated visual JSON context back into the model. This allows the model to act as a "Film & TV

Audio Alignment Expert," utilizing the established visual IDs to eliminate auditory ambiguity (e.g., matching a voice to a specific visible speaker) and generate a highly precise audio track.

It is worth noting that we also adopt this same structured prompt for the Gemini series models across the benchmark evaluations in this paper, changing only the requested timestamp representation in the format ablation of Section B.1. While the prompt is highly complex, Gemini models exhibit exceptional instruction-following capabilities capable of handling it, whereas other baseline models struggle with such complexity. Furthermore, we specifically selected Gemini-2.5-Pro over the newer Gemini 3.x series for data generation because, even under each model's preferred timestamp format, the 3.x models remain weaker than Gemini-2.5-Pro in fine-grained timestamp grounding.

The exact prompts used in this two-stage pipeline are presented below.

Stage 1: Prompt for Visual and Overlay Track Generation

```
### Role & Goal
You are a **Senior Multimodal Data Engineer**. Your sole task is to parse an input **Video** into a **JSON object**
      that strictly adheres to the defined Schema, containing parallel Visual and Overlay tracks.

### Core Principles
1. **Separation of Physical and Digital**:
    * **Scene Text**: Belongs to the physical world (e.g., street signs, logos on clothes). Must be described in the
          **Visual Track**.
    * **Overlay**: Belongs to the digital world (e.g., subtitles, watermarks, UI). Must be described in the **Overlay
          Track**.
2. **Absolute Timestamp Precision (0.1s)**:
    * Timestamps for visual and auditory elements must precisely match video frames and waveforms, with a strict error
          margin within 0.1s.
3. **Action Density**:
    * Visual descriptions must contain high-density action details (micro-expressions, body language, interactions).
4. **Sparse Anchoring Strategy**:
    * **Core Entities (VIPs)**: Cross-shot protagonists, key props, and **visible speakers**. Must be anchored using
          **ID (<ID:x>)**.
    * **Background/Minor Entities**: Pure background extras, passersby with no dialogue.
       **Strictly DO NOT assign IDs**. Use natural language full-feature descriptions.

### Input Data
1. **Video File**: A video file.

---

### Chain of Thought (CoT) - [Must be executed internally to determine Caption Quality]

Before generating the JSON, you must perform the following deep **internal narrative analysis** in the background:

1. **Entity Filtering**:
    * *Decision*: Who is the absolute protagonist? What is the key prop? -> **Assign ID**.
    * *Decision (Crucial)*: **Who has lines and is visible?** (To ensure lip_sync logic closure) -> **Assign ID**.
    * *Decision*: Who is just a background board? Who is the blurry crowd shouting in the back? -> **Do NOT Assign ID
          **.

2. **Subject Profiling**:
    * For ID subjects: Establish a global feature library.
    * For non-ID subjects: Extract visual features required for the current shot each time they appear.

3. **World Building**:
    * **Spatial Definition**: Indoor/Outdoor? Specific location (Study/Battlefield/Cyber Street)?
    * **Furnishing Details**: Identify main furniture (mahogany desk/metal shelves), prop status (worn/brand new/
          cluttered).
    * **Lighting & Atmosphere**: Where is the light source (rim light/overhead)? Color temperature (dim yellow/cold
          blue)? What atmosphere does it create?

4. **Temporal Granularity Strategy [Critical Step]**:
    * **Assess Action Density**: Observe the frequency and complexity of actions in the frame.
    * **Decision Branch**:
       * **Case A: High Frequency/Chaos**: If actions change extremely fast ->
          **Adopt "Event-based" granularity**. Merge a series of related actions into a longer Visual Event. The
                description focus is "what event happened".
       * **Case B: Low Frequency/Clear**: If actions are soothing, clear, and have distinct start/end points -> **
             Adopt "Action-based" granularity**. You must precisely decompose every micro-action, with timestamps
             strictly synchronized.

---

### Key Constraints (Absolute Red Lines)

#### 1. Reject Uncertainty
* **Strictly forbidden** to use vague words like "maybe", "seems", "probably", "or".
* If visual features are clear, **assert directly** (e.g., say "a study room").
* If visual features are blurry, **describe only visible elements**.

#### 2. Entity Registration Logic (VIP Sparse Strategy)
```

```
* **Admission Criteria (Assign ID if ANY is met)**:
    1. **Narrative VIP**: The protagonist running through the video narrative.
    2. **Object Persistence**: Key props undergoing drastic physical state changes.
    3. **AV Anchor (Audio-Visual Alignment)**: **Characters clearly speaking (Dialogue) within the frame who require ‘
          lip_sync: true‘ tagging via ID**.
* **Strictly Forbidden (Blacklist)**:
    * **Groups**: e.g., "a group of soldiers".
    * **Pure Background Passersby**: Passersby with no interaction, no dialogue.
* **Content Purity**: The ‘description‘ field in the entity list must ONLY contain
    **inherent visual features** of the entity.

#### 3. Visual Track Narrative Standards
* **ID Reference**: Use ‘<ID:x>‘ ONLY for registered entities.
* **Non-ID Description**: For unregistered entities, **every mention must include a brief visual adjective**.
* **Dual Description**: Must generate both ‘short_description‘ and ‘long_description‘.
* **Scene Text Handling**: If there is text on an object, it must be mentioned in the long description. Wrap text
      content in double quotes.
* **Transition Description**: If special transitions are included, describe the method.

#### 4. Overlay Track Definition (Post-Production)
* Record ONLY **non-physical** post-production overlay elements (subtitles, watermarks).
* **Position Description**: Use natural language directional words.
* Strictly split Overlay Events. **Strictly Forbidden** to merge multiple sequentially appearing Overlay Events into
      the same timestamp.

#### 5. Data Format Specifications
* **Time**: Seconds, keep 1 decimal place. Timestamps precise to 0.1s.
* **Language**: Keep original language for specific text content (OCR) or spoken quotes.
    **All other descriptions must be in English.**

---

### JSON Schema Definition

{
  "color_palette": "string. Describe the dominant tone, color temperature, and saturation.",
  "atmosphere": "string. Describe the emotion or atmosphere conveyed by the video.",
  "environment": "string. Scene environment type and spatial features.",
  "style": "string. Visual aesthetics and style.",
  "light": "string. Lighting type, source, and emotional color.",
  "short_caption": "string. (15-25 words) Minimalist summary. Strictly forbidden to use <ID:x>.",
  "long_caption": "string. (80-150 words) Detailed objective visual narrative. Strictly forbidden to use <ID:x>; use
       natural pronouns.",
  "entities": [
    {
      "id": "<ID:x>", // starts from 1
      "type": "subject | object",
      "description": "Detailed description of constant visual features. Do NOT reference any ‘<ID:x>‘ here.",
    }
  ],
  "visual_events": [
    {
      "time_range": ["0.0", "1.8"],
      "action_density": "high (Event-based) | low (Action-based)",
      "short_description": "string. (e.g., ’’An East Asian woman (<ID:1>) looks at the phone (<ID:2>)’’)",
      "long_description": "string. Detailed description with camera movement and actions."
    }
  ],
  "overlay_events": [
    {
      "time_range": ["0.0", "1.8"],
      "type": "subtitle | title | logo | ui_overlay | vfx_tag",
      "description": "string"
    }
  ]
}
```

Stage 2: Prompt for Audio Track Generation (with Visual Context Injected)

```
### Role & Goal You are a **Film & TV Audio Alignment Expert**. You are executing the **second step of a multimodal
      annotation pipeline**. Your task is to generate high-precision audio track data based on the input **Video**
      and the pre-generated **Visual Context (Structured Visual Data)**.

### Input Data
1. **Video File**: Video file (containing audio).
2. **Visual Context JSON**: The JSON generated in step one, containing ‘entities‘ (ID definitions) and ‘visual_events
      ‘ (actions/timeline).

### Core Task Utilize the visual information from the Visual Context to eliminate auditory ambiguity and generate the
       **Audio Track**.

### Core Strategy
**Describe sound as if you are writing a novel or a screenplay.**
```

```
* **DO NOT Write**: "Near-field pickup", "Dry sound", "High SNR", "Wide frequency response".
* **DO Write**: "Clear speaking voice", "Very clean", "No echo", "Muffled sound".

### Key Constraints (Absolute Red Lines)
1. **Reject Uncertainty**: Strictly forbidden to use vague words like "maybe", "seems".
2. **Background Layer Priority**: When generating the Audio Track, you **must first check** for the existence of **
      Music** or **Ambience** that covers the entire clip.
3. **Temporal Alignment**: Timestamps must be precise to 0.1s.
4. **Sorting Order**: Recommended order for Audio list: 1. Music/Ambience -> 2. Voice -> 3. SFX.
5. **Language & Accent Fidelity**:
    * Must precisely identify the language/dialect. **Strictly forbidden** to generalize as just "Chinese".
    * **[Transcription Iron Rule]**: The `description` must follow **"Verbatim Transcription"**.
        * *Good Case*: Hearing Cantonese "Why?", write: "Cantonese male voice, the boy (<ID:1>) asks doubtfully: \"Why
              ?\"" (Keep the quote in original language).
6. **Auditory Salience**:
    * **No Hallucination**: If there is an action in the visual but it is **inaudible** in the audio, you are **
          strictly forbidden** from labeling that SFX.
    * **Ignore Weak Noise**: Ignore tiny friction sounds or noise floor.
    * **Merge Principle**: Merge a coherent sentence by the same speaker, or a continuous dense series of similar
          sound effects, into a single Event.
7. **L-Cut/J-Cut Segmentation Rules (Critical)**:
    * **Single Sentence Independence**: You must follow the strategy of **"One Sentence, One Break"**. Maximize the
          isolation of every sentence.
8. **Special Handling for Lip-Sync False**: If a person is speaking but `lip_sync` is `false`, **you must explicitly
      state the reason in the description**.
9. **Lip-Sync Strict Judgment Rules (Anti-Hallucination)**:
    * **Strictly Forbidden to Guess**: `lip_sync` can only be `true` when **ALL** the following conditions are met:
        * The speaker `<ID:x>` is inside the frame.
        * The shot size is **Medium Shot (MS) or closer**.
        * The face is **not obstructed** and **not completely turned away**.
    * **Default Fallback**: If it is **Long Shot (LS/ELS)**, **Back View**, **Blurry**, or **Uncertain**, strictly
          set to `false`.

---

### JSON Schema Output

{
  "audio_summary": "string. (Approx. 100 words. Describe the auditory experience of the entire clip in plain English
        .)",
  "audio_events": [
    {
      "time_range": ["0.0", "1.8"],
      "type": "music | ambience | dialogue | voice_over | vocal | sfx",
      "description": "string. (Description in English, Quote in Original Language)",
      "lip_sync": "boolean (true ONLY if the semantic identity (<ID:x>) speaks within the frame AND lip movements
            match)"
    }
  ]
}
```

It is important to note that the two aforementioned prompts exclusively generate audio-visual captions in a structured JSON format. To accommodate diverse user requirements and downstream applications, we also convert these captions into a free-form Natural Language (NL) format. This is achieved by directly rewriting the structured JSON outputs into coherent paragraphs. For this rewriting process, we continue to employ Gemini-2.5-Pro. The specific prompt used for this transformation is detailed below:

Prompt for Rewriting JSON to Natural Language

```
You are a video description rewriting expert. Rewrite the structured video description below into fluent, natural
      English text.

**Rewriting rules**:

1. **Background paragraph (1st paragraph)**: Keep the atmosphere / environment / lighting / style / color
      descriptions. Add connectors for better flow without changing the meaning.

2. **Integrated narrative paragraph (2nd paragraph)**: Merge [Summary] and [Audio] summary into one coherent
      paragraph -- first summarize what happens visually, then transition with "On the auditory side" / "Sound-wise"
      to describe how the soundtrack complements the narrative. Remove all <ID:x> tags; use descriptive language
      instead.

3. **Timestamped detail paragraphs**:
   - Use [Visual] time ranges as the primary timeline, one paragraph per segment.
   - Naturally weave in temporally overlapping **short** audio events (dialogue, SFX) into the corresponding visual
         paragraph, e.g. "around 5.5 s, the rustling of paper emerges clearly in the foreground".
   - Keep "lip-synced" annotations where present.
   - Remove all <ID:x> tags.
```

```
4. **Full-duration audio at the end**: Audio events spanning nearly the entire video (background music, ambient noise
      ) become separate timestamped paragraphs placed after all visual paragraphs.

5. If [Overlay] content exists (subtitles / UI), mention it naturally in the corresponding timestamped paragraph.

6. Timestamp format: `[start_s-end_s]`.

Output only the rewritten text with no extra explanation or headings.

---

Here is the structured description to rewrite:
```

A.1.3 Statistical Analysis of Human Annotation Consistency

As discussed in Section 3.2.2, although the initial queries and descriptions for the RL stage are synthesized by Gemini, we found that Gemini itself suffers from hallucination issues and lacks precise timestamp grounding capabilities. To ensure the high quality of the RL dataset, we conducted a rigorous manual annotation process involving three independent professional human annotators.

For the single-choice QA pairs, the annotation process was straightforward: a sample was considered valid and retained only if all three annotators independently arrived at the exact same answer. However, for the timestamp grounding data, the annotators were required to label the precise start and end times (as floating-point numbers) for specific events, dialogues, and subtitles. Due to the continuous nature of video time, it is practically impossible for three humans to produce identical floating-point annotations, and subjective bias inherently leads to deviations.

As illustrated in Fig. 7, we observed that the variance in annotations is largest for visual events. This observation aligns with human intuition, as the boundaries of a visual action or event are often blurry and subjective. Conversely, the temporal boundaries for dialogue and subtitles are much clearer and objective, resulting in significantly tighter distributions.

To maintain exceptionally high data quality while accounting for these inherent differences, we calculated the maximum time discrepancy among the three annotators for each sample. We established strict, category-specific tolerance thresholds: samples where the maximum discrepancy exceeded 0.6s for events, 0.2s for dialogue, and 0.1s for subtitles were discarded. For the retained high-quality samples, we computed the mean of the three annotators' start and end times as the final ground truth. Through this rigorous filtering process, we successfully curated the final 2K high-quality RL dataset.

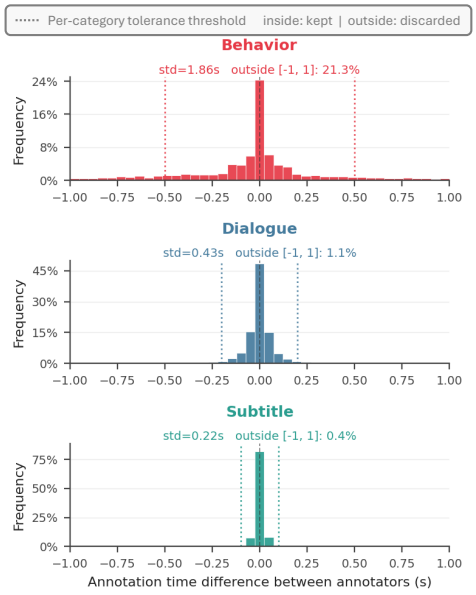


Figure 7: Annotation consistency and tolerance thresholds across three data categories. We plot the maximum annotation time difference among three independent human labelers. Visual events show higher variance due to their inherently blurred temporal boundaries, whereas dialogue and subtitles present much tighter distributions. To maintain high data quality, we apply category-specific tolerance thresholds, discarding samples where the maximum annotator discrepancy exceeds 0.6s, 0.2s, and 0.1s for events, dialogue, and subtitles, respectively.

A.2 Model Training Details

Our model training pipeline leverages Qwen3-Omni-Instruct [16] as the base model. The entire training process is implemented using the `ms-swift` [48] framework and is executed on a distributed cluster comprising 64 $\times$ NVIDIA H20 (140GB) GPUs across 8 nodes.

Supervised Fine-Tuning (SFT) Stage. During the SFT stage, we apply full-parameter tuning, ensuring that the multimodal aligners, the vision encoder, and the audio encoder are all jointly updated. To equip the model with the dual-output capabilities discussed in Appendix B.8, each of the curated 110K SFT videos is accompanied by annotations in both structured JSON and NL formats. This dual-format annotation strategy expands the dataset to 220K samples. Crucially, to enhance the model's robustness to varying temporal resolutions, each sample is processed under two distinct configurations: 2 FPS with 500 TPF, and 10 FPS with 200 TPF. This strategy further expands the dataset to 440K training samples. The model is trained on this combined dataset for 2 epochs with a global batch size of 128. We employ a cosine decay learning rate schedule with a warm-up ratio of 0.01, a maximum and minimum learning rate of 1×10^{-5} and 1×10^{-7}, respectively.

Reinforcement Learning (RL) Stage. In the RL stage, we adopt the GSPO algorithm. The hyperparameters for the clipping mechanism are set to $\epsilon = 3 \times 10^{-4}$ and $\epsilon_{\text{high}} = 4 \times 10^{-4}$. We utilize a cosine decay learning rate schedule with a warm-up ratio of 0.05, a maximum learning rate of 2×10^{-6}, and a minimum learning rate of 1×10^{-7}. The RL batch size is set to 1024, constructed from 128 distinct prompts per batch, with each prompt generating 8 responses. These responses are subsequently used to compute the policy gradient, and the model is updated over 4 micro-steps. We train the model on 2K RL videos, comprising 1K semantic QA samples and 1K timestamp samples, for 5 epochs and randomly shuffle the data before each epoch. Consistent with the evaluation methodology of the Deci-Timestamp Bench, we formulate the reward function as follows: for single-choice QA queries, we use a binary reward (assigning 1 for a correct answer and 0 for an incorrect one); for timestamp queries, we use the tIoU as the continuous reward. Finally, we emphasize that Doubao-Seed-2.0-Pro is utilized as the judge model to evaluate the generated captions, answering the QA queries and extracting the precise timestamps from the model's caption outputs to compute these rewards.

A.3 Evaluation Protocols

A.3.1 Evaluation Procedures and Metrics Details

In this section, we provide detailed descriptions of the evaluation procedures and metrics for the four public benchmarks used in Section 4.2. It is worth noting that this work primarily aims to construct high-quality, audio-visually aligned data to facilitate the training of temporally controllable video generation models. Since the maximum generation duration of current state-of-the-art video generation models (e.g., Sora 2 [10], Veo [9], and Kling [4]) is typically within 30 seconds, we carefully filter the evaluation benchmarks to align with this practical constraint. Specifically, we exclude videos longer than 35 seconds to maintain consistent temporal processing, and we discard videos that do not contain an audio track, as audio-visual consistency is a fundamental prerequisite for modern video generation tasks. The dataset statistics before and after this filtering process are detailed in Table 5.

UGC [18]. UGC is designed for evaluating omni-modal video captioning on short user-generated content. Following our filtering criteria, the evaluation subset consists of 783 TikTok videos, each under 35 seconds in duration and containing meaningful audio segments. Each video's caption is evaluated by a judge model that assigns scores on a 1-to-5 scale across three dimensions: visual, audio, and details. These dimension scores are then normalized and aggregated to produce a final caption quality score.

DailyOmni [46]. DailyOmni is an audio-visual question answering benchmark depicting diverse everyday life scenarios sourced from multiple platforms. These videos are densely multimodal, offering rich visual and auditory cues. After applying our filters, the evaluation subset comprises 377 videos and 649 multiple-choice question-answer pairs, distributed across six core tasks.

AVUT [45]. AVUT is a comprehensive audio-visual video question answering benchmark focusing on fine-grained audio-visual understanding. Our filtered evaluation subset contains 415 videos and 1,674 multiple-choice question-answer pairs covering various complex multimodal reasoning tasks.

VCapsBench [47]. Unlike the other three benchmarks that assess audio-visual understanding, VCapsBench focuses purely on the visual modality. It evaluates a model's ability to generate coherent and semantically rich descriptions of diverse video scenes. For our evaluation, we utilize a filtered subset containing 3,109 videos and 56,898 fine-grained annotations to assess the model's capability in capturing both main events and detailed visual interactions.

Table 5: Statistics of the four public benchmarks before and after our duration and audio-presence filtering. **Before Filter** shows the original video/question counts; **After Filter** shows the subset we use for evaluation.

Benchmark	Before Filter		After Filter	
	#Videos	#Questions	#Videos	#Questions
AVUT	2,655	11,608	415	1,674
DailyOmni	684	1,197	377	649
UGC	1,000	1,000	783	783
VCapsBench	5,666	104,335	3,109	56,898

Evaluation Protocol for VQA Benchmarks. As detailed in Section 4.2, we adapt AVUT and DailyOmni, which were originally designed for VQA, to evaluate caption quality using an LLM-as-a-Judge approach [19]. Specifically, the target model first generates a caption for the video. This generated caption is then provided as the *sole* text input to a judge model (Doubao-Seed-2.0-Pro) to answer the original VQA question. If the caption lacks necessary details or hallucinates, the judge will answer incorrectly. This proxy task effectively measures the density, accuracy, and completeness of the multimodal information captured in the generated captions.

A.3.2 Prompts for Evaluation

When evaluating the four public semantic benchmarks (UGC, DailyOmni, AVUT, and VCapsBench), we aim to maximize the semantic performance of the baseline models. Therefore, we strictly follow their respective official recommended user and system prompts for caption generation, rather than imposing a unified prompt.

Conversely, for the timestamp evaluation on our Deci-Timestamp Bench, it is worth noting that most existing baseline captioning models are not natively trained to output fine-grained timestamps in their generated descriptions. To fairly evaluate and compare their timestamp grounding capabilities, we explicitly instruct all evaluated models to generate timestamped natural-language descriptions using a unified content and output schema. We additionally test the clock-style `MM:SS.ss` representation recommended for Gemini video understanding, changing only the timestamp instruction and examples while retaining the required precision. Table 1 reports the stronger format for each Gemini model, and Section B.1 provides the full ablation. The decimal-second prompt is presented below; the clock-style variant replaces its timestamps with `MM:SS.ss`.

Evaluation Prompt for Deci-Timestamp Bench

```
You are a multimodal video analyst with access to both visual and audio streams. Watch the video carefully and write
      a detailed natural-language description in English. Use timestamps in seconds with 1 decimal place (e.g. "3.2")
       throughout.

**Output format**:

Your description should include the following sections in order:

**1. Summary**: A concise 1-2 sentence overview of the entire video.

**2. Entities**: List the main characters and key objects. For each, assign an ID (e.g. <ID:1>) and describe their
      static visual appearance (clothing, hair, build, etc.). Do NOT assign IDs to background crowds or minor
      elements.

**3. Visual Events**: Describe the video as a sequence of temporal segments. For each segment, clearly state the time
       range (e.g. "From 0.0s to 3.5s, ...") and describe the actions, expressions, spatial layout, and camera
      movement in detail.

**4. Overlay Events**: Describe any non-physical digital overlays visible on screen (subtitles, watermarks, logos, UI
       elements). For each, state the time range and quote the text content. If there are no overlays, write "No
      overlay events."

**5. Audio Events**: Describe all significant audio elements heard in the video, including background music, ambient
      sounds, sound effects, and dialogue. For each, state the time range.
- For dialogue: identify the language, link to the speaker <ID:x> when visible, and transcribe the spoken content
      verbatim in its original language (in quotes). Note whether the speaker's lip movement is clearly visible (
      lip_sync).
- For music/ambience/sfx: describe genre, mood, instruments, or sound characteristics.
```

```
**Example format**:

Summary: A woman in a red dress picks up a coffee cup while soft piano music plays, and she speaks to someone off-
      screen.

Entities:
- <ID:1>: A young woman with long black hair, wearing a red dress.

Visual Events:
- From 0.0s to 3.5s, the woman (<ID:1>) reaches forward and picks up a white coffee cup from a wooden table. She
      brings it to her lips and takes a sip. The camera holds a static medium shot.
- From 3.5s to 6.0s, she sets the cup down and leans back in her chair, smiling softly. The camera slowly zooms in on
       her face.

Overlay Events:
- From 0.0s to 2.0s, a white Chinese subtitle appears at the bottom center: "你好世界".

Audio Events:
- From 0.0s to 8.5s, soft piano background music with a calm, melancholic mood.
- From 1.2s to 3.8s, Mandarin female voice, the woman (<ID:1>) says gently: "你今天怎么了？" (lip_sync: visible).

Now analyze the video and write the description.
```

B Additional Experiments and Analysis

B.1 Impact of Timestamp Representation for Gemini Models

To test whether a unified decimal-second format disadvantages Gemini models, we rerun all four proprietary baselines with clock-style `MM:SS.ss` timestamps. Only the requested timestamp representation changes; the videos, 0.1s precision requirement, frame rate, evaluator, and metrics remain identical.

Table 6: Timestamp-format ablation for Gemini models on Deci-Timestamp Bench. Only the requested timestamp representation changes; videos, frame rate, precision requirement, evaluator, and metrics remain fixed. Blue rows denote the format reported in Table 1.

Model	Format	Event				Dialogue				Subtitle			
		R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓
Gemini-2.5-Flash	seconds	24.1	11.6	26.7	1.74	43.0	22.3	38.8	0.85	25.8	15.0	26.2	0.91
Gemini-2.5-Flash	`MM:SS.ss`	33.5	19.6	32.0	1.40	53.9	35.5	47.5	0.60	34.8	18.7	30.5	0.60
Gemini-2.5-Pro	seconds	54.5	37.1	51.9	1.44	77.3	58.6	68.1	0.57	70.8	41.2	58.4	0.41
Gemini-2.5-Pro	`MM:SS.ss`	45.1	31.7	45.3	1.39	62.9	50.0	57.1	0.70	61.8	38.2	52.0	0.42
Gemini-3.0-Flash	seconds	23.7	13.4	23.6	1.61	19.9	8.2	19.0	0.96	20.6	10.5	18.8	0.78
Gemini-3.0-Flash	`MM:SS.ss`	38.4	19.2	36.7	1.32	52.7	31.6	46.9	0.74	46.8	24.3	40.8	0.54
Gemini-3.1-Pro	seconds	29.0	18.8	28.1	1.69	37.9	21.5	34.0	0.94	30.7	16.9	27.5	0.67
Gemini-3.1-Pro	`MM:SS.ss`	45.5	26.8	43.1	1.46	52.3	31.2	46.2	0.94	52.8	26.6	46.3	0.50

As shown in Table 6, the clock-style format improves tIoU and both recall metrics on every track for Gemini-2.5-Flash, Gemini-3.0-Flash, and Gemini-3.1-Pro. Gemini-2.5-Pro is the exception: decimal seconds yield higher tIoU and recall on all three tracks, despite a negligible change in Event Δt. We therefore report decimal seconds for Gemini-2.5-Pro and `MM:SS.ss` for the other three models in Table 1. This model-specific choice avoids penalizing a baseline through timestamp-format mismatch; under their preferred formats, the Gemini 3.x models still remain behind Gemini-2.5-Pro and Deci-Omni-Captioner across all three tracks.

B.2 Match Rate and Non-Zero-Overlap Timestamp Grounding

The all-sample tIoU and recall metrics in Table 1 jointly measure whether a model describes a target and how accurately it localizes that target, because an unmatched description receives zero. To isolate localization from target coverage, we define a target as matched when its evaluated tIoU is non-zero, report the resulting match rate, and recompute tIoU and Δt over non-zero-overlap targets only.

Table 7: Target match rates (%) on Deci-Timestamp Bench. A target is matched when its evaluated tIoU is non-zero. Gemini rows use the same model-specific timestamp formats as Table 1.

Model	Event↑	Dialogue↑	Subtitle↑
Proprietary Models			
Gemini-2.5-Flash	62.9	73.8	51.7
Gemini-2.5-Pro	83.9	97.3	89.1
Gemini-3.0-Flash	71.0	78.9	70.4
Gemini-3.1-Pro	72.8	76.6	77.9
Open-source VL Models			
MiMo-VL-7B-RL	65.6	-	82.4
Qwen2.5-VL-7B-Instruct	52.2	-	46.4
Qwen3-VL-8B-Instruct	70.5	-	88.8
Qwen3-VL-30B-A3B-Instruct	**81.7**	-	90.3
Open-source Omni Models			
Qwen2.5-Omni-7B	48.2	48.0	44.9
AVoCaDO	27.2	23.4	25.5
UGC-VideoCaptioner	44.2	41.0	33.7
Video-SALMONN2+7B	54.5	41.0	40.1
Qwen3-Omni-Instruct	70.1	74.6	73.4
Qwen3-Omni-Captioner	63.4	76.6	74.2
Deci-Omni-Captioner (SFT)	78.3	96.1	**94.4**
Deci-Omni-Captioner (RL)	78.6	**96.9**	**94.4**

Table 8: Timestamp grounding over non-zero-overlap targets. Each cell reports tIoU (%) / Δt (s). Gemini rows use the same model-specific timestamp formats as Table 1.

Model	Event↑ / ↓	Dialogue↑ / ↓	Subtitle↑ / ↓
Proprietary Models			
Gemini-2.5-Flash	50.8 / 1.40	64.4 / 0.60	59.1 / 0.60
Gemini-2.5-Pro	61.8 / 1.44	70.0 / 0.57	65.5 / 0.41
Gemini-3.0-Flash	51.7 / 1.32	59.5 / 0.74	58.0 / 0.54
Gemini-3.1-Pro	59.2 / 1.46	60.4 / 0.94	59.5 / 0.50
Open-source VL Models			
MiMo-VL-7B-RL	49.4 / 2.03	-	40.3 / 1.59
Qwen2.5-VL-7B-Instruct	43.7 / 2.92	-	20.1 / 5.02
Qwen3-VL-8B-Instruct	61.0 / 1.24	-	54.1 / 1.69
Qwen3-VL-30B-A3B-Instruct	62.2 / 1.31	-	59.9 / 0.78
Open-source Omni Models			
Qwen2.5-Omni-7B	45.4 / 1.89	32.9 / 2.52	32.3 / 2.06
AVoCaDO	44.4 / 2.22	37.2 / 1.94	36.8 / 1.19
UGC-VideoCaptioner	48.6 / 4.64	30.9 / 3.05	29.0 / 2.80
Video-SALMONN2+7B	41.2 / 4.50	27.9 / 4.28	21.9 / 4.30
Qwen3-Omni-Instruct	51.9 / 1.91	37.1 / 2.69	37.2 / 1.99
Qwen3-Omni-Captioner	54.5 / 1.63	36.7 / 2.77	38.0 / 2.13
Deci-Omni-Captioner (SFT)	64.4 / 1.26	81.6 / 0.26	85.2 / 0.11
Deci-Omni-Captioner (RL)	**65.0 / 1.22**	**88.3 / 0.12**	**88.1 / 0.10**

As shown in Tables 7 and 8, Deci-Omni-Captioner (RL) remains the strongest open-source model in tIoU over non-zero-overlap targets on every track, confirming that its advantage is not caused by zero-scoring unmatched outputs. For example, Qwen3-VL-30B-A3B-Instruct matches more Event targets than Deci-Omni-Captioner (RL) (81.7% vs. 78.6%) but has lower Event tIoU over non-zero-overlap targets (62.2% vs. 65.0%). Likewise, Gemini-2.5-Pro and Deci-Omni-Captioner (RL) have similar Dialogue match rates (97.3% vs. 96.9%), whereas their Dialogue tIoU values over non-zero-overlap targets are 70.0% and 88.3%, respectively.

B.3 Cross-Judge Robustness

To assess sensitivity to the LLM evaluator, we re-score fixed captions while holding the targets, ground truth, and metric computation constant. For both Deci-Timestamp Bench and the public semantic benchmarks, we compare the reported Doubao-Seed-2.0-Pro scores with Claude-Sonnet-4.6 and GPT-5.5.

Table 9: Cross-judge robustness on Deci-Timestamp Bench. Captions, targets, ground truth, and metrics are fixed; only the LLM extractor changes. Each cell reports Match (%) / tIoU (%).

Track	Captioner	Doubao-Seed-2.0-Pro	Claude-Sonnet-4.6	GPT-5.5
Event	Deci-Omni (RL)	78.6 / 51.1	79.0 / 52.0	83.9 / 55.3
	Qwen3-Omni-Captioner	63.4 / 34.5	64.7 / 34.1	65.6 / 35.1
	Gemini-2.5-Pro	83.9 / 51.9	87.1 / 53.3	87.9 / 53.7
Dialogue	Deci-Omni (RL)	96.9 / 85.5	95.3 / 84.1	97.7 / 85.7
	Qwen3-Omni-Captioner	76.6 / 28.1	73.8 / 29.2	86.3 / 32.3
	Gemini-2.5-Pro	97.3 / 68.1	97.7 / 68.5	98.4 / 69.8
Subtitle	Deci-Omni (RL)	94.4 / 83.1	93.6 / 83.0	96.3 / 84.5
	Qwen3-Omni-Captioner	74.2 / 28.2	75.3 / 27.0	73.8 / 28.5
	Gemini-2.5-Pro	89.1 / 58.4	88.4 / 57.7	90.3 / 58.6

Table 10: Cross-judge robustness on public semantic benchmarks. Generated captions are fixed and only the evaluator changes. VCapsBench is excluded from this analysis due to its substantially larger evaluation volume.

Benchmark	Captioner	Doubao-Seed-2.0-Pro	Claude-Sonnet-4.6	GPT-5.5
AVUT	Deci-Omni (RL)	67.98	67.32	71.86
	Qwen3-Omni-Captioner	48.57	50.60	57.95
	Gemini-2.5-Pro	73.24	72.70	78.26
DailyOmni	Deci-Omni (RL)	69.03	69.49	74.88
	Qwen3-Omni-Captioner	58.86	60.86	69.95
	Gemini-2.5-Pro	75.65	76.12	81.36
UGC	Deci-Omni (RL)	73.43	70.33	72.29
	Qwen3-Omni-Captioner	52.97	63.91	61.65
	Gemini-2.5-Pro	78.09	71.27	76.39

Table 9 shows that Deci-Omni-Captioner consistently retains a large Dialogue and Subtitle margin over Qwen3-Omni-Captioner and Gemini-2.5-Pro. Event grounding remains a near-tie with Gemini-2.5-Pro: their tIoU difference is at most 1.6 points and its direction varies with the extractor, so we do not claim a judge-independent Event ordering between them. On AVUT, DailyOmni, and UGC, Table 10 preserves the same model ordering under all three evaluators despite shifts in absolute scores.

B.4 SSCA Token Attribution and Output Transfer

We quantify how much of each completion receives a span-specific rather than global advantage using the RL checkpoint's own production-schema captions. We tokenize each caption and convert the attributed character intervals to token intervals with the same overlap rule used during training. Background denotes every completion token outside all attributed spans; these tokens continue to receive the global advantage and are not masked.

Table 11: Token coverage of SSCA-attributed spans, measured on 400 production-schema JSON captions from the RL checkpoint (1,371 completion tokens per caption on average). Background tokens receive the global advantage in Eq. (3).

Rollout Type	Attributed Span	Attributed	Background
Timestamp (3 tIoU rewards)	One event per track	16.2%	83.8%
Semantic QA (2 rewards)	`visual_events` + `audio_events`	54.1%	45.9%
Overall (≈50/50 mix)	—	≈35%	≈65%

As shown in Table 11, the three queried events in a timestamp rollout jointly cover only 16.2% of its completion tokens, leaving 83.8% unrelated to those local rewards. Semantic-QA rewards govern larger whole-section spans, but still leave 45.9% of tokens in the background. Across the approximately balanced RL mixture, roughly 65% of completion tokens retain the global advantage. These measurements quantify the reward dilution that motivates assigning each local reward only to its corresponding span.

Table 12: Transfer of JSON-trained SSCA gains to NL inference on the Deci-Timestamp Bench. SSCA span attribution is applied only to structured JSON rollouts during RL; the resulting checkpoint is prompted for either NL or JSON output at inference.

Stage (Output)	Event				Dialogue				Subtitle			
	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓
SFT (NL)	53.1	37.1	49.5	1.30	80.1	66.4	70.8	0.33	86.5	77.5	75.1	0.16
RL/SSCA (NL)	54.5	41.5	49.7	1.27	89.5	76.6	78.3	0.22	92.9	82.8	81.6	0.15
RL/SSCA (JSON)	56.7	42.9	51.1	1.22	94.9	89.8	85.5	0.12	93.3	87.3	83.1	0.10

Although SSCA uses explicit JSON spans during RL, its gains are not restricted to JSON inference. As shown in Table 12, prompting the RL/SSCA checkpoint for NL output improves tIoU over the SFT checkpoint from 70.8% to 78.3% on Dialogue and from 75.1% to 81.6% on Subtitle, while preserving Event performance (49.5% to 49.7%). Thus, structured span-level credit assignment transfers to free-form captions without requiring span attribution at inference.

B.5 Disentangling Frame Rate and Timestamp-Token Granularity

To distinguish the contribution of dense temporal sampling from that of explicit fine-grained time references, we evaluate the 2×2 combinations of $\{2, 10\}$ FPS and $\{1\text{s}, 0.1\text{s}\}$ timestamp-token granularity, with inference settings matched to each training configuration.

Table 13: Factorial ablation of video frame rate and explicit timestamp-token granularity on Deci-Timestamp Bench. Overall is the annotation-count-weighted mean tIoU over all 747 targets.

FPS	Token Granularity	Overall↑	Event↑	Dialogue↑	Subtitle↑
2	1s	46.8	31.0	57.6	49.6
2	0.1s	54.4	39.3	60.5	61.2
10	1s	59.1	45.9	65.9	63.6
10	0.1s (production)	**70.6**	**48.8**	**79.2**	**80.5**

Table 13 shows consistent gains from both factors. At 2 FPS, reducing token granularity from 1s to 0.1s improves overall tIoU from 46.8% to 54.4% (+7.6); at 10 FPS, it improves from 59.1% to 70.6% (+11.5). Conversely, increasing the frame rate from 2 to 10 FPS yields +12.3 points with 1s tokens and +16.2 points with 0.1s tokens. The full configuration reaches 70.6%, well above either factor alone, indicating that dense visual sampling and fine-grained timestamp tokens provide complementary improvements.

B.6 Computational Overhead of High-FPS Processing

We measure the annotation, input-sequence, and training overhead of increasing the sampling rate from 2 to 10 FPS on 400 short clips (median duration 11.7s). Although Gemini's billed visual input increases by 4.3×, the generated caption (approximately 1.6K tokens) dominates the total API cost, so end-to-end annotation cost increases by 2.2×. For model training, we trade per-frame spatial resolution for temporal density: 10 FPS uses 200 visual tokens per frame versus 500 at 2 FPS, with adjacent frames merged in pairs.

Table 14: Annotation and training overhead when increasing the sampling rate from 2 to 10 FPS. Training measurements use Qwen3-Omni-30B-A3B on the same 64 NVIDIA H20 (140GB) GPUs, with TP=4, EP=4, CP=1, micro-batch size 1, global batch size 128, maximum length 35K, and 0.1s timestamp granularity.

Metric	2 FPS	10 FPS	Overhead
Annotation cost / clip	\$0.025	\$0.055	2.2×
Input sequence length / sample	∼6.1K	∼10.4K	1.7×
Training step time	21.9s	35.3s	1.6×
Peak memory / GPU	70.6GB	84.2GB	+19%

As shown in Table 14, a fivefold increase in frame rate results in a 1.7× longer input sequence, 1.6× greater step time, and 19% higher peak memory at unchanged parallelism. Thus, the computational overhead remains substantially below the nominal 5× frame-rate increase for the short-video regime considered in this work.

B.7 Impact of Frame Rates

In Section 4.2 we evaluate baseline open-source Omni models at 2 FPS / 500 TPF, whereas all VL models and our Deci-Omni-Captioner are evaluated at 10 FPS / 200 TPF. We justify this asymmetric protocol with a head-to-head comparison of every open-source baseline under both sampling configurations on the Deci-Timestamp Bench (Table 15) and on the four public semantic benchmarks (Table 16). Based on the experimental results, we can draw the following three conclusions:

Table 15: Comparison of open-source models on the Deci-Timestamp Bench under two sampling configurations: a higher frame rate with shorter per-frame tokens (fps=10, tpf=200) and a lower frame rate with longer per-frame tokens (fps=2, tpf=500).

Model	FPS / TPF	Event				Dialogue				Subtitle			
		R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓	R1@0.5↑	R1@0.7↑	tIoU↑	Δt (s)↓
Open-source VL Models													
MiMo-VL-7B-RL	10/200	30.8	16.1	32.4	2.03	-	-	-	-	29.2	12.7	33.2	1.59
	2/500	25.9	14.7	27.5	2.01	-	-	-	-	24.3	15.7	28.9	1.54
Qwen2.5-VL-7B-Instruct	10/200	21.0	10.3	22.8	2.92	-	-	-	-	3.4	1.1	9.3	5.02
	2/500	26.8	14.3	26.2	2.22	-	-	-	-	7.1	3.7	12.9	4.00
Qwen3-VL-8B-Instruct	10/200	47.3	29.9	43.0	1.24	-	-	-	-	52.8	34.5	48.0	1.69
	2/500	32.1	17.9	32.1	1.88	-	-	-	-	28.5	10.5	30.8	1.62
Qwen3-VL-30B-A3B-Instruct	10/200	55.6	37.5	50.8	1.31	-	-	-	-	62.2	37.8	54.1	0.78
	2/500	45.1	29.0	43.8	1.44	-	-	-	-	34.1	12.7	33.3	1.21
Open-source Omni Models													
Qwen2.5-Omni-7B	10/200	17.0	8.9	18.2	11.41	2.7	0.4	7.0	3.27	9.0	3.0	10.3	2.01
	2/500	21.4	8.0	21.9	1.89	8.6	4.3	15.8	2.52	7.9	3.4	14.5	2.06
AVoCaDO	10/200	2.7	0.9	4.5	5.56	3.5	1.6	5.6	2.26	3.0	0.4	4.6	1.99
	2/500	12.1	6.2	12.1	2.22	5.5	2.0	8.7	1.94	6.4	1.9	9.4	1.19
UGC-VideoCaptioner	10/200	13.8	5.8	14.5	10.35	0.0	0.0	2.4	11.70	0.4	0.0	1.3	12.70
	2/500	22.3	12.1	21.5	4.64	8.2	3.1	12.7	3.05	4.5	2.2	9.8	2.80
Video-SALMONN2+7B	10/200	5.8	3.6	10.0	15.83	2.0	0.8	4.2	21.63	0.4	0.0	1.7	17.36
	2/500	19.6	11.6	22.4	4.50	5.1	1.6	11.5	4.28	3.0	0.7	8.8	4.30
Qwen3-Omni-Instruct	10/200	3.1	0.4	9.3	9.73	2.0	0.4	3.6	15.93	0.4	0.4	4.0	14.78
	2/500	37.1	19.6	36.4	1.91	18.8	10.5	27.7	2.69	23.6	9.4	27.3	1.99
Qwen3-Omni-Captioner	10/200	5.4	2.7	9.9	10.76	3.1	1.2	5.2	19.84	1.5	0.4	4.0	16.23
	2/500	33.9	20.1	34.5	1.63	20.7	8.6	28.1	2.77	24.7	8.2	28.2	2.13
Deci-Omni-Captioner (RL)	10/200	56.7	42.9	51.1	1.22	94.9	89.8	85.5	0.12	93.3	87.3	83.1	0.10
	2/500	49.6	35.7	46.3	1.24	93.4	85.2	82.5	0.16	91.0	76.0	76.8	0.17

Open-source VL models mostly benefit from higher FPS. As shown in Table 15, three out of four VL baselines (MiMo-VL-7B-RL, Qwen3-VL-8B-Instruct, and Qwen3-VL-30B-A3B-Instruct) achieve better timestamp grounding at 10 FPS / 200 TPF. This improvement occurs because a higher sampling rate provides denser visual information, allowing the models to capture more precise temporal boundaries for visual events and text appearances.

Open-source Omni models collapse at higher FPS. All six open-source Omni baselines suffer a catastrophic degradation on timestamp grounding when switching to 10 FPS, as shown in Table 15. A similar degradation persists on semantic benchmarks (Table 16). We attribute this collapse to a severe training-inference mismatch: these models are post-trained at around 2 FPS and encode temporal references implicitly through TM-RoPE. Increasing the frame density alters the mapping between absolute time and positional codes, leaving the models without reliable temporal anchors.

Deci-Omni-Captioner improves with higher FPS. Our model is the only Omni model that consistently prefers 10 FPS on the Deci-Timestamp Bench, with the largest gain on Subtitle. On semantic benchmarks, it also gains marginally on highly dynamic subsets (e.g., UGC and VCapsBench). This robustness is driven by our explicit timestamp tokens, which decouple temporal grounding from implicit TM-RoPE encoding, allowing denser frames to supply more visual evidence without re-scaling temporal references. Furthermore, as detailed in Appendix A.2, our SFT phase explicitly mixes both 2 FPS / 500 TPF and 10 FPS / 200 TPF configurations. This exposes the model to high-density sub-second calibration during training, perfectly aligning with the high-FPS inference regime. Consequently, the higher frame rate successfully resolves sub-second boundaries (e.g., subtitle transitions) and captures highly dynamic visual details that are aliased away at 2 FPS.

Table 16: Comparison of open-source Omni models on the public semantic benchmarks under two sampling configurations.

Model	FPS / TPF	Omni-Bench								VL-Bench		
		AVUT			DailyOmni	UGC				VCapsBench		
		gemini	human	Overall	Overall	Visual	Audio	Details	Overall	AR	IR ↓	CR
Open-source Omni Models												
Qwen2.5-Omni-7B	10/200	36.89	43.29	38.77	47.92	62.50	29.66	40.36	44.16	37.88	17.32	45.81
	2/500	40.61	44.51	41.76	55.01	65.42	41.71	42.86	49.99	33.49	16.87	40.29
AVoCaDO	10/200	61.25	59.35	60.69	63.48	71.90	58.14	51.16	60.39	55.75	16.06	66.42
	2/500	62.18	60.57	61.71	70.42	71.21	56.88	50.29	59.46	53.87	16.86	64.80
UGC-VideoCaptioner	10/200	42.89	46.34	43.91	49.92	66.39	46.41	43.35	52.04	46.49	16.16	55.45
	2/500	43.49	45.93	44.21	54.70	66.41	47.74	43.55	52.56	44.74	16.51	53.59
Video-SALMONN2+7B	10/200	50.51	51.63	50.84	62.87	67.48	49.45	44.90	53.94	42.25	16.32	50.49
	2/500	52.12	53.25	52.45	61.63	67.23	49.02	44.42	53.55	41.61	16.86	50.05
Qwen3-Omni-Instruct	10/200	40.44	48.58	42.83	51.62	69.73	36.53	45.95	50.73	46.56	14.16	54.24
	2/500	42.72	50.41	44.98	55.93	70.32	44.57	46.67	53.85	46.51	14.98	54.71
Qwen3-Omni-Captioner	10/200	38.75	48.78	41.70	51.93	67.79	32.36	44.27	48.13	53.36	16.13	63.61
	2/500	46.19	54.27	48.57	58.86	69.40	42.94	46.56	52.97	53.61	16.28	64.04
Deci-Omni-Captioner (RL)	10/200	67.77	68.50	67.98	69.03	78.65	74.30	67.36	73.43	64.03	15.50	75.78
	2/500	66.78	69.34	67.53	69.80	78.37	74.13	66.44	72.97	63.05	16.00	75.06

B.8 Quantitative and Qualitative Results: Two Output Modes of Deci-Omni-Captioner

A key advantage of Deci-Omni-Captioner is its flexibility to generate comprehensive multimodal annotations in two distinct modes: NL and structured JSON. This dual-output capability is achieved by simply swapping the instruction prompts during inference. To instruct the model to generate a fluid, narrative-style NL description, we use the following prompt:

Prompt for NL Output

```
### System Prompt
You are a helpful assistant.

### User Prompt
You are a multimodal expert. Please watch this single-shot video and generate a detailed natural-language description
       covering both visual and auditory information.

**Output format**:

1. **Background paragraph**: Describe the overall atmosphere, scene environment, lighting conditions, visual style,
      and color palette.

2. **Integrated narrative paragraph**: Provide a coherent summary of the events in the video, weaving in the auditory
       experience to describe how sounds complement and drive the narrative.

3. **Timestamped detail paragraphs**: Use `[start_s-end_s]` format to mark time ranges; describe visual actions and
      details for each segment, naturally integrating any sound effects, dialogue, or other audio events that occur
      within that time range.

4. **Full-duration audio paragraphs**: Background music or ambient noise that spans nearly the entire video should be
       described in separate timestamped paragraphs placed after all visual paragraphs.

**Constraints**: Timestamps must be precise to 0.1s; use natural, fluent English; do not use JSON or structural tags.
```

Conversely, to instruct the model to generate a highly structured, machine-readable JSON representation, we use the following prompt:

Prompt for JSON Output

```
### System Prompt
You are a helpful assistant.

### User Prompt
You are a multimodal expert. Please analyze the single-shot video and generate a JSON containing Visual, Overlay, and
       Audio tracks.

**1. Visual Layer**:
* Extract global style, lighting, and atmosphere.
* **Entity Registration**: Assign `<ID:x>` ONLY to narrative protagonists, key props, and **speakers with visible
      faces**.
* **Event Segmentation**: Segment based on action density; detail subject actions, interaction details, and camera
      movement changes.
```

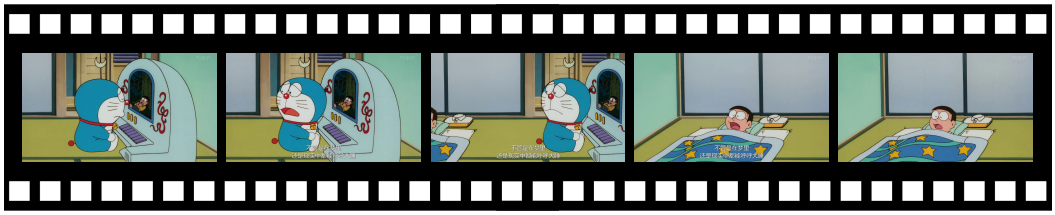
```
**2. Overlay Layer**:
* Independently record non-physical elements such as subtitles and UI; strictly prohibit mixing them into visual
      descriptions.

**3. Audio Layer**:
* **Layered Identification**: Prioritize extracting Music/Ambience, then process Dialogue/SFX.
* **Transcription**: Retain the original language and dialects; content must be verbatim ("what you hear is what you
      get").
* **Lip-Sync**: Mark as ‘true‘ ONLY when the speaker ID is clearly visible in the frame and lip movements match the
      audio; otherwise, strictly set to ‘false‘.

**Constraints**: Timestamps must be precise to 0.1s; vague vocabulary is strictly prohibited; strictly adhere to the
      output Schema.
```

While both formats offer high degrees of freedom for downstream video generation applications, our empirical analysis reveals a performance trade-off between them. As demonstrated in Table 17, generating annotations in the NL format results in slightly weaker temporal grounding and semantic scores compared to the JSON format. This is largely because the JSON schema forces the model to decouple complex multimodal events into independent, orthogonal tracks (Visual, Audio, Overlay) before grounding them, whereas the NL format requires the model to interweave these events into a fluid narrative, increasing the cognitive load of temporal tracking. Therefore, to achieve the highest precision and maximum structural freedom for downstream tasks, we recommend the JSON format.

Figure 8 showcases a complete, real-world annotation example generated by Deci-Omni-Captioner, illustrating both the narrative fluidity of the NL output and the fine-grained, track-separated precision of the JSON output.

Table 17: Comparison of two output formats (NL vs JSON) of Deci-Omni-Captioner (RL) on the public semantic benchmarks.

Model	**Omni-Bench**								**VL-Bench**		
	AVUT			**DailyOmni**	**UGC**				**VCapsBench**		
	gemini	**human**	**Overall**	**Overall**	**Visual**	**Audio**	**Details**	**Overall**	**AR**	**IR** $\downarrow$	**CR**
NL	65.57	62.40	64.64	70.57	77.50	70.68	60.49	69.55	63.64	15.73	75.52
JSON	67.77	68.50	67.98	69.03	78.65	74.30	67.36	73.43	64.03	15.50	75.78



Natural Language Caption

This lighthearted, comedic, and sleepy anime clip is set in a traditional Japanese-style indoor bedroom, which features a green tatami mat floor, a wooden door, and a large sliding fusuma door. The scene is lit with bright, even, flat lighting characteristic of traditional 2D animation, with no distinct shadows or complex light sources visible. It uses a classic cel-shaded anime art style, with vibrant, saturated primary colors dominated by the blue of the robotic cat, the green of the tatami mats, and the light blue of the dream machine, paired with a neutral color temperature.

In the brightly lit Japanese-style bedroom, a blue robotic cat sits with a tired expression in front of a light blue dream machine, while a young bespectacled boy is visible sleeping inside the machine's screen. The camera then pans left to reveal the same boy sleeping soundly in a bed on the tatami floor, covered by a blue and yellow star-patterned blanket, as the robotic cat remains partially visible on the right edge of the frame. Sound-wise, the audio is very clean and simple, typical of a modern anime dub, with a quiet, intimate soundscape that evokes a silent bedroom with no background music or ambient noise, and high recording quality free of echo or distortion.

[0.0s–2.7s] In a medium shot, the round blue robotic cat, who has large white eyes, a red nose, six whiskers, a red collar fitted with a yellow bell, and a red and white striped tail, sits on the green tatami floor with a tired, half-lidded expression. To his right stands the light blue, upright, retro-futuristic dream machine, which has a large arched screen, a purple keyboard, yellow buttons, and red swirling decorative patterns on its side; the young bespectacled boy is visible sleeping inside the machine's screen. A static white "WOWOW" logo with smaller Japanese text "シネマ" underneath it is displayed in the top right corner of the screen throughout this segment. Around 1.2s, the robotic cat speaks in a sleepy, drawn-out tone, saying "夢の中でも現実でも、くうすか寝ちゃって", his lines lip-synced to his movements, and from 1.0s to 4.3s, white subtitles with a thin black outline appear at the bottom center of the screen, reading "不管是 在梦里 还是现实中都能呼呼大睡".

[2.7s–5.6s] The camera pans left, moving away from the robotic cat and the dream machine to reveal the young bespectacled boy sleeping soundly in a bed on the tatami floor, covered up to his chin with a blue blanket decorated with yellow stars. The robotic cat remains partially visible on the far right of the frame, and the "WOWOW" logo stays in the top right corner for the full length of this segment. The robotic cat's sleepy line of dialogue continues until 4.7s, and around 4.7s, the boy lets out a loud, deep snore as he sleeps.

JSON Format Caption

"color_palette": "Bright and saturated primary colors, dominated by the blue of the cat, the green of the floor, and the light blue of the machine. The tone is vibrant and clean.",
"atmosphere": "Comedic and lighthearted, with a hint of surrealism due to the dream sequence.",
"environment": "An indoor, traditional Japanese-style bedroom with green tatami mat flooring, a wooden door frame, and a large sliding fusuma door.",
"style": "Classic 2D anime, characteristic of the Doraemon series.",
"light": "Bright, even, and flat lighting, typical of traditional cel animation, with no distinct shadows or complex light sources."

"short_caption": "A blue robotic cat looks at a machine showing a boy's dream, then the camera pans to reveal the boy himself sleeping and snoring loudly in his bed.",
"long_caption": "In a brightly lit anime-style bedroom, a blue robotic cat sits with a weary expression, observing a light blue machine. The machine's screen displays a cartoon boy sleeping in a forest. The cat speaks with a tired look on its face. The camera then pans to the left, revealing the cartoon boy from the screen is actually a classmate sleeping in a real bed. He is covered by a blue and yellow star-patterned blanket and is snoring loudly.",
"audio_summary": "The audio experience is very clean and focused, taking place in a quiet indoor setting. The clip begins with the clear, high-pitched Japanese voice of a robotic cat speaking a full sentence in the foreground. This is immediately followed by the loud, deep, and resonant sound of a boy snoring, which dominates the latter half of the clip. There is no background music or ambient noise, making the dialogue and sound effect the sole auditory elements. The sound quality is modern and clear, with no echo or distortion.",

"overlay_events": [{"time_range": ["0.0", "5.6"], "type": "logo", "description": "A static white logo \"WOWOW\" with Japanese characters \"シネマ\" below it is displayed in the top right corner of the screen."}, {"time_range": ["0.8", "4.3"], "type": "subtitle", "description": "White Chinese subtitles appear at the bottom center of the screen, reading \"不管是 在梦里 还是现实中都能呼呼大睡\"."}]

"entities": [{"id": "<ID:1>", "type": "subject", "description": "A blue, round, robotic cat with large white eyes, a red nose, six whiskers, a red collar with a yellow bell, and a small red tail with a red ball at the end."}, {"id": "<ID:2>", "type": "object", "description": "A light blue, retro-futuristic machine with a large, curved top. It has a screen on the front, a purple keyboard below it, and red swirling patterns on its body."}, {"id": "<ID:3>", "type": "subject", "description": "A young Japanese boy with short black hair, wearing round glasses and a yellow short-sleeved shirt."}]

"visual_events": [{"time_range": ["0.0", "2.9"], "action_density": "low (Action-based)", "short_description": "A blue robotic cat (<ID:1>) looks at a machine (<ID:2>) and speaks with a tired expression.", "long_description": "In a medium shot, a blue robotic cat (<ID:1>) sits on a green tatami floor, looking towards a light blue machine (<ID:2>) with a weary, half-lidded expression. The machine's screen shows a cartoon boy (<ID:3>) sleeping in a forest. The cat's mouth opens and closes as it speaks."}, {"time_range": ["2.9", "5.6"], "action_density": "low (Action-based)", "short_description": "The camera pans left to show a boy (<ID:3>) sleeping and snoring in bed.", "long_description": "The camera pans left, moving away from the cat (<ID:1>) and the machine (<ID:2>). The pan reveals a boy (<ID:3>) sleeping soundly in a bed, covered by a blue blanket with yellow stars. His mouth is wide open as he snores loudly."}]

"audio_events": [{"time_range": ["1.2", "4.7"], "type": "dialogue", "description": "Japanese male voice, the robotic cat (<ID:1>) speaks in a clear, slightly high-pitched, and weary tone: \"夢の中でも現実でも、くうすか寝ちゃって。\"", "lip_sync": true}, {"time_range": ["4.7", "5.6"], "type": "sfx", "description": "The boy (<ID:3>) snores loudly and deeply.", "lip_sync": false}]

Figure 8: A comprehensive annotation example generated by the proposed Deci-Omni-Captioner. The framework outputs multimodal descriptions in two distinct formats. (Top) NL Caption: This format presents a cohesive narrative that sequentially details the static visual properties (e.g., style, lighting, environment), the overarching dynamic elements, and a fine-grained, timestamped breakdown of distinct visual and auditory events. (Bottom) JSON Format Caption: This format offers a highly structured, machine-readable representation, categorized into distinct color-coded blocks. It isolates static background attributes (orange), global video summaries (purple), overlay elements such as logos and subtitles (yellow), explicitly defined visual entities (red), and temporally grounded visual (green) and audio (blue) events that are directly linked to the aforementioned entities.

C Broader Impacts

The release of Deci-Omni-Captioner and the Deci-Timestamp Bench is poised to significantly advance the field of temporally controllable video generation. By providing an open-source captioning framework with sub-second timestamp grounding for dialogue and on-screen text together with improved visual-event localization, we lower the barrier for researchers and practitioners to create high-quality, audio-visually aligned training data. This can accelerate the development of more realistic and structurally coherent video generation models, with positive applications spanning creative arts, education, and multimedia production. While we do not foresee direct negative societal impacts specific to our contributions, as with any generative AI technology, improvements in video generation could potentially be misused for creating deceptive content. We encourage the community to adopt responsible usage guidelines and develop robust detection mechanisms alongside the advancement of these generative tools.

D Limitations

While Deci-Omni-Captioner achieves SOTA timestamp grounding and semantic captioning performance, we acknowledge several limitations that present opportunities for future work. First, our comprehensive data construction pipeline relies on an ensemble of advanced models (e.g., Gemini-2.5-Pro, HunyuanOCR, and Doubao-Seed-ASR-2.0) to explicitly calibrate temporal anchors and generate multi-reward queries. Increasing the sampling rate from 2 to 10 FPS raises per-clip annotation cost by 2.2×, training step time by 1.6×, and peak memory by 19% (Appendix B.6), so computational cost remains a consideration at billion-video scale. Second, our current framework and the Deci-Timestamp Bench focus on short-form videos (typically under 35 seconds) to closely align with the practical requirements of modern temporally controllable video generation models. Extending this fine-grained temporal grounding capability to extremely long-form content, such as full-length movies, remains an interesting future direction. Third, timestamp grounding remains more challenging for visual events, where the model obtains 51.1% tIoU and 1.22s Δt, compared with 85.5%/0.12s on Dialogue and 83.1%/0.10s on Subtitle. This gap reflects the greater semantic ambiguity of visual-event boundaries. Finally, although SSCA is compatible in principle with other advantage-based policy-gradient objectives, we have validated it only under GSPO; evaluating its interaction with alternative clipping and importance-sampling schemes remains future work.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paperś contributions and scope?

   Answer: [Yes]

   Justification: We explicitly state our contributions in the introduction (Section 1) and abstract, and all claims are supported by extensive experiments in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include a dedicated Limitations section in Appendix D to discuss the computational overhead and the scope of video durations.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper focuses on empirical methods for multimodal captioning and does not introduce new theoretical results or proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed descriptions of the model architecture, data construction pipelines, and experimental setups in Section 3, Section 4, and the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: Due to file size constraints, we cannot include the dataset and model checkpoints in the supplemental material. We will fully open-source the test set, model checkpoints, and corresponding code upon paper acceptance.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Detailed training configurations, hyperparameters, and evaluation protocols are comprehensively provided in Section 4 and the Appendix.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our evaluations rely on standard deterministic metrics (e.g., tIoU, Accuracy) on established benchmarks, which is standard practice in this field, making error bars less applicable.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the compute resources used for training and evaluation in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms to the NeurIPS Code of Ethics, and our data construction relies on publicly available resources and ethical human annotation.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a dedicated Broader Impacts section in Appendix C to discuss both the positive potential for advancing video generation and the general risks of generative AI.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our captioning model and dataset do not inherently pose a high risk for misuse, as they are primarily descriptive tools rather than raw generative models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We properly cite all base models (e.g., Qwen3-Omni, Gemini), tools, and public benchmarks used, and respect their respective open-source licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We introduce the Deci-Timestamp Bench and our SFT/RL datasets, which are thoroughly documented in Section 3, Section 4, and the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We detail the human annotation process for our RL dataset and Deci-Timestamp Bench, including consistency criteria, in Section 3 and the Appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our human annotation involves standard video labeling tasks that do not pose risks to participants or require IRB approval.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: We explicitly describe the extensive use of LLMs/MLLMs (e.g., Gemini-2.5-Pro, Doubao-Seed-2.0-Pro) for data synthesis and evaluation in Sections 3 and 4.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.