VISD: Enhancing Video Reasoning via Structured Self-Distillation

Hao Lin[1,3*] **Kunyang Lv**[2*] **Xu Jiang**[3*] **Jingqi Tian**[4]
Zhongjing Du[3] **Jiayu Ding**[3] **Qiaoman Zhang**[3] **Hongbo Jin**[3†]

[1]School of Software Engineering, HUST
[2]Wuhan University [3]Peking University [4]Tsinghua University

Project Page: `https://lkyyy111.github.io/VISD`

Abstract

Training VideoLLMs for complex reasoning remains challenging due to sparse sequence-level rewards and the lack of fine-grained credit assignment over long, temporally grounded reasoning trajectories. While reinforcement learning with verifiable rewards (RLVR) provides reliable supervision, it fails to capture token-level contributions, leading to inefficient learning. Conversely, existing self-distillation methods offer dense supervision but lack structure and diagnostic specificity, and often interact unstably with reinforcement learning. In this work, we propose **VISD**, a structured self-distillation framework that introduces diagnostically meaningful privileged information for video reasoning. VISD employs a video-aware judge model to decompose reasoning quality into multiple dimensions, including answer correctness, logical consistency, and spatio-temporal grounding, and uses this structured feedback to guide a teacher policy for token-level supervision. To stably integrate dense supervision with RL, we adopt a direction–magnitude decoupling mechanism, where rollout-level advantages computed from rewards determine update direction, while structured privileged signals modulate token-level update magnitudes. This design enables semantically aligned and fine-grained credit assignment, improving both reasoning faithfulness and training efficiency. Additionally, VISD incorporates curriculum scheduling and EMA-based teacher stabilization to support robust optimization over long video sequences. Experiments on diverse benchmarks show that VISD consistently outperforms strong baselines, improving answer accuracy and spatio-temporal grounding quality. Notably, VISD reaches these gains with nearly $\mathbf{2}\times$ faster convergence in optimization steps, highlighting the effectiveness of structured self supervision in improving both performance and sample efficiency for VideoLLMs.

1 Introduction

Recent progress in VideoLLMs [32, 54, 12, 52] has significantly advanced multimodal reasoning, enabling models to answer complex questions grounded in dynamic visual content. Despite these improvements, training VideoLLMs remains fundamentally challenging. Unlike static image or text reasoning, video understanding requires aligning long reasoning trajectories with temporally evolving visual evidence [33, 11, 48, 42], where both the correctness of the final answer and the faithfulness of intermediate reasoning are crucial. This introduces a unique difficulty: learning signals must not only indicate whether an output is correct, but also identify *where* and *why* the reasoning process

[*]These authors contributed equally to this work.
[†]Corresponding author.

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

succeeds or fails. Without such fine-grained guidance, training often becomes inefficient, requiring a large number of samples to propagate sparse rewards through long reasoning trajectories.

A natural approach to training such models is reinforcement learning with verifiable rewards [15, 11, 27] (RLVR), where correctness signals are derived from ground-truth answers or external verifiers. While effective in providing reliable optimization directions, these rewards are inherently sparse and operate at the sequence level [15, 50, 58]. As a result, they fail to capture the heterogeneous contributions of individual tokens within long reasoning chains, especially when errors arise from subtle temporal misalignment or incorrect visual grounding. This limitation becomes more pronounced in video settings, where a correct answer may still be supported by flawed reasoning, and vice versa.

An alternative line of work leverages self-distillation to provide dense token-level supervision [57, 20, 50, 39]. However, existing approaches typically treat the additional supervision as unstructured or modality-agnostic signals. When directly applied to VideoLLMs, this leads to two key issues. First, the supervision lacks diagnostic specificity, making it difficult to distinguish between different types of reasoning errors, such as logical inconsistency versus grounding failure [31, 35]. Second, the interaction between self-distillation and reinforcement learning is often unstable, as auxiliary signals may override or conflict with reward-driven optimization [23, 26].

In this work, we propose **VISD**, a structured self-distillation framework that integrates reinforcement learning with structured privileged information. The central idea is to elevate self-distillation from a generic auxiliary signal to a structured supervision space that captures the compositional nature of reasoning errors in video tasks. Concretely, VISD introduces a video-aware judge model that evaluates each sampled reasoning trajectory and produces structured feedback along multiple dimensions, including answer correctness, reasoning consistency, and spatio-temporal grounding quality. This feedback is used as privileged information to condition a teacher policy, which provides token-level guidance that is both fine-grained and diagnostically meaningful. In contrast to prior approaches [57, 20, 50, 49] that rely on implicit or latent supervision, our design explicitly encodes the causes of reasoning failures, enabling more targeted corrections during training.

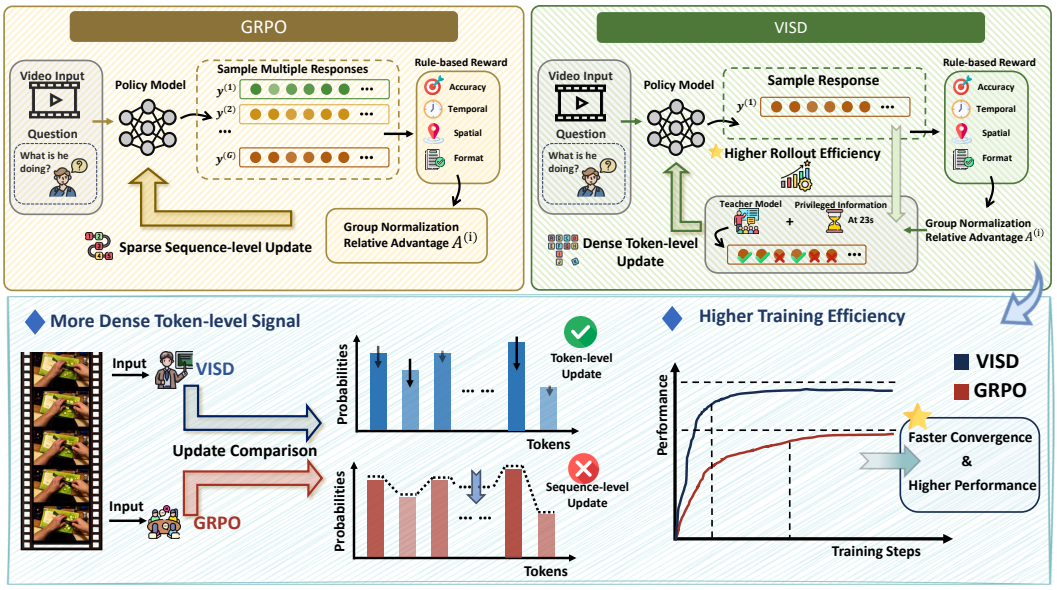


Figure 1: *Comparison between RLVR method and VISD*. **(a)** The RLVR method, e.g., GRPO, provides sparse sequence-level signals and fails to capture fine-grained spatial and temporal evidence. **(b)** VISD uses dense token-level updates, thereby improving convergence speed and overall performance.

To effectively integrate structured supervision with reinforcement learning, we extend the direction–magnitude decoupling paradigm, with inspiration from RLSD [50]. Specifically, rollout-level advantages computed from rewards determine the direction of policy updates, ensuring alignment with task objectives, while structured privileged information modulates the magnitude of token-level updates. This separation allows the model to benefit from dense supervision without compromising the stability of reward-driven optimization. Importantly, because the privileged information is structured, the resulting credit assignment is not only fine-grained but also semantically aligned with different

aspects of video reasoning, as summarized in Figure 1. As a result, VISD not only improves the quality and faithfulness of reasoning, but also significantly enhances learning efficiency by enabling faster convergence with more informative and structured training signals.

Training VideoLLMs further poses challenges due to long-horizon dependencies and heterogeneous reward signals [12, 33, 11]. VISD addresses these issues through a set of principled optimization strategies. We adopt a curriculum that gradually transitions from structured self-distillation to reinforcement learning and maintain an exponential moving average teacher to stabilize token-level supervision. Together, these designs enable robust and scalable training under complex video reasoning scenarios.

We evaluate VISD on the Open-o3-Video benchmarks [33] and demonstrate consistent improvements over strong baselines across a wide range of video reasoning tasks [12, 11]. Beyond quantitative gains in accuracy, VISD also yields better grounding quality and more interpretable reasoning behaviors, highlighting the benefits of structured supervision.

In summary, our main contributions are as follows:

- We propose **VISD**, a structured self-distillation framework tailored for video reasoning. VISD leverages a video-aware judge model to evaluate reasoning quality across multiple dimensions, providing diagnostically meaningful token-level supervision.
- We integrate a stable self-distillation training design tailored for video reasoning, combining direction–magnitude decoupled optimization with feedback-conditioned teacher replay, top-K local support, curriculum annealing, and EMA teacher stabilization.
- We demonstrate the effectiveness of VISD through extensive evaluation on diverse video benchmarks. VISD consistently outperforms strong baselines, improving answer accuracy and spatio-temporal grounding quality while achieving nearly $\mathbf{2}\times$ faster convergence in optimization steps.

2 Related Work

VISD is most closely related to three lines of work: video large language models, reinforcement learning with verifiable rewards, and on-policy/self-distillation methods for fine-grained credit assignment. Recent VideoLLMs extend vision-language models to dynamic visual understanding through frame sampling, temporal aggregation, and video-specific instruction tuning [32, 54, 56, 53, 12, 24, 19, 9]. While these models improve video question answering and temporal reasoning, they are usually trained from static annotations and therefore provide limited supervision for identifying which step of a long reasoning trajectory causes an error. RLVR methods address reasoning by optimizing sequence-level rewards [15, 38, 11, 25, 44], but their scalar advantages remain coarse for localized temporal, spatial, and object-grounding mistakes. Self-distillation and on-policy distillation provide denser teacher-derived token signals [17, 13, 37, 1, 57, 20, 50], yet existing formulations often treat the teacher signal as generic distributional guidance rather than an explicit diagnosis of multimodal reasoning failures. VISD differs by conditioning the teacher on video-aware judge feedback: the rollout-level reward still determines the update direction, while feedback-conditioned teacher replay modulates token-level magnitudes according to answer consistency, temporal coverage, and spatial grounding errors. Extended related works are provided in Appendix A.

3 Method

3.1 VISD Framework Overview

We propose **VISD**, a structured self-distillation framework designed for video reasoning, where the core challenge lies in aligning long-horizon reasoning trajectories with fine-grained spatio-temporal evidence. Unlike conventional reinforcement learning approaches that rely on sparse sequence-level rewards [15, 50], VISD introduces structured privileged supervision to enable token-level credit assignment that is both semantically meaningful and modality-aware.

Formally, given a video-question pair $x = (v, q)$, a policy model π_θ autoregressively generates a reasoning trajectory $y = (y_1, \ldots, y_T)$. The trajectory may include intermediate reasoning steps,

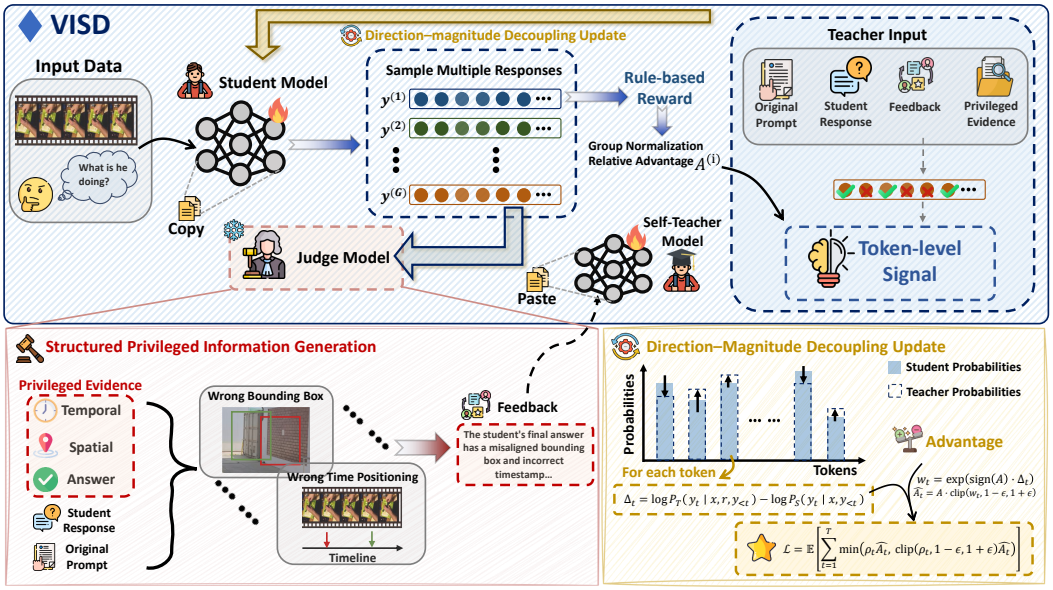


Figure 2: *Overview of VISD.* VISD first uses a video-aware judge to generate structured privileged feedback for each student rollout, then replays the same completion with a feedback-conditioned teacher. The rollout-level reward advantage determines the update direction, while the teacher-student discrepancy modulates token-level update magnitudes for fine-grained credit assignment.

temporal references, object mentions, and the final answer. A verifier provides a scalar reward $R(x, y)$, which reflects task-level correctness and, when applicable, grounding quality. However, such reward signals are inherently sparse and fail to capture the heterogeneous contributions of individual tokens in long reasoning sequences.

To address this limitation, VISD augments the standard on-policy training pipeline with two auxiliary components. First, a teacher policy is instantiated from the same model but conditioned on privileged information, providing token-level assessments along the student's trajectory. Second, a dedicated judge model produces structured feedback tailored to video reasoning, which serves as the source of privileged information. The resulting framework retains the on-policy nature of reinforcement learning while introducing a structured signal pathway that refines credit assignment at token level.

The overall training objective follows a policy-gradient formulation, illustrated in Figure 2:

$$\nabla_\theta \mathcal{J} = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[\sum_{t=1}^{T} \hat{A}_t \nabla_\theta \log \pi_\theta(y_t | x, y_{<t}) \right] \tag{1}$$

where $\hat{A}_t$ denotes a token-level advantage that integrates both environment reward and structured privileged information. Crucially, VISD preserves the reliability of environment-driven optimization while enriching it with fine-grained, structured signals that are particularly suited for video reasoning.

The full VISD training procedure is summarized in Algorithm 1 in Appendix C.1.

3.2 Structured Privileged Information Generation

A key observation underlying our design is that, in video reasoning, not all errors are equal: incorrect predictions may arise from temporal misalignment, spatial grounding failures, or inconsistencies between reasoning and answer. Capturing such distinctions is essential for effective learning, yet existing self-distillation methods [57, 20, 50] treat privileged information as an unstructured signal, limiting their ability to guide fine-grained corrections.

To overcome this limitation, VISD introduces a judge model J that transforms each rollout trajectory into a structured representation of reasoning quality. Given an input x, a sampled trajectory y, verified answer-side information $a^\star$, and available grounding evidence e, the judge produces structured feedback $f = J(x, y, a^\star, e)$. We package the teacher-side privileged context as $r = (a^\star, e, f)$. The feedback f encapsulates multiple complementary aspects of evaluation. Specifically, it assesses (i)

the correctness of the predicted answer, (ii) the consistency between the reasoning process and the answer, and (iii) the underlying causes of inconsistencies when they arise. More importantly, to account for the unique challenges of video understanding, the judge incorporates modality-specific feedback by explicitly evaluating temporal alignment and spatial grounding. This includes identifying whether the reasoning references incorrect time windows, whether objects are mislocalized, and whether the cited evidence is inconsistent with the visual content, etc. The teacher distribution is thus defined as:

$$P_T(y_t|x, r, y_{<t}) = \pi_{\bar{\theta}}(y_t|x, r, y_{<t}) \tag{2}$$

where $\bar{\theta}$ denotes the teacher parameters; in the main EMA setting, $\bar{\theta} \equiv \theta_{\text{ema}}$. The conditioning on r allows the teacher to rescore the same student completion using verified answer/evidence and rollout-specific judge feedback.

3.3 Direction–Magnitude Decoupled Structured Self-Distillation

Given a sampled trajectory, we compute the token-level discrepancy between teacher and student distributions on a compact top-K local support. At each position t, we construct a teacher-centered support

$$\mathcal{U}_t = \text{TopK}_{P_T(\cdot|x,r,y_{<t})}(K) \cup \{y_t\}, \tag{3}$$

which augments the teacher top-K candidates with the realized on-policy token so that the generated action is always covered. Let $\widetilde{P}_T(\cdot \mid x, r, y_{<t})$ and $\widetilde{P}_S(\cdot \mid x, y_{<t})$ denote the teacher and student distributions renormalized over $\mathcal{U}_t$. We define:

$$\Delta_t = \text{sg}\left(\log \widetilde{P}_T(y_t|x, r, y_{<t}) - \log \widetilde{P}_S(y_t|x, y_{<t})\right) \tag{4}$$

where $\text{sg}(\cdot)$ denotes the stop-gradient operator. This top-K local-support discrepancy measures how the structured privileged information revises the model's belief about the realized token within a locally relevant candidate set. In contrast to standard self-distillation, where such discrepancies are treated as flat signals, here Δ_t implicitly encodes structured corrections derived from the judge, including temporal, spatial, and logical inconsistencies. We provide the detailed top-K support derivation, including direction preservation and bounded magnitude redistribution, in Appendix E.3.

Unlike existing mainstream self-distillation paradigms, VISD adopts a direction–magnitude decoupling strategy to integrate structured privileged information into policy optimization. The central principle is to separate *what direction the policy should move* from *how strongly each token should be updated*, thereby preventing unreliable signals from corrupting the optimization trajectory.

The sequence-level advantage A, computed from the reward, determines whether a trajectory should be reinforced or penalized. We then define a token-level reweighting factor:

$$w_t = \exp\left(\text{sign}(A) \cdot \Delta_t\right) \tag{5}$$

When the trajectory has positive advantage, tokens favored by the teacher receive larger update magnitudes; when it has negative advantage, tokens disfavored by the teacher receive stronger suppression. To ensure stability, we clip the reweighting factor:

$$m_t = \text{clip}(w_t, 1 - \epsilon_w, 1 + \epsilon_w), \tag{6}$$

where ϵ_w bounds the amount of token-level credit redistribution. This formulation keeps the sign of the token-level advantage determined by the rollout-level advantage, while the structured privileged information redistributes credit within the trajectory. Because the teacher is conditioned on diagnostically meaningful feedback, the resulting reweighting reflects not only confidence differences, but also the semantic causes of errors.

3.4 Stable Policy Optimization for Long Video Sequences

Optimizing VideoLLMs over long reasoning trajectories is inherently challenging due to sparse rewards, heterogeneous supervision signals, and the instability introduced by token-level credit assignment. In practice, these challenges are further amplified by the multi-dimensional nature of video reasoning, where correctness, reasoning consistency, and spatio-temporal grounding often induce rewards with different scales and statistical properties. VISD addresses these issues through a set of design choices that ensure stable and robust optimization.

We adopt a *curriculum strategy* that gradually transitions from structured self-distillation to reinforcement learning. At the early stage of training, updates are modulated by the teacher-induced token-level reweighting, allowing the model to quickly acquire fine-grained reasoning patterns guided by structured privileged information. As training progresses, the influence of self-distillation is gradually annealed, and the optimization increasingly relies on environment rewards. Formally, for rollout i, the final token-level advantage is defined as:

$$\hat{A}_t^{(i)} = A^{(i)} \cdot \left((1-\lambda) + \lambda \cdot m_t^{(i)}\right) \tag{7}$$

where $\lambda \in [0,1]$ controls the strength of teacher-induced token reweighting. This formulation effectively interpolates between structured self-distillation and standard reinforcement learning, ensuring both efficient early learning and stable long-term convergence.

To stabilize the teacher signal, we maintain an exponential moving average version of the policy parameters to construct the teacher model. Specifically, the teacher parameters $\bar{\theta}$ are updated as:

$$\bar{\theta} \leftarrow \tau\bar{\theta} + (1-\tau)\theta \tag{8}$$

where τ is a decay factor close to 1. The teacher distribution is then defined using $\pi_{\bar{\theta}}$ instead of the instantaneous student parameters. This design reduces high-frequency fluctuations in the teacher signal and mitigates instability caused by rapidly changing policy distributions, which is especially critical in long-horizon video reasoning.

Finally, we optimize the policy using the GRPO-style clipped surrogate objective:

$$\mathcal{J}_{\text{VISD}} = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\min\left(\rho_t^{(i)}\hat{A}_t^{(i)}, \text{clip}(\rho_t^{(i)}, 1-\epsilon_{\text{pg}}, 1+\epsilon_{\text{pg}})\hat{A}_t^{(i)}\right)\right] \tag{9}$$

where $\rho_t^{(i)}$ is the importance sampling ratio between the current and old policies for token t in rollout i, and ϵ_{pg} is the policy-gradient clipping range. We do not add a reference-model KL penalty or a separate distillation loss; the teacher only modifies the magnitude of the GRPO advantage. This preserves the stability benefits of trust-region optimization while incorporating structured token-level credit assignment. These mechanisms together enable VISD to effectively scale structured self-distillation to long video sequences, achieving stable optimization despite the complexity of multi-dimensional rewards and long-horizon dependencies.

4 Experiments

4.1 Experimental Setup

Benchmarks. We evaluate VISD on the Open-o3-Video [33] benchmark suite, which consists of diverse video reasoning tasks covering temporal understanding, object grounding, and compositional reasoning. The benchmarks include both short and long video scenarios, requiring models to reason over dynamic visual content and produce multi-step explanations. We also evaluate on Video-MME-v2 [12], which provides a more challenging video understanding setting with fine-grained level-wise and consistency-oriented metrics. We additionally evaluate zero-shot temporal grounding on Charades-STA [14].

Implementation Details. VISD starts from the same SFT initialization recipe as Open-o3-Video and then applies our self-distillation RL training for video reasoning. The SFT initialization establishes the grounded reasoning format, while the VISD RL stage optimizes answer correctness and grounding quality on STGR-RL prompts through feedback-conditioned teacher replay. During RL, VISD uses feedback-conditioned teacher replay only as a training-time token-credit signal. Unless otherwise specified, all VISD (Ours) results are evaluated with the student checkpoint at optimization step 1200. The Open-o3-Video and VisionCoach baselines are evaluated using their fully trained RL checkpoints, which require more than 2300 optimization steps; thus, VISD reaches stronger performance with about 50% of the optimization steps. More implementation, training, and evaluation details are provided in Appendices C.1, C.2, and C.3.

Table 1: **V-STAR results.** Chain1 denotes *what–when–where*, while Chain2 denotes *what–where–when*. * indicates rows evaluated by us.

Model	What	When (Temporal IoU)		Where (Spatial IoU)		Overall	
	Acc	Chain1	Chain2	Chain1	Chain2	mAM	mLGM
Proprietary Models							
Gemini-2-Flash	53.0	24.5	23.8	4.6	2.2	26.9	35.6
GPT-4o	60.8	16.7	12.8	6.5	3.0	26.8	38.2
Open-source Models							
TRACE	17.6	19.1	17.1	0.0	0.0	12.0	13.3
Oryx-1.5-7B	20.5	13.5	14.8	10.1	3.5	15.1	13.8
VideoChat2	36.2	13.7	12.5	2.5	1.0	17.0	20.3
Qwen2.5-VL-7B*	33.5	15.4	13.8	17.0	2.5	19.3	22.4
InternVL-2.5-8B	44.2	8.7	7.8	0.7	0.1	17.6	24.9
Video-LLaMA3	41.9	23.0	23.1	0.9	0.2	21.7	27.0
LLaVA-Video	49.5	10.5	12.2	1.9	1.3	20.8	27.3
Open-o3-Video-7B*	60.1	25.0	23.6	25.6	5.6	33.4	45.8
VisionCoach-7B*	61.2	25.4	25.4	**27.1**	5.3	34.3	47.5
VISD (Ours)	**61.9**	**26.8**	**27.2**	26.7	**6.3**	**35.1**	**48.9**
Δ vs. Qwen2.5-VL-7B	+28.4	+11.4	+13.4	+9.7	+3.8	+15.8	+26.5

Table 2: **General benchmark results.** "LRR" refers to LongVideo-Reason-eval [7]. The Average score is computed over the bold-faced dataset-level metrics. * indicates rows evaluated by us; for rows marked with †, the WorldSense scores are evaluated by us.

Model	WorldSense				VideoMMMU				LRR	TVGBench	Average
	Overall	Recognition	Understanding	Reasoning	Overall	Comprehension	Adaptation	Perception	Acc	mIoU	Score
Proprietary Models											
GPT-4o	42.6	-	-	-	61.2	-	-	66.0	-	-	-
Tool-calling Methods											
LongVT-RL-7B*	36.1	33.1	39.1	37.8	52.1	49.3	43.0	64.0	75.1	10.6	43.5
Tool-free Methods											
Qwen2.5-VL-7B†	35.5	32.9	38.4	37.0	51.2	-	-	64.7	59.3	16.3	40.6
VideoRFT-7B†	38.4	36.5	40.0	40.1	51.1	-	-	66.0	69.4	14.3	43.3
VideoR1-7B†	36.9	35.3	37.7	39.0	52.4	-	-	65.3	68.9	9.6	42.0
Open-o3-Video-7B*	38.9	37.4	39.7	40.5	52.6	49.1	41.3	67.4	69.1	19.2	44.9
VisionCoach-7B*	39.2	38.4	40.2	39.6	**53.8**	49.0	42.2	**70.3**	70.7	21.0	46.2
VISD (Ours)	**39.9**	**39.4**	**40.4**	**40.5**	**53.8**	**49.7**	**42.3**	69.3	**73.5**	**24.8**	**48.0**
Δ vs. Qwen2.5-VL-7B	+4.4	+6.5	+2.0	+3.5	+2.6	-	-	+4.6	+14.2	+8.5	+7.4

4.2 Main Results

Spatio-Temporal Grounding on V-STAR [9]. Table 1 compares VISD with closed-source models [21, 10], general video LLMs [4], and recent reasoning-focused frameworks [11, 33, 22]. VISD improves answer accuracy over Qwen2.5-VL-7B by +28.4 points and obtains the best overall V-STAR scores, improving mAM/mLGM over VisionCoach from 34.3/47.5 to 35.1/48.9. The gains appear not only in final answering, but also in temporal localization and spatial grounding, which is the setting where sparse sequence-level rewards are least informative.

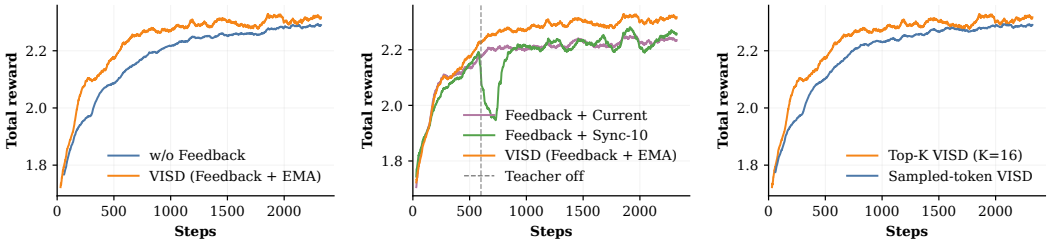
General Video Understanding. Beyond V-STAR, VISD also improves the average score across broader video reasoning benchmarks [12, 19, 18, 7, 45]. As shown in Table 2, VISD outperforms Qwen2.5-VL-7B by +7.4 average points and improves over VisionCoach by +1.8 average points, with the strongest gains on long-video reasoning and temporal grounding metrics such as LRR and TVGBench. On Video-MME-v2, VISD also improves the non-linear score from 8.6 to 9.4 over VisionCoach and obtains the best average accuracy (Table 3). These results suggest that feedback-conditioned token supervision preserves general video QA ability while improving the grounding-oriented behaviors targeted by VISD. Additional qualitative examples are provided in Appendix F.

Table 4: **Charades-STA results.** We report zero-shot temporal grounding with recall at IoU thresholds 0.3, 0.5, and 0.7, together with mIoU.

Model	R@0.3	R@0.5	R@0.7	mIoU
General Video LLMs				
VideoChat	9.0	3.3	1.3	6.5
VideoLLaMA	10.4	3.8	0.9	7.1
Video-ChatGPT	20.0	7.7	1.7	13.7
Valley	28.4	1.8	0.3	21.4
Temporal Grounding Video LLMs				
GroundingGPT	-	29.6	11.9	-
Momentor	42.6	26.6	11.6	28.5
TimeChat	-	32.2	13.4	-
HawkEye	50.6	31.4	14.5	33.7
VTG-LLM	51.2	33.8	15.7	34.4
VTimeLLM	51.0	27.5	11.4	31.2
Grounded Reasoning Models				
Open-o3-video	62.6	45.6	24.5	42.5
VisionCoach	63.2	45.8	**24.7**	42.7
VISD (Ours)	**67.3**	**46.1**	21.6	**44.6**

Table 3: **Video-MME-v2 results.** We report the official non-linear score, level-wise scores, consistency, coherence, and average accuracy using 64 sampled frames without subtitles.

Model	Frames	Non-Lin Score	Level 1	Level 2	Level 3	Consistency	Coherence	Avg Acc
Qwen2.5-VL-7B	64	7.9	11.4	6.7	6.7	8.2	7.3	21.3
VideoRFT-7B	64	8.2	11.8	7.4	6.8	8.8	7.2	21.7
Video-R1-7B	64	7.7	11.1	6.7	6.4	8.0	7.2	20.7
Open-o3-Video-7B	64	7.8	11.0	6.9	6.6	8.3	6.9	21.2
VisionCoach-7B	64	8.6	12.1	7.3	7.5	9.0	7.9	22.1
VISD (Ours)	64	**9.4**	**13.4**	**8.2**	**7.9**	**10.2**	**8.0**	**23.8**



(a) **(b)** **(c)**

Figure 3: *Training ablation curves.* **(a)** Feedback conditioning compares VISD with and without judge feedback, showing that structured feedback leads to a stronger final reward trajectory. **(b)** Teacher update strategy compares current-policy, Sync-10, and EMA teachers, where EMA provides the most stable optimization and best final reward. **(c)** Top-K support versus sampled token compares two teacher-student credit formulations, showing a modest but consistent advantage for VISD's local top-K support.

Temporal Grounding on Charades-STA. On Charades-STA [14], VISD improves R@0.3, R@0.5, and mIoU over Open-o3-Video and VisionCoach (Table 4), indicating stronger zero-shot moment localization. This is a useful stress test because the model must output a temporal segment without task-specific fine-tuning on the target benchmark.

4.3 Ablation Studies

We study three questions that correspond to the main design choices in VISD. First, does structured judge feedback provide additional benefit beyond using verified answer information alone? Second, does the improvement come from a particular teacher-parameter update heuristic, or is the update rule mainly a stabilization choice? Third, is it sufficient to compare the teacher only on the realized sampled token, or does VISD's local top-K support provide a more useful token-level magnitude signal? We pair reward curves with downstream benchmark tables because training reward alone may not fully reflect grounding quality.

Table 5 reports the final training statistics corresponding to the ablation curves in Figure 3. Reward and answer accuracy summarize final optimization quality, while gradient norm serves as a stability diagnostic for teacher-update strategies.

Table 5: *Training ablation on feedback conditioning and teacher update strategy.* We report final training reward, answer accuracy, and gradient norm for the main feedback and teacher-parameter variants, averaged over the final 100 training steps.

Method	Feedback	Teacher	Reward ↑	Ans. Acc. ↑	Grad Norm ↓
VISD w/o Feedback	No	EMA	2.280	0.565	**2.63**
VISD w/ current-policy teacher	Yes	Current	2.228	0.557	3.49
VISD w/ Sync-10 teacher	Yes	Sync-10	2.253	0.558	11.63
VISD w/ EMA teacher (Ours)	Yes	EMA	**2.318**	**0.571**	3.58

Structured feedback. The no-feedback variant keeps the same policy-gradient objective and teacher replay, but conditions the teacher only on verified answer-side information. Structured feedback improves final reward from 2.280 to 2.318 and answer accuracy from 0.565 to 0.571 (Figure 3**(a)** and Table 5). The same trend appears in Tables 6 and 7, where the full VISD configuration obtains the best overall VSTAR score and the best average general-video score. This suggests that the benefit is not merely from exposing the verified answer to the teacher; the judge feedback contributes diagnostic context that changes how token-level update magnitudes are distributed along the rollout. Because

Table 6: *VSTAR ablation results.* We compare feedback/teacher variants and top-K versus sampled-token credit on answer accuracy, temporal IoU, spatial IoU, and overall VSTAR metrics.

Model	What	When (Temporal IoU)		Where (Spatial IoU)		Overall	
	Acc	Chain1	Chain2	Chain1	Chain2	mAM	mLGM
Ablation Variants							
VISD w/o Feedback	59.7	25.2	24.8	23.7	5.5	33.1	45.3
VISD w/ current-policy teacher	60.2	23.8	21.7	23.5	5.7	32.5	44.8
VISD w/ Sync-10 teacher	58.6	**28.8**	**27.3**	11.2	2.7	31.2	42.8
VISD w/ EMA teacher (Ours)	**61.9**	26.8	27.2	**26.7**	**6.3**	**35.1**	**48.9**
Top-K Analysis							
VISD w/ sampled token	61.3	25.2	25.4	24.8	6.1	34.0	47.0
VISD w/ top-K (Ours)	**61.9**	**26.8**	**27.2**	**26.7**	**6.3**	**35.1**	**48.9**

Table 7: *General video reasoning ablation results.* We evaluate the same ablation variants on broad video understanding and temporal-grounding benchmarks, reporting the average over dataset-level metrics.

Model	WorldSense				VideoMMMU				LRR	TVGBench	Average
	Overall	Recognition	Understanding	Reasoning	Overall	Comprehension	Adaptation	Perception	Acc	mIoU	Score
Ablation Variants											
VISD w/o Feedback	38.8	37.8	39.1	40.2	52.6	49.0	38.7	70.0	73.4	22.3	46.8
VISD w/ current-policy teacher	39.2	37.8	**40.6**	40.1	52.0	48.7	39.0	68.3	**75.3**	18.7	46.3
VISD w/ Sync-10 teacher	38.8	37.8	39.5	39.8	48.9	43.3	40.0	63.3	64.7	24.0	44.1
VISD w/ EMA teacher (Ours)	**39.9**	**39.4**	40.4	**40.5**	**53.8**	**49.7**	**42.3**	69.3	73.5	**24.8**	**48.0**
Top-K Analysis											
VISD w/ sampled token	38.8	38.2	39.1	39.5	**55.0**	**51.3**	**42.7**	**71.0**	72.9	23.9	47.7
VISD w/ top-K (Ours)	**39.9**	**39.4**	**40.4**	**40.5**	53.8	49.7	42.3	69.3	**73.5**	**24.8**	**48.0**

the reinforcement advantage still determines the update direction, this extra context refines credit assignment without replacing the reward signal. This mirrors the distinction made in recent self-distillation work between using richer feedback and performing denser token-level credit assignment: in VISD, the judge supplies the richer feedback, while teacher replay turns it into token-wise magnitude modulation. Additional benchmark, training-curve, and mechanism analyses are provided in Appendices D.1, D.2, and E.1.

Teacher update strategy. With judge feedback fixed, we compare a current-policy teacher, a Sync-10 teacher, and an EMA teacher (Figure 3**(b)**). The three variants are close while teacher guidance is active, suggesting that the early gain comes mainly from feedback-conditioned token credit rather than a particular synchronization trick. After the teacher weight is annealed to zero, Sync-10 shows a sharp transient drop and a much larger final gradient norm, while EMA gives the best reward, answer accuracy, VSTAR overall score, and general-video average (Tables 5, 6, and 7). We therefore view EMA as a stabilizing parameterization of the feedback-conditioned teacher, not as the primary source of VISD's gains. Additional optimization diagnostics are provided in Appendix D.3.

Top-K support vs. sampled token. We compare VISD's teacher top-K local support with a sampled-token variant that computes the teacher–student comparison only on the realized token. The sampled-token variant is viable and remains close on several general video benchmarks, but top-K maintains a stronger reward trajectory and better grounding scores, including VSTAR mAM/mLGM and TVGBench (Figure 3**(c)**; Tables 6 and 7). This pattern is consistent with the motivation of local-support distillation: the teacher top-K set gives a compact distribution-level comparison around the prefix, whereas the sampled-token variant observes only one realized point from that support.

5 Conclusion

In this work, we presented VISD, a structured self-distillation framework that addresses the challenge of fine-grained credit assignment in long-horizon video reasoning. Recognizing that errors in video QA and grounded reasoning are inherently structured—arising from temporal misalignment, logical inconsistency, or spatial grounding failures—VISD leverages a video-aware judge to generate diagnostically meaningful, token-level supervision. We integrate this privileged information into reinforcement learning through direction-magnitude decoupling mechanism, which preserves policy stability while enabling semantically aligned credit assignment. Supported by curriculum scheduling and EMA-based teacher stabilization, VISD robustly scales to complex video sequences. Extensive experiments demonstrate that VISD significantly accelerates convergence and improves accuracy,

grounding, and interpretability over strong baselines. Future work will explore more efficient judge modeling and richer feedback structures for broader grounded reasoning applications.

Acknowledgments and Disclosure of Funding

The authors received no specific funding for this work. Hao Lin gratefully acknowledges the support of Huazhong University of Science and Technology and the NeurIPS 2026 Financial Assistance Program for his attendance at NeurIPS 2026. The authors declare no competing interests.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

[2] Ruichuan An, Sihan Yang, Renrui Zhang, Zijun Shen, Ming Lu, Gaole Dai, Hao Liang, Ziyu Guo, Shilin Yan, Yulin Luo, et al. Unictokens: Boosting personalized understanding and generation via unified concept tokens. *arXiv preprint arXiv:2505.14671*, 2025.

[3] Ruichuan An, Sihan Yang, Ziyu Guo, Wei Dai, Zijun Shen, Haodong Li, Renrui Zhang, Xinyu Wei, Guopeng Li, Wenshan Wu, et al. Genius: Generative fluid intelligence evaluation suite. *arXiv preprint arXiv:2602.11144*, 2026.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Ruizhe Chen, Tianze Luo, Zhiting Fan, Heqing Zou, Zhaopeng Feng, Guiyang Xie, Hansheng Zhang, Zhuochen Wang, Zuozhu Liu, and Zhang Huaijian. Datasets and recipes for video temporal grounding via reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 983–992, 2025.

[6] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. In *The Thirteenth International Conference on Learning Representations*, 2024.

[7] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, et al. Scaling rl to long videos. *arXiv preprint arXiv:2507.07966*, 2025.

[8] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[9] Zixu Cheng, Jian Hu, Ziquan Liu, Chenyang Si, Wei Li, and Shaogang Gong. V-star: Benchmarking video-llms on video spatio-temporal reasoning. *arXiv preprint arXiv:2503.11495*, 2025.

[10] 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.

[11] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

[12] Chaoyou Fu, Haozhi Yuan, Yuhao Dong, Yi-Fan Zhang, Yunhang Shen, Xiaoxing Hu, Xueying Li, Jinsen Su, Chengwu Long, Xiaoyao Xie, et al. Video-mme-v2: Towards the next stage in benchmarks for comprehensive video understanding. *arXiv preprint arXiv:2604.05015*, 2026.

[13] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018.

[14] Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275, 2017.

[15] 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.

[16] Yongxin Guo, Jingyu Liu, Mingda Li, Qingbin Liu, Xi Chen, and Xiaoying Tang. Trace: Temporal grounding video llm via causal event modeling. *arXiv preprint arXiv:2410.05643*, 2024.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. 2015.

[18] Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. Worldsense: Evaluating real-world omnimodal understanding for multimodal llms. *arXiv preprint arXiv:2502.04326*, 2025.

[19] Kairui Hu, Penghao Wu, Fanyi Pu, Wang Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *arXiv preprint arXiv:2501.13826*, 2025.

[20] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, et al. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

[21] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

[22] Daeun Lee, Shoubin Yu, Yue Zhang, and Mohit Bansal. Visioncoach: Reinforcing grounded video reasoning via visual-perception prompting. *arXiv preprint arXiv:2603.14659*, 2026.

[23] Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026.

[24] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206, 2024.

[25] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. Videochat-r1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

[26] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026.

[27] Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-hinting language models enhance reinforcement learning. *arXiv preprint arXiv:2602.03143*, 2026.

[28] Jinshan Liu, Haoran Qin, Xiaobing Tu, Jiacheng Liu, Jiahui Hu, Zhengan Yan, Yukun Xie, Kerui Shen, Jinkui Ren, Yuqi Lin, et al. Linca: Accelerating diffusion models via learnable decomposed feature caching. In *European Conference on Computer Vision*, pages 463–481. Springer, 2026.

[29] Ruyang Liu, Chen Li, Haoran Tang, Yixiao Ge, Ying Shan, and Ge Li. St-llm: Large language models are effective temporal learners. In *European Conference on Computer Vision*, pages 1–18. Springer, 2024.

[30] Zuyan Liu, Yuhao Dong, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution. *arXiv preprint arXiv:2409.12961*, 2024.

[31] Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026.

[32] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12585–12602, 2024.

[33] Jiahao Meng, Xiangtai Li, Haochen Wang, Yue Tan, Tao Zhang, Lingdong Kong, Yunhai Tong, Anran Wang, Zhiyang Teng, Yujing Wang, et al. Open-o3 video: Grounded video reasoning with explicit spatio-temporal evidence. *arXiv preprint arXiv:2510.20579*, 2025.

[34] Kun Ouyang, Yuanxin Liu, Haoning Wu, Yi Liu, Hao Zhou, Jie Zhou, Fandong Meng, and Xu Sun. Spacer: Reinforcing mllms in video spatial reasoning. *arXiv preprint arXiv:2504.01805*, 2025.

[35] Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Simko: Simple pass@ k policy optimization. *arXiv preprint arXiv:2510.14807*, 2025.

[36] Haoran Qin, Zhengan Yan, Shikang Zheng, Xiaobing Tu, Jiacheng Liu, Yuqi Lin, Chang Zou, JinShan Liu, Peiliang Cai, Xiantao Zhang, et al. Avits: Adaptive spatiotemporal token selection for efficient dynamic-resolution generation. In *European Conference on Computer Vision*, pages 257–273. Springer, 2026.

[37] Andrei A Rusu, Sergio Gomez Colmenarejo, Caglar Gulcehre, Guillaume Desjardins, James Kirkpatrick, Razvan Pascanu, Volodymyr Mnih, Koray Kavukcuoglu, and Raia Hadsell. Policy distillation. *arXiv preprint arXiv:1511.06295*, 2015.

[38] 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.

[39] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

[40] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-xl: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169, 2025.

[41] Shangkun Sun, Ruyang Liu, Haoran Tang, Yixiao Ge, Haibo Lu, Jiankun Yang, and Chen Li. Ppllava: Varied video sequence understanding with prompt guidance. *arXiv preprint arXiv:2411.02327*, 2024.

[42] Xiaokun Sun, Zezhong Wu, Zewen Ding, and Linli Xu. Mvp: Enhancing video large language models via self-supervised masked video prediction. *arXiv preprint arXiv:2601.03781*, 2026.

[43] Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, et al. Traceable evidence enhanced visual grounded reasoning: Evaluation and methodology. *arXiv preprint arXiv:2507.07999*, 2025.

[44] Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. Videorft: Incentivizing video reasoning capability in mllms via reinforced fine-tuning. *arXiv preprint arXiv:2505.12434*, 2025.

[45] Ye Wang, Ziheng Wang, Boshen Xu, Yang Du, Kejun Lin, Zihan Xiao, Zihao Yue, Jianzhong Ju, Liang Zhang, Dingyi Yang, et al. Time-r1: Post-training large vision language model for temporal video grounding. *arXiv preprint arXiv:2503.13377*, 2025.

[46] Yi Wang, Xinhao Li, Ziang Yan, Yinan He, Jiashuo Yu, Xiangyu Zeng, Chenting Wang, Changlian Ma, Haian Huang, Jianfei Gao, et al. Internvideo2. 5: Empowering video mllms with long and rich context modeling. *arXiv preprint arXiv:2501.12386*, 2025.

[47] Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-rts: Rethinking reinforcement learning and test-time scaling for efficient and enhanced video reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28114–28128, 2025.

[48] Ziyue Wang, Sheng Jin, Zhongrong Zuo, Jiawei Wu, Han Qiu, Qi She, Hao Zhang, and Xudong Jiang. Video-ktr: Reinforcing video reasoning via key token attribution. *arXiv preprint arXiv:2601.19686*, 2026.

[49] Ziang Yan, Xinhao Li, Yinan He, Zhengrong Yue, Xiangyu Zeng, Yali Wang, Yu Qiao, Limin Wang, and Yi Wang. Videochat-r1. 5: Visual test-time scaling to reinforce multimodal reasoning by iterative perception. *arXiv preprint arXiv:2509.21100*, 2025.

[50] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

[51] Zuhao Yang, Sudong Wang, Kaichen Zhang, Keming Wu, Sicong Leng, Yifan Zhang, Bo Li, Chengwei Qin, Shijian Lu, Xingxuan Li, et al. Longvt: Incentivizing" thinking with long videos" via native tool calling. *arXiv preprint arXiv:2511.20785*, 2025.

[52] Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, et al. Video-o3: Native interleaved clue seeking for long video multi-hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026.

[53] Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, et al. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*, 2025.

[54] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. In *Proceedings of the 2023 conference on empirical methods in natural language processing: system demonstrations*, pages 543–553, 2023.

[55] Haoji Zhang, Xin Gu, Jiawen Li, Chixiang Ma, Sule Bai, Chubin Zhang, Bowen Zhang, Zhichao Zhou, Dongliang He, and Yansong Tang. Thinking with videos: Multimodal tool-augmented reinforcement learning for long video reasoning. *arXiv preprint arXiv:2508.04416*, 2025.

[56] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025.

[57] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

[58] 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.

[59] Yuanhong Zheng, Ruichuan An, Xiaopeng Lin, Yuxing Liu, Sihan Yang, Huanyu Zhang, Haodong Li, Qintong Zhang, Renrui Zhang, Guopeng Li, et al. Pearl: Personalized streaming video understanding model. *arXiv preprint arXiv:2603.20422*, 2026.

[60] Changhai Zhou, Qian Qiao, Yuhua Zhou, Yuxin Wu, Shichao Weng, WEIZHONG ZHANG, and Cheng Jin. Large language model compression with global rank and sparsity optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ZaPmQ0NHs4.

[61] Haipeng Zhou, Jinshan Liu, He Zhang, Xuequan Lu, Jun Ma, and Lei Zhu. Kiroshi: An agentic perception system for high-accuracy image parsing. In *European Conference on Computer Vision*, pages 182–200. Springer, 2026.

Appendix Contents

A Extended Related Work **16**

B Limitations and Future Works **17**

C Implementation and Evaluation Details **17**

C.1 Algorithm Details 17

C.2 Training Details 19

C.3 Evaluation Details 20

D Additional Experimental Results **21**

D.1 Benchmark-Level Ablation Results 21

D.2 Additional Training Curves 21

D.3 Teacher Stability Diagnostics 22

D.4 Training Details for Ablation Settings 24

E Mechanistic Analysis and Qualitative Diagnostics **24**

E.1 How Judge Feedback Affects Credit Assignment 24

E.2 Trajectory-Dependent Judge Feedback 24

E.3 Top-K Local Support Ratio 24

E.4 Feedback-Conditioned Credit as an Information Layer 25

F Qualitative Case Studies **29**

G Judge Prompt Template **34**

G.1 System Prompt 34

G.2 User Prompt Assembly 35

H Broader Impacts **35**

A Extended Related Work

Video large language models. Recent advances in video large language models (VideoLLMs) have extended the success of vision-language models [3, 2] to dynamic visual understanding [32, 54, 56, 53, 29, 59]. Existing approaches typically build upon pretrained image-language backbones and incorporate temporal modeling through frame sampling [56, 53], temporal attention modules [46], or lightweight adapters that aggregate information across frames [6, 40]. These designs have enabled strong performance on a wide range of video tasks [12, 24, 19, 9, 41], including question answering, captioning, and temporal reasoning. Despite these improvements, most VideoLLMs are trained using supervised fine-tuning or imitation learning on static annotations [56, 53]. While effective for aligning model outputs with ground-truth answers, such training paradigms treat reasoning as a one-shot prediction problem and do not explicitly model the process of multi-step reasoning over time. As a result, models often produce superficially correct answers that are not faithfully grounded in the underlying video content, especially in long or complex scenarios where errors may arise from incorrect temporal alignment or missing visual evidence [45, 5, 34, 43]. Several recent works attempt to address these limitations by incorporating iterative reasoning or external feedback signals [25, 44, 47, 55, 61]. However, these methods typically rely on heuristic supervision or unstructured feedback, which limits their ability to provide precise guidance for correcting intermediate reasoning steps. Consequently, enabling fine-grained and interpretable credit assignment remains an open challenge in VideoLLM training.

Reinforcement learning for multimodal reasoning. Reinforcement learning has emerged as a powerful paradigm for improving reasoning capabilities in large language models by optimizing policies against task-level rewards [15, 38, 60]. In particular, reinforcement learning with verifiable rewards (RLVR) provides a principled way to incorporate correctness signals derived from ground-truth answers or external verifiers. This paradigm has been increasingly applied to multimodal settings [11, 25, 44], where reward functions may jointly consider answer accuracy, grounding consistency, and other task-specific criteria. While reinforcement learning offers reliable optimization signals, its effectiveness is fundamentally constrained by the granularity of reward design [15, 38, 50]. In most cases, rewards are defined at the sequence level, providing only a scalar signal for an entire reasoning trajectory. This leads to coarse credit assignment, making it difficult to identify which parts of a long reasoning sequence are responsible for success or failure. The problem is particularly severe in video reasoning, where errors are often localized and structured, such as referencing an incorrect time window or misidentifying objects in specific frames. Recent efforts have explored combining reinforcement learning with auxiliary supervision to improve credit assignment [57, 20, 50], including approaches that incorporate token-level guidance or policy regularization. However, these methods typically treat auxiliary signals as generic constraints or smoothness priors, without explicitly modeling the structure of reasoning errors [57, 20, 50]. As a result, they struggle to provide targeted corrections in complex multimodal scenarios.

Self-distillation and fine-grained credit assignment. Self-distillation provides an alternative approach to improving sequence models by introducing dense, token-level supervision derived from teacher signals [17, 13, 37, 50]. Among them, RLSD explores direction–magnitude decoupled self-distillation in scientific reasoning, while VISD adapts this principle to video reasoning, where feedback must account for answer correctness, reasoning consistency, temporal alignment, and spatial grounding. In the context of reasoning tasks, recent methods leverage self-distillation to refine intermediate predictions, often by comparing model outputs under different conditions or by integrating teacher policies into reinforcement learning frameworks [57, 20, 50]. These approaches enable more fine-grained credit assignment than pure reinforcement learning, and have been shown to improve training efficiency and reasoning quality. However, existing self-distillation methods largely rely on unstructured or modality-agnostic signals [57, 20, 50]. The teacher guidance is typically expressed as a distributional difference or preference signal, without explicitly capturing the underlying causes of reasoning errors. Unlike on-policy distillation methods that introduce dense distribution-matching or reverse-KL-style objectives, VISD does not add a separate distillation loss; the teacher signal only reweights the magnitude of the on-policy gradient while the reward advantage determines its direction. This limitation becomes more pronounced in video reasoning, where failures may stem from diverse sources [33, 45, 34, 43], including logical inconsistency, temporal misalignment, and incorrect visual grounding. Without distinguishing between these factors, token-level supervision may lack diagnostic specificity, leading to suboptimal learning dynamics.

Furthermore, integrating self-distillation with reinforcement learning introduces additional challenges. Naively combining the two can result in unstable optimization, as auxiliary signals may conflict with reward-driven updates or dominate the learning process. This highlights the need for a principled framework that not only provides fine-grained supervision, but also aligns it with the underlying structure of multimodal reasoning and preserves the stability of policy optimization [28, 36].

B Limitations and Future Works

Our current evaluation mainly focuses on publicly available benchmarks for video reasoning and grounding. While these benchmarks cover short-video reasoning, long-video understanding, temporal grounding, and spatial grounding, they still cannot fully capture the diversity and noise of real-world video use cases. Future work can therefore extend VISD to more open-ended, noisy, and complex real-world video tasks, including settings with weaker supervision, more diverse video quality, and less standardized answer formats.

Our current method mainly focuses on video reasoning with explicit temporal and spatial grounding signals. This setting is a natural testbed for studying fine-grained credit assignment, but it does not exhaust the full range of structured outputs that VideoLLMs may need to produce. Future work can extend VISD to richer structured outputs, such as denser event descriptions or more compositional grounded reasoning formats, as well as more open-ended interactive video tasks.

C Implementation and Evaluation Details

C.1 Algorithm Details

Teacher and judge roles. The feedback-conditioned teacher is initialized from the same SFT-initialized model as the student. For each sampled student rollout, an external LLM judge reads the question, the student completion, the verified answer, and available grounding evidence, then produces structured process feedback about answer correctness, reasoning consistency, temporal alignment, and spatial grounding. The judge uses these video-specific cues to diagnose the student's rollout, but the final training reward is still computed by the original verifier from the ground-truth answer and grounding annotations. The full judge prompt template is provided in Appendix G. In the main experiments, we use GPT-5.4 as the judge because it provides strong instruction following and reasoning capabilities for diagnosing long video reasoning trajectories. The judge is used only during training: it does not replace the verifier reward, does not provide gradients, and is not used at inference time. Instead, its feedback is treated as privileged diagnostic context that tells the teacher not only whether a rollout is correct, but also how its reasoning or grounding succeeds or fails. The teacher does not generate a new answer; it replays the same student completion under the hidden answer/evidence/feedback context and provides stop-gradient token scores. The resulting teacher-student token score is used only to reweight the on-policy gradient magnitude, while the rollout-level advantage determines the update direction.

Reward. The verifier reward is consistent with Open-o3-Video [33] and VisionCoach [22], covering answer correctness, format correctness, final-answer grounding, and reasoning-trace grounding. VISD does not introduce a new reward source; instead, it changes how the resulting rollout reward is converted into token-level credit through feedback-conditioned teacher reweighting. Each sampled rollout is scored by this verifier before computing the group-relative advantage. For a prompt x, completion y, verified answer $a^\star$, and available grounding evidence e, the scalar reward is the sum of the applicable components:

$$\begin{aligned} R(x, y, a^\star, e) &= \sum_{k \in \mathcal{K}(x)} \alpha_k r_k(x, y, a^\star, e), \\ \mathcal{K}(x) &\subseteq \{\text{ans}, \text{fmt}, \text{ans-tmp}, \text{ans-spa}, \text{thk-tmp}, \text{thk-spa}\}. \end{aligned} \tag{10}$$

Here $\mathcal{K}(x)$ masks out reward terms that are not supported by the current sample, and the main experiments use the configured component weights α_k when summing the rollout reward. The answer reward checks exact correctness for multiple-choice questions and ROUGE-based similarity for free-form answers. For temporal and spatial grounding answers, the verifier additionally computes temporal IoU between parsed answer intervals and ground-truth intervals, and visual IoU between

parsed answer boxes and ground-truth boxes. The format reward checks the structured response format, including paired <think> and <answer> tags and valid grounding tags such as <obj>, <box>, and <t> when grounding is required.

For reasoning traces, VISD parses timestamps and object-box-time tuples from the <think> section. When interval supervision is available, the thinking temporal reward measures the fraction of parsed timestamps that fall inside the annotated interval:

$$r^{\text{seg}}_{\text{thk-tmp}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}[s^\star \le t_m \le e^\star]\,. \tag{11}$$

When key-frame supervision is available, it instead uses an adaptive proximity score to the nearest annotated time:

$$r^{\text{pt}}_{\text{thk-tmp}} = \frac{1}{M}\sum_{m=1}^{M} \exp\left(-\frac{\min_j |t_m - t_j^\star|^2}{2\sigma_s^2}\right), \tag{12}$$

where σ_s is annealed during training so that early learning tolerates coarse temporal localization and later learning encourages tighter alignment. For thinking spatial reward, a predicted tuple contributes only when its timestamp matches an annotated key frame within a small temporal tolerance; the reward then averages IoU scores between predicted boxes and ground-truth boxes on the matched frame:

$$r_{\text{thk-spa}} = \frac{1}{Z}\sum_{m\in\mathcal{I}} \max_{b^\star\in\mathcal{B}^\star(t_m)} \text{IoU}(b_m, b^\star), \qquad \mathcal{I} = \{m : \min_j |t_m - t_j^\star| \le \tau\}. \tag{13}$$

If no valid prediction or supervision is available for a component, that component contributes zero or is masked out. The summed rollout reward in Eq. (10) is then used to compute the standard group-relative advantage in Eq. (14); in the main VISD experiments, judge feedback and teacher replay do not create separate reward components or change the reward-determined advantage direction.

GRPO-style VISD objective. For each prompt x, the old policy samples a group of rollouts $\{y^{(i)}\}_{i=1}^{G}$. After scoring each rollout with the verifier, VISD first computes the standard group-relative advantage:

$$A^{(i)} = \frac{R(x, y^{(i)}) - \mu_G}{\sigma_G + \epsilon}, \qquad \mu_G = \frac{1}{G}\sum_{j=1}^{G} R(x, y^{(j)}), \tag{14}$$

where σ_G is the standard deviation of rewards within the same rollout group. At token position t of rollout i, let $\Delta_t^{(i)}$ denote the stop-gradient top-K teacher-student log-ratio on the realized token, as defined in Appendix E.3. VISD converts this ratio into a positive token modifier and reweighted advantage:

$$m_t^{(i)} = \text{clip}\Big(\exp\left(\text{sign}(A^{(i)})\Delta_t^{(i)}\right), 1-\epsilon_w, 1+\epsilon_w\Big), \qquad \hat{A}_t^{(i)} = A^{(i)}\left[(1-\lambda_s) + \lambda_s m_t^{(i)}\right]. \tag{15}$$

The final policy objective keeps the GRPO clipped surrogate form, but replaces the rollout-level advantage with the token-reweighted advantage:

$$\mathcal{J}_{\text{VISD}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\min\Big(\rho_t^{(i)}(\theta)\hat{A}_t^{(i)}, \text{clip}\Big(\rho_t^{(i)}(\theta), 1-\epsilon_{\text{pg}}, 1+\epsilon_{\text{pg}}\Big)\hat{A}_t^{(i)}\Big)\Bigg]. \tag{16}$$

Here the token-level importance ratio is

$$\rho_t^{(i)}(\theta) = \frac{\pi_\theta(y_t^{(i)} \mid x, y_{<t}^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)} \mid x, y_{<t}^{(i)})}. \tag{17}$$

The implementation minimizes $-\mathcal{J}_{\text{VISD}}$. When $\lambda_s = 0$, $\hat{A}_t^{(i)} = A^{(i)}$, so VISD recovers the standard GRPO clipped surrogate.

Algorithm 1 VISD training procedure with feedback-conditioned teacher replay

Require: Student policy π_θ, EMA teacher $\pi_{\bar{\theta}}$, RL data $\mathcal{D} = \{(x, a^\star, e)\}$, judge J, verifier R, group size G, top-K size K, VISD mixing schedule λ_s, credit clip bound ϵ_w

for each training iteration s **do**
  Set rollout policy $\theta_{\text{old}} \leftarrow \theta$
  Sample a batch of video-question prompts $\{x\}$ from $\mathcal{D}$
  for each prompt x with verified answer/evidence $(a^\star, e)$ **do**
  // Step 1: on-policy video reasoning rollouts
    Sample G completions $y^{(1)}, \ldots, y^{(G)} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$
  // Step 2: rollout-level reward and update direction
    for $i = 1, \ldots, G$ **do**
      Obtain reward $R^{(i)} \leftarrow R(x, y^{(i)}, a^\star, e)$
    end for
    Compute group-relative advantages $A^{(i)} \leftarrow (R^{(i)} - \mu_G)/(\sigma_G + \epsilon)$
  // Step 3: trajectory-dependent diagnosis
    for $i = 1, \ldots, G$ **do**
      Generate structured feedback $f^{(i)} \leftarrow J(x, y^{(i)}, a^\star, e)$
      Set privileged context $r^{(i)} \leftarrow (a^\star, e, f^{(i)})$ and replay the same completion $y^{(i)}$
      for $t = 1, \ldots, |y^{(i)}|$ **do**
        Construct $\mathcal{U}_t^{(i)} \leftarrow \mathrm{TopK}_{P_T(\cdot \mid x, r^{(i)}, y_{<t}^{(i)})}(K) \cup \{y_t^{(i)}\}$
        Renormalize teacher and student distributions on $\mathcal{U}_t^{(i)}$
        Compute the top-K support log-ratio as in Eq. (4)
        $\Delta_t^{(i)} \leftarrow \mathrm{sg}\Big[\log \widetilde{P}_T(y_t^{(i)} \mid x, r^{(i)}, y_{<t}^{(i)}) - \log \widetilde{P}_S(y_t^{(i)} \mid x, y_{<t}^{(i)})\Big]$
        Convert to clipped token weight $m_t^{(i)} \leftarrow \mathrm{clip}\Big(\exp(\mathrm{sign}(A^{(i)})\Delta_t^{(i)}), 1 - \epsilon_w, 1 + \epsilon_w\Big)$
        Set token advantage $\hat{A}_t^{(i)} \leftarrow A^{(i)} \cdot \Big((1 - \lambda_s) + \lambda_s m_t^{(i)}\Big)$
      end for
    end for
  end for
  // Step 4: direction-preserving policy update
  Update θ with the clipped policy-gradient objective using $\{\hat{A}_t^{(i)}\}$
  Update EMA teacher $\bar{\theta} \leftarrow \tau\bar{\theta} + (1 - \tau)\theta$ and anneal λ_s
end for

C.2 Training Details

Training data and initialization. VISD uses the same SFT initialization recipe as Open-o3-Video, and our main method is applied in the subsequent RL stage. The SFT initialization uses STGR-CoT-30k, which contains four groups of grounded reasoning data: temporal grounding CoT samples, spatial grounding samples, spatio-temporal reasoning samples curated from temporal grounding and dense-captioning sources, and general video reasoning CoT samples. This initialization stage teaches the model to produce the grounded reasoning format used throughout our experiments, including temporal references, object mentions, bounding boxes, and final answers. Starting from this SFT-initialized model, we further optimize VISD on STGR-RL-36k with on-policy rollouts. The RL data follows the same broad task coverage and includes temporal grounding, spatial grounding, spatio-temporal reasoning, and general video QA prompts. Each RL example contains a video, a question, a verified answer, and, when available, grounding-related metadata such as key frames, key objects, temporal segments, or spatial regions. The RL stage optimizes answer correctness and spatio-temporal grounding under task-specific rewards, while VISD introduces judge feedback and teacher replay as training-time privileged signals for token-level credit reweighting.

Optimization details. We train VISD on 2 nodes with 8 NVIDIA A100 GPUs per node, using DeepSpeed ZeRO-3 and FlashAttention-2. Both the SFT initialization stage and the RL stage are trained for one epoch with learning rate 1×10^{-6}. For each prompt, the policy samples $G = 4$ on-policy completions. The maximum prompt length is 16,384 tokens and the maximum completion length is 768 tokens. We use a per-device training batch size of 1, gradient accumulation of 1, learning rate 1×10^{-6}, weight decay 0.01, and gradient clipping with maximum norm 5. The maximum visual

input resolution is controlled by a pixel budget of 401,408 pixels during training. The main VISD result evaluates the student policy from the EMA-teacher training setting at optimization step 1200.

VISD training configuration. While teacher guidance is active, we maintain an EMA teacher and update it every optimization step with update rate 0.01. We do not use a reference model, a teacher KL loss, or a separate distillation loss in the main VISD setting. The VISD top-K support size is $K = 16$, and the token weight is clipped to $[0.8, 1.2]$. The VISD mixing coefficient starts from 0.5 and linearly anneals to 0 over the first 600 optimization steps, gradually reducing the objective to standard on-policy RL. Table 8 summarizes the main model configuration and optimization hyperparameters used in VISD.

Table 8: *Main model configuration and training hyperparameters for VISD.* The table summarizes the backbone, teacher/evaluator setup, rollout configuration, token-credit reweighting, and optimization hyperparameters.

Parameter	Value
Model, teacher, and evaluator	
Base VideoLLM	Qwen2.5-VL-7B
Teacher initialization	Same as student initialization
Teacher update	EMA, update rate 0.01
VSTAR answer evaluator	Qwen2.5-72B-Instruct
Rollout and sequence length	
Rollouts per prompt	4
Max prompt length	16,384
Max completion length	768
Training pixel budget	401,408
VISD credit reweighting	
Top-K support size	16
Token-weight clip range	$[0.8, 1.2]$
Initial VISD mixing coefficient	0.5
Mixing annealing steps	600
Reference-model KL	Not used
Auxiliary distillation loss	Not used
Optimization	
SFT epochs	1
RL epochs	1
Per-device batch size	1
Gradient accumulation	1
Optimizer	AdamW
Learning rate	1×10^{-6}
Weight decay	0.01
Max gradient norm	5
Precision	bfloat16

C.3 Evaluation Details

Baselines. We compare VISD with both proprietary and open-source video reasoning models. For VSTAR, we include proprietary models such as GPT-4o [21] and Gemini-2-Flash [10], as well as open-source VideoLLMs including TRACE [16], Oryx-1.5-7B [30], VideoChat2 [24], Qwen2.5-VL-7B [4], InternVL-2.5-8B [8], Video-LLaMA3 [53], LLaVA-Video [56], Open-o3-Video [33], and VisionCoach [22]. For broader video understanding benchmarks, we additionally compare against recent reasoning-oriented or tool-based baselines, including LongVT-RL [51], VideoRFT [44], and Video-R1 [11]. For zero-shot temporal grounding on Charades-STA [14], we report representative general VideoLLMs and temporal-grounding VideoLLMs, following the standard comparison protocol used in prior work. Rows marked with * are evaluated by us under the same inference pipeline, while rows marked with † combine available reported results with our own evaluations when necessary.

VSTAR metrics. VSTAR evaluates grounded video reasoning along three dimensions: *what*, *when*, and *where*. The *what* dimension is measured by answer accuracy. The *when* dimension is measured by temporal IoU between predicted and ground-truth temporal intervals. The *where*

dimension is measured by spatial IoU between predicted and ground-truth bounding boxes. VSTAR reports two reasoning chains: Chain1 follows the order *what–when–where*, while Chain2 follows *what–where–when*. For each chain, the arithmetic mean is computed as

$$\mathrm{AM} = \frac{1}{3}\left(\mathrm{Acc} + \mathrm{tIoU} + \mathrm{sIoU}\right), \tag{18}$$

and VSTAR further reports mAM by averaging AM over the two chains. The benchmark also reports mLGM, a logarithmic geometric-style aggregate that reduces the collapse caused by any single near-zero grounding dimension. Together, mAM and mLGM summarize answer correctness, temporal localization, and spatial grounding into overall spatio-temporal reasoning scores.

General video understanding metrics. For WorldSense [18], VideoMMMU [19], Video-MME-v2 [12], and LongVideo-Reason [7], we follow the official benchmark protocols and report the benchmark-level accuracy or score used by each dataset. For WorldSense and VideoMMMU, we additionally report their category-level scores when available. For Video-MME-v2 [12], we report the official non-linear score, level-wise scores, consistency, coherence, and average accuracy using 64 sampled frames without subtitles.

Temporal grounding metrics. For TVGBench [45] and Charades-STA [14], we evaluate temporal grounding using temporal Intersection over Union (tIoU) between the predicted and ground-truth intervals. For Charades-STA, we report Recall at IoU thresholds 0.3, 0.5, and 0.7, together with mean IoU. For TVGBench, we report mean IoU following the benchmark protocol.

Evaluation protocol. All evaluations are conducted on 8 NVIDIA A100 GPUs. For VSTAR [9], we follow the official evaluation pipeline with 16 sampled frames and report answer accuracy, temporal IoU, spatial IoU, mAM, and mLGM. The generated VSTAR answers are evaluated by a Qwen2.5-72B-Instruct judge following the benchmark protocol. For Video-MME-v2 [12], we sample 64 frames without subtitles and report the official non-linear score, level-wise scores, consistency, coherence, and average accuracy. For WorldSense [18], VideoMMMU [19], Video-MME-v2 [12], LongVideo-Reason [7], TVGBench [45], and Charades-STA [14], we use the corresponding official or benchmark-provided evaluation scripts, keeping decoding deterministic for all evaluated models.

D Additional Experimental Results

D.1 Benchmark-Level Ablation Results

Tables 9 and 10 provide additional benchmark-level ablation results for Video-MME-v2, Charades-STA, and TVGBench. The VSTAR and general video reasoning ablation results are reported in Section 4.3. These results complement the training curves in Section 4.3: the benchmark scores are not meant to replace the main comparison, but to check whether the same design choices remain visible under downstream evaluation.

Table 9: *Video-MME-v2 ablation results.* All evaluated ablation models use 64 sampled frames without subtitles. We separate teacher/feedback ablations from the top-K analysis because they vary different design axes.

Model	Frames	Non-Lin Score	Level 1	Level 2	Level 3	Consistency	Coherence	Avg Acc
Ablation Variants								
VISD w/o Feedback	64	7.6	11.0	6.4	6.5	8.1	6.7	20.9
VISD w/ current-policy teacher	64	7.7	10.2	7.0	6.7	8.0	6.9	21.2
VISD w/ Sync-10 teacher	64	8.1	12.6	6.7	6.5	9.1	6.3	21.7
VISD w/ EMA teacher (Ours)	64	**9.4**	**13.4**	**8.2**	**7.9**	**10.2**	**8.0**	**23.8**
Top-K Analysis								
VISD w/ sampled token	64	8.1	11.9	6.7	6.9	8.6	7.2	21.5
VISD w/ top-K (Ours)	64	**9.4**	**13.4**	**8.2**	**7.9**	**10.2**	**8.0**	**23.8**

D.2 Additional Training Curves

Top-K support versus sampled token. Figure 4 provides the component-level training curves for the top-K support and sampled-token variants discussed in Section 4.3. The sampled-token variant remains competitive on several signals, but the top-K support gives a slightly stronger and more

Table 10: *Detailed temporal-grounding ablation results.* We report Charades-STA and TVGBench temporal-grounding metrics. Teacher/feedback ablations and top-K analysis are shown as separate blocks.

Model	Charades-STA				TVGBench			
	R@0.3	R@0.5	R@0.7	mIoU	R@0.3	R@0.5	R@0.7	mIoU
Ablation Variants								
VISD w/o Feedback	60.5	40.7	18.5	40.2	32.5	21.4	11.8	22.3
VISD w/ current-policy teacher	48.8	27.5	11.2	31.8	28.2	16.4	7.0	18.7
VISD w/ Sync-10 teacher	66.6	42.8	19.2	43.2	**35.6**	22.2	10.4	24.0
VISD w/ EMA teacher (Ours)	**67.3**	**46.1**	**21.6**	**44.6**	34.5	**24.5**	**13.4**	**24.8**
Top-K Analysis								
VISD w/ sampled token	63.6	41.4	18.8	41.7	35.0	21.5	12.8	23.9
VISD w/ top-K (Ours)	**67.3**	**46.1**	**21.6**	**44.6**	34.5	**24.5**	**13.4**	**24.8**

stable trend on total reward and several grounding-related components, matching the benchmark-level differences in Tables 6, 7, 9, and 10.

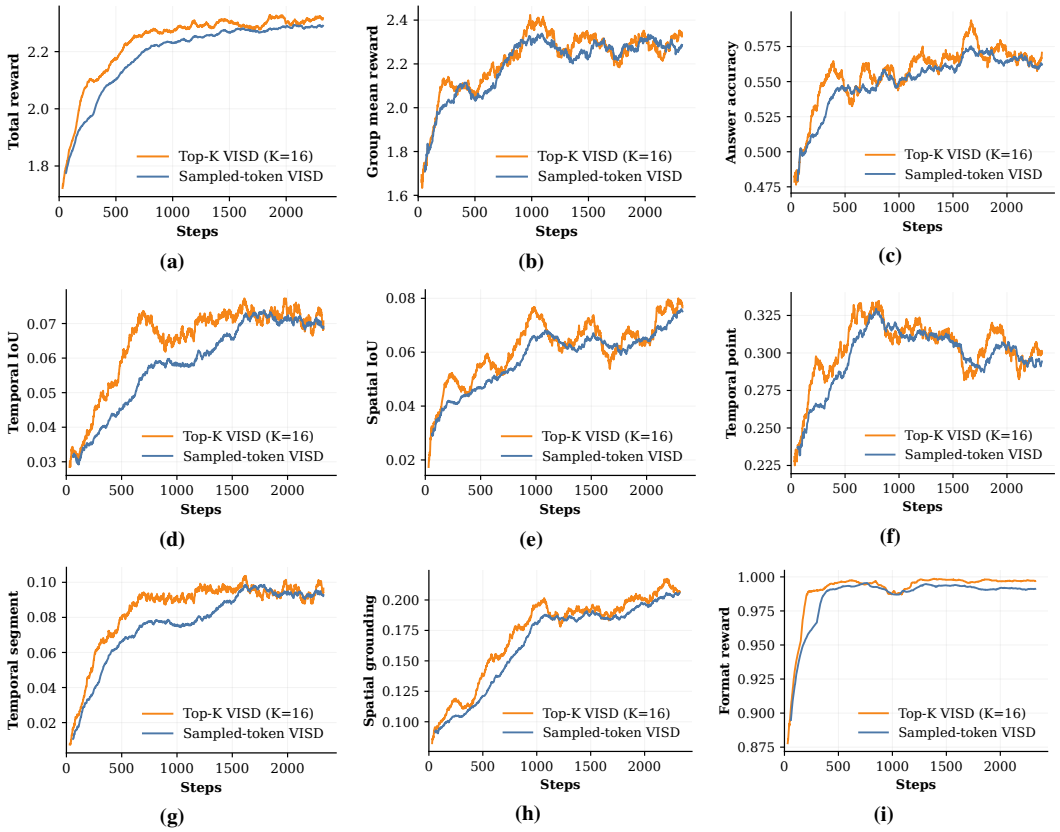


Figure 4: *Top-K support versus sampled-token training curves.* We compare **(a)** total reward, **(b)** group mean reward, **(c)** answer accuracy, **(d)** temporal IoU, **(e)** spatial IoU, **(f)** temporal point, **(g)** temporal segment, **(h)** spatial grounding, and **(i)** format reward.

Feedback conditioning. Figure 5 compares the feedback and no-feedback variants across individual reward components. We report each component as a separate small panel rather than a dense single-axis multi-panel plot for readability. These curves are noisier than the total reward curve, especially for sparse grounding sub-rewards, so we keep them in the appendix. They are nevertheless useful as a sanity check: VISD's improvement is not caused by a single formatting artifact, but is accompanied by stronger answer and grounding-related signals over training.

D.3 Teacher Stability Diagnostics

Figure 6 provides additional optimization diagnostics for the teacher update ablation. The Sync-10 teacher synchronizes parameters with the student every 10 steps and keeps them frozen between

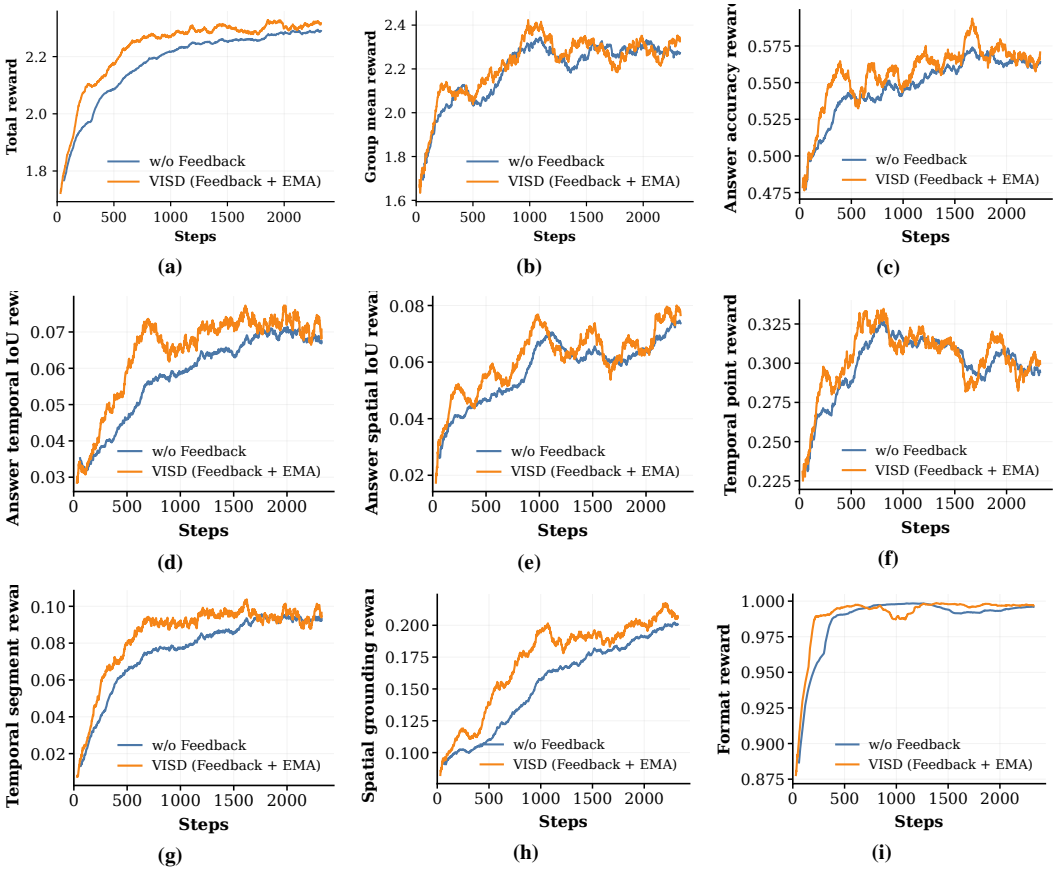


Figure 5: *Component-level reward curves for feedback ablation.* Each panel compares VISD with and without judge feedback for one reward component: **(a)** total reward, **(b)** group mean reward, **(c)** answer accuracy, **(d)** temporal IoU, **(e)** spatial IoU, **(f)** temporal point, **(g)** temporal segment, **(h)** spatial grounding, and **(i)** format.

synchronizations, whereas the current-policy teacher uses the instantaneous student parameters. The EMA teacher smooths high-frequency parameter changes and gives a more stable teacher signal. Consistent with Table 5, the Sync-10 variant exhibits a substantially larger gradient norm and a sharper disturbance as the teacher weight approaches zero, while EMA preserves the best final reward.

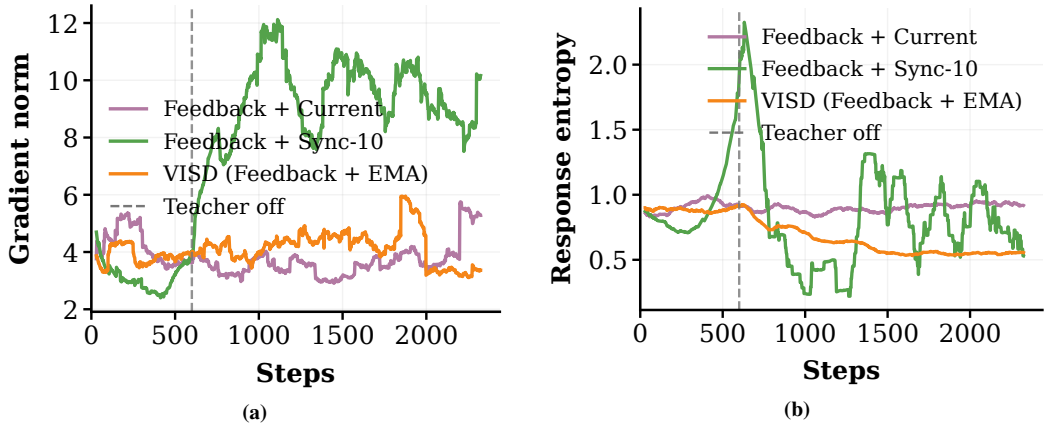


Figure 6: *Teacher update stability diagnostics.* We show **(a)** gradient norm and **(b)** response entropy for different teacher parameterizations.

D.4 Training Details for Ablation Settings

The ablation curves are plotted by training step. For the feedback comparison, the no-feedback setting uses the same reinforcement-learning and teacher-replay pipeline but removes the judge-generated process feedback. Teacher-weight annotations are shown only in the teacher-update comparison, where all compared settings share the same teacher schedule.

For the teacher-update ablation, all compared variants use judge feedback. The current-policy teacher directly uses the latest student parameters. The Sync-10 teacher synchronizes teacher parameters with the student every 10 training steps and keeps them frozen between synchronizations. The EMA teacher updates teacher parameters as an exponential moving average of the student parameters. All reported final metrics in Table 5 are averaged over the final 100 training steps.

E Mechanistic Analysis and Qualitative Diagnostics

E.1 How Judge Feedback Affects Credit Assignment

The ablation in Section 4.3 shows that judge feedback improves training, but it does not by itself explain *how* the feedback enters the policy update. VISD uses the judge as a trajectory-level diagnostic channel. As illustrated in Appendix E.2, different rollouts of the same question receive different judge evaluations: the judge distinguishes failed, partially correct, and correct trajectories, and points to whether the error comes from the answer, temporal coverage, spatial grounding, or the consistency between reasoning and answer. This suggests that the feedback is not merely an additional prompt field, but a rollout-dependent description of *how* the attempt succeeds or fails.

The teacher then converts this diagnosis into token-level magnitude modulation. An answer-only teacher observes the verified answer-side information, while the feedback-conditioned teacher additionally observes the judge diagnosis. Thus, the teacher replay distribution changes from $q(\cdot \mid x, a^\star, y_{<t})$ to $q(\cdot \mid x, a^\star, f, y_{<t})$, where f denotes judge feedback. The rollout-level advantage still determines whether the whole trajectory should be reinforced or suppressed; the teacher-student ratio only modulates the relative update magnitude of individual tokens. This direction–magnitude separation prevents the judge from overwriting the reward direction, while still allowing feedback to redistribute learning pressure within the trajectory.

We provide a formal decomposition and token-level visualizations in Appendix E.4. The qualitative comparison shows that feedback-conditioned replay does not uniformly change all tokens relative to answer-only replay. Instead, the additional changes are sparse and tend to appear around tokens tied to diagnosed process errors, such as spurious actions, incomplete temporal coverage, or answer-critical object descriptions. These visualizations are intended as mechanism evidence rather than standalone performance evidence: they illustrate how the extra information in judge feedback can influence credit assignment over process tokens on top of answer-only supervision.

E.2 Trajectory-Dependent Judge Feedback

Figure 7 shows the judge feedback for different rollouts of the same video question. The judge assigns different evaluations to failed, partially correct, and correct trajectories, and provides diagnosis about the corresponding failure modes. This example supports the role of the judge as a rollout-dependent diagnostic channel: the feedback describes ***not only whether a rollout succeeds, but also how its reasoning, answer, and temporal evidence agree or conflict***. VISD uses this signal as privileged context for teacher replay, while the rollout-level reward advantage remains responsible for the sign of the policy update.

E.3 Top-K Local Support Ratio

This section clarifies the top-K ratio used by VISD. Let $q_t(v)$ denote the feedback-conditioned teacher distribution at token position t, and let $\pi_t(v)$ denote the student distribution under the ordinary rollout context:

$$q_t(v) = q(v \mid x, a^\star, e, f, y_{<t}), \qquad \pi_t(v) = \pi_\theta(v \mid x, y_{<t}). \tag{19}$$

VISD constructs a teacher-centered local support by augmenting the teacher top-K candidates with the realized rollout token:

$$\mathcal{U}_t = \mathrm{TopK}_{q_t}(K) \cup \{y_t\}. \tag{20}$$

The augmentation by y_t is not an additional sampled-token objective; it only ensures that the on-policy token receiving the policy-gradient update is always covered by the local support. Within $\mathcal{U}_t$, teacher and student probabilities are renormalized as

$$\widetilde{q}_t(v) = \frac{q_t(v)}{\sum_{u\in\mathcal{U}_t} q_t(u)}, \qquad \widetilde{\pi}_t(v) = \frac{\pi_t(v)}{\sum_{u\in\mathcal{U}_t} \pi_t(u)}, \qquad v \in \mathcal{U}_t. \tag{21}$$

VISD then computes the teacher-student evidence ratio on the realized token:

$$\Delta_t^K = \mathrm{sg}\left[\log \widetilde{q}_t(y_t) - \log \widetilde{\pi}_t(y_t)\right]. \tag{22}$$

Unlike a top-K KL loss, Eq. 22 does not introduce gradients over all tokens in $\mathcal{U}_t$. It only provides a stop-gradient scalar used to reweight the policy-gradient term of the generated token.

Local-support calibration. Let

$$Q_t(\mathcal{U}_t) = \sum_{u\in\mathcal{U}_t} q_t(u), \qquad P_t(\mathcal{U}_t) = \sum_{u\in\mathcal{U}_t} \pi_t(u). \tag{23}$$

Then Eq. 22 admits the exact decomposition

$$\Delta_t^K = \underbrace{\log q_t(y_t) - \log \pi_t(y_t)}_{\text{token evidence ratio}} + \underbrace{\log \frac{P_t(\mathcal{U}_t)}{Q_t(\mathcal{U}_t)}}_{\text{local-support calibration}}. \tag{24}$$

This identity shows that the top-K ratio preserves the original teacher-student evidence ratio on the realized token, while adding a calibration term that accounts for how much probability mass the teacher and student assign to the teacher-supported local region.

Direction preservation and bounded redistribution. VISD converts the local-support ratio into a positive token weight:

$$m_t = \mathrm{clip}\left(\exp\left(\mathrm{sign}(A)\Delta_t^K\right), 1-\epsilon_w, 1+\epsilon_w\right). \tag{25}$$

The token-level advantage is

$$\hat{A}_t = A\left[(1-\lambda) + \lambda m_t\right]. \tag{26}$$

Because $m_t > 0$, the reweighting cannot flip the rollout-level update direction:

$$\mathrm{sign}(\hat{A}_t) = \mathrm{sign}(A), \qquad A \neq 0. \tag{27}$$

Moreover, since $m_t \in [1-\epsilon_w, 1+\epsilon_w]$,

$$\frac{\hat{A}_t}{A} \in [1-\lambda\epsilon_w,\ 1+\lambda\epsilon_w]. \tag{28}$$

Thus, the teacher-conditioned signal only redistributes token-level update magnitudes within a bounded range, and the update naturally reduces to the standard rollout advantage as λ anneals to zero.

E.4 Feedback-Conditioned Credit as an Information Layer

This section formalizes the difference between answer-only teacher replay and feedback-conditioned teacher replay. Let $a^\star$ denote verified answer-side information and let f denote judge feedback for a sampled student rollout. At token position t, we compare the two teacher contexts on a common local support and let $\widetilde{q}_t^{\,a}$ and $\widetilde{q}_t^{\,a,f}$ denote the renormalized teacher distributions under answer-only and feedback-conditioned contexts, respectively. The answer-only teacher-student log-ratio is

$$\Delta_t^{\mathrm{ans}} = \log \widetilde{q}_t^{\,a}(y_t) - \log \widetilde{\pi}_t(y_t), \tag{29}$$

whereas the feedback-conditioned log-ratio is

$$\Delta_t^{\mathrm{fb}} = \log \widetilde{q}_t^{a,f}(y_t) - \log \widetilde{\pi}_t(y_t). \tag{30}$$

Their difference isolates the contribution of judge feedback:

$$\delta_t^{\mathrm{judge}} = \Delta_t^{\mathrm{fb}} - \Delta_t^{\mathrm{ans}} = \log \widetilde{q}_t^{a,f}(y_t) - \log \widetilde{q}_t^{a}(y_t). \tag{31}$$

Equation 31 makes explicit that judge feedback acts as an additional information layer on top of answer-only supervision. It does not introduce a separate teacher-generation target; instead, it changes how the teacher scores the same original student completion.

VISD applies this signal through a positive token-wise reweighting factor. As shown in Appendix E.3, the judge-conditioned teacher can change the relative magnitude assigned to different tokens, but it cannot flip the rollout-level reinforcement direction. This property is important for our interpretation: feedback contributes diagnostic information for *which process-relevant tokens* should receive different update magnitudes, while the reward advantage still decides whether the trajectory is promoted or suppressed. Figure 8 should therefore be read as a controlled comparison: the student rollout, reward, and displayed token sequence are fixed, and only the teacher-side context changes from answer-only information to feedback-conditioned diagnostic context. The visual differences between **(a)** and **(b)** instantiate $\delta_t^{\mathrm{judge}}$ in Equation 31: they show where the added feedback context changes the teacher-side token evidence used for magnitude reweighting.

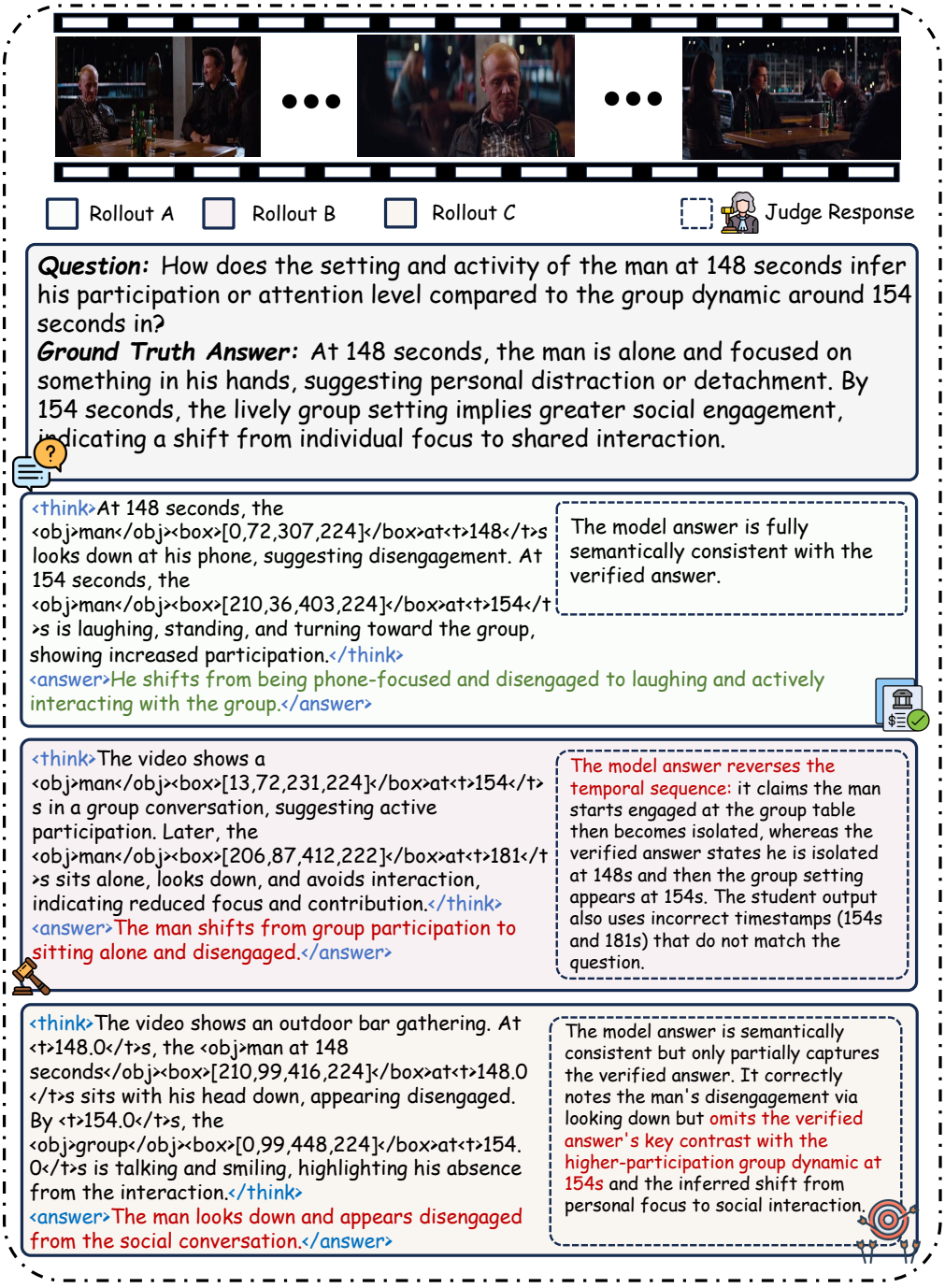
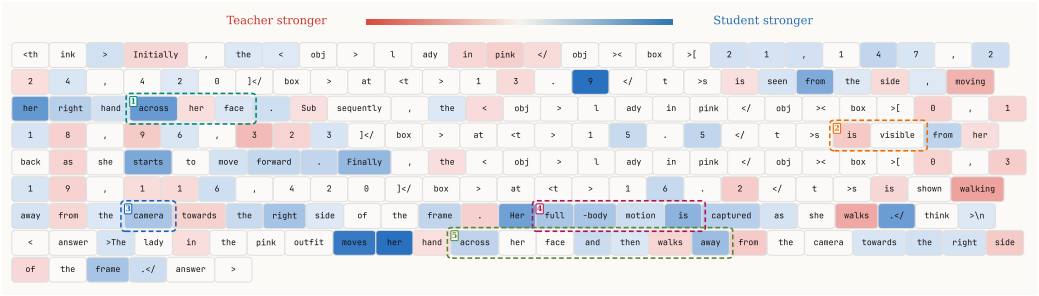
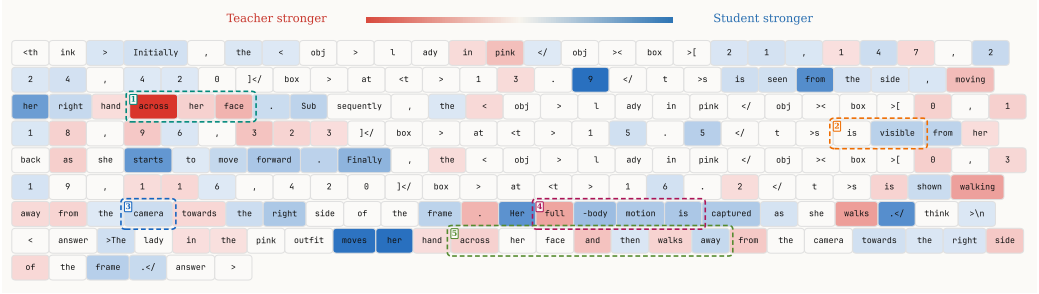


Figure 7: *Trajectory-dependent judge feedback.* For the same video question, the judge provides different evaluations and diagnoses for different student rollouts. The feedback is used as privileged information for teacher replay rather than as a replacement for the reinforcement reward.



(a) Answer-only teacher replay



(b) Feedback-conditioned teacher replay

Figure 8: *Answer-only versus feedback-conditioned token credit.* **(a)** and **(b)** visualize the same fixed student rollout; only the teacher context changes. Warmer and cooler token backgrounds indicate positive or negative teacher-student token evidence used to modulate policy-gradient magnitude. In the answer-only replay, the teacher is conditioned on verified answer-side information, so the token-credit pattern mainly reflects whether the generated trajectory is compatible with the final answer. In the feedback-conditioned replay, the teacher additionally receives the judge diagnosis of the rollout, which specifies process-level issues in the reasoning trajectory. The important comparison is therefore not whether every token changes, but whether feedback changes token evidence at reasoning, grounding, and answer-relevant positions. The numbered outlines mark representative process-relevant positions affected by feedback: **1** marks the early evidence `across her face`, which changes from student-favored to teacher-favored after feedback; **2** marks the intermediate visibility claim `is visible`; **3** marks later camera evidence; **4** marks `full-body motion is`; and **5** marks the answer phrase `across her face and then walks away`. The most visible changes occur around the action evidence in **1**, the motion evidence in **4**, and the answer-level action phrase in **5**. Most other tokens remain close to the answer-only pattern, indicating that feedback selectively changes process-relevant credit rather than uniformly shifting the entire trajectory. This illustrates the intended mechanism of VISD: the rollout-level advantage still determines the update direction, whereas judge feedback adds an information layer that redistributes token-level update magnitudes toward process-relevant parts of the same trajectory.

F Qualitative Case Studies

To further illustrate the effectiveness of VISD, we provide qualitative case studies covering diverse video understanding scenarios. As shown in Figures 9–17, VISD grounds its reasoning in precise visual evidence, localizing relevant objects, actors, actions, and temporal intervals before producing the final answer. Compared with existing video reasoning models, which often rely on incomplete observations or incorrect temporal and spatial evidence, VISD demonstrates stronger alignment between visual grounding, intermediate reasoning, and response generation. These examples show that VISD not only answers correctly, but also provides more faithful and interpretable reasoning processes across both spatial and temporal video understanding tasks.

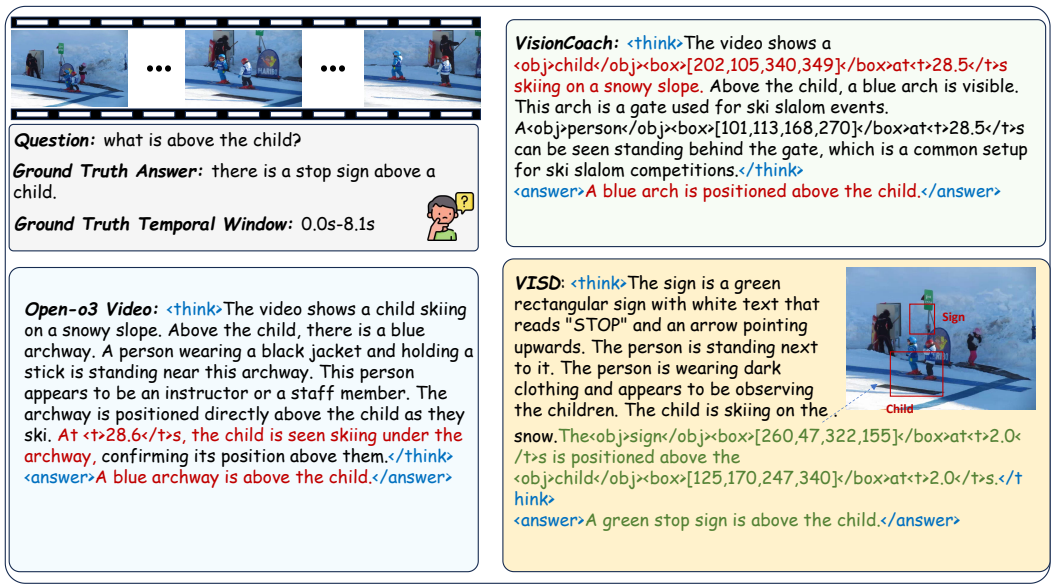


Figure 9: *Visualization*. For spatial relation reasoning, VISD accurately localizes the queried child and identifies the object positioned above him, providing precise visual evidence while avoiding confusion with nearby objects. In contrast, related video reasoning models either give incorrect answers or rely on incomplete spatial grounding.

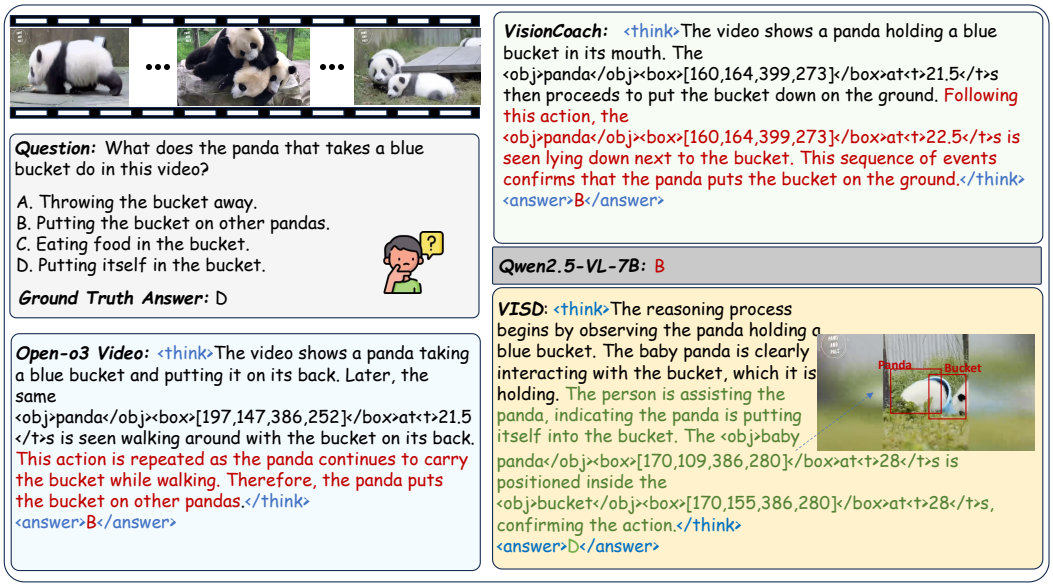


Figure 10: *Visualization*. For temporal action reasoning, VISD grounds the panda and bucket across relevant frames and correctly infers that the panda is putting itself in the bucket. Competing models miss this action transition and produce incorrect answers.

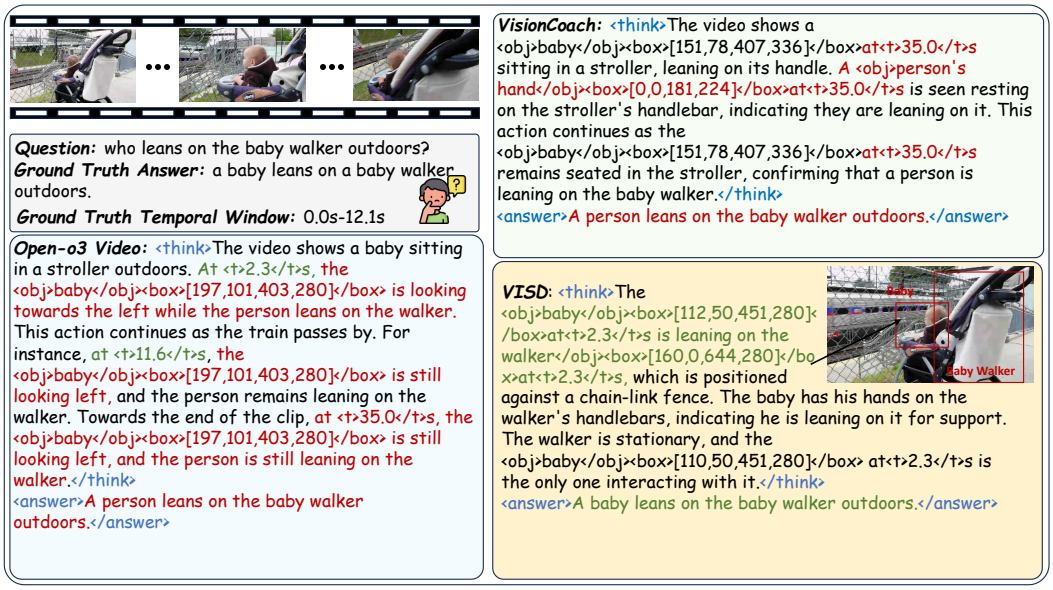


Figure 11: *Visualization*. For outdoor spatial interaction reasoning, VISD accurately localizes both the baby and the baby walker, identifies the baby as the one leaning on the walker, and provides precise visual evidence grounded in the relevant temporal window. Other models instead confuse the person with the baby and produce an incorrect answer.

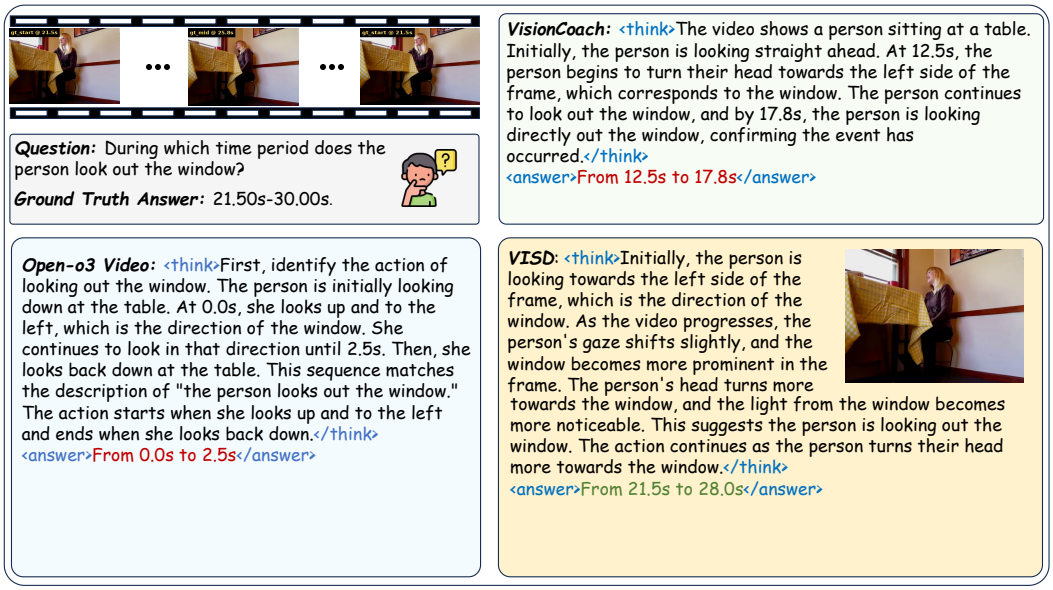


Figure 12: *Visualization*. For temporal localization reasoning, VISD accurately grounds the person's gaze direction across frames and identifies the correct time window when the person looks out of the window. Other models focus on earlier or incomplete head movements and produce incorrect intervals, whereas VISD captures the sustained window-looking action and matches the ground-truth period.

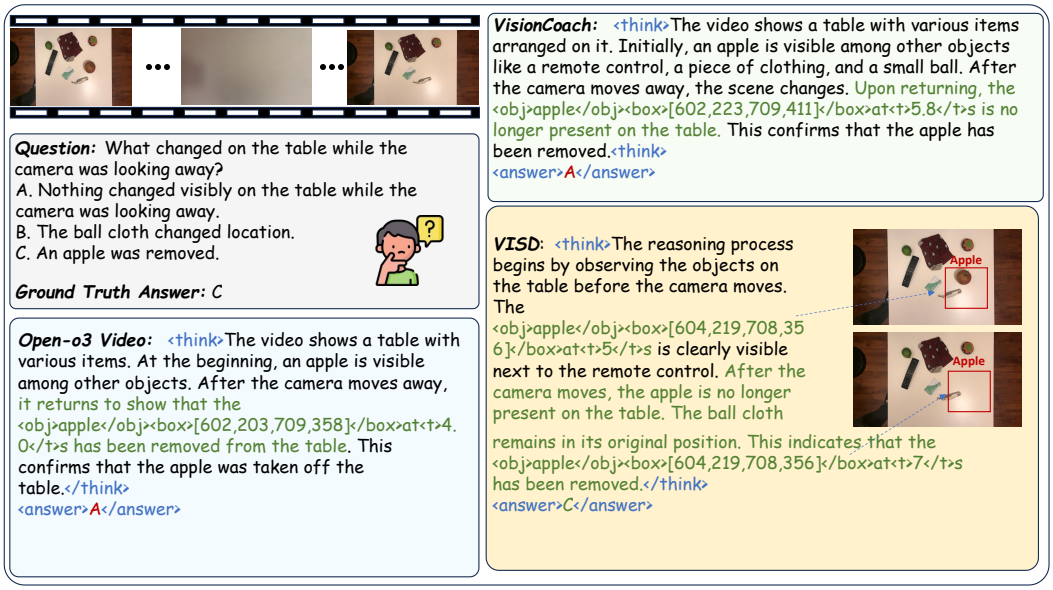


Figure 13: *Visualization*. In object disappearance reasoning, VISD explicitly grounds the apple before and after the camera movement, correctly identifying that the apple has been removed from the table. Although related models partially recognize the same visual change in their reasoning process, they still produce an inconsistent final answer, highlighting VISD's stronger alignment between visual grounding, reasoning, and response generation.

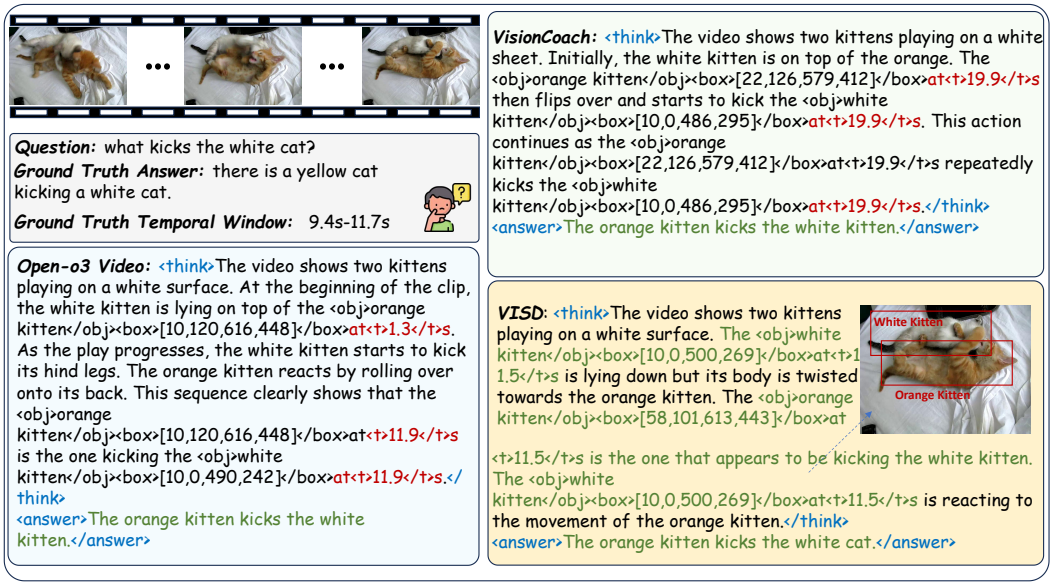


Figure 14: *Visualization*. VISD not only answers correctly but also localizes the action within the correct temporal window, identifying the moment when the orange kitten kicks the white kitten. Although related video reasoning models produce the correct final answer, they rely on inaccurate temporal evidence.

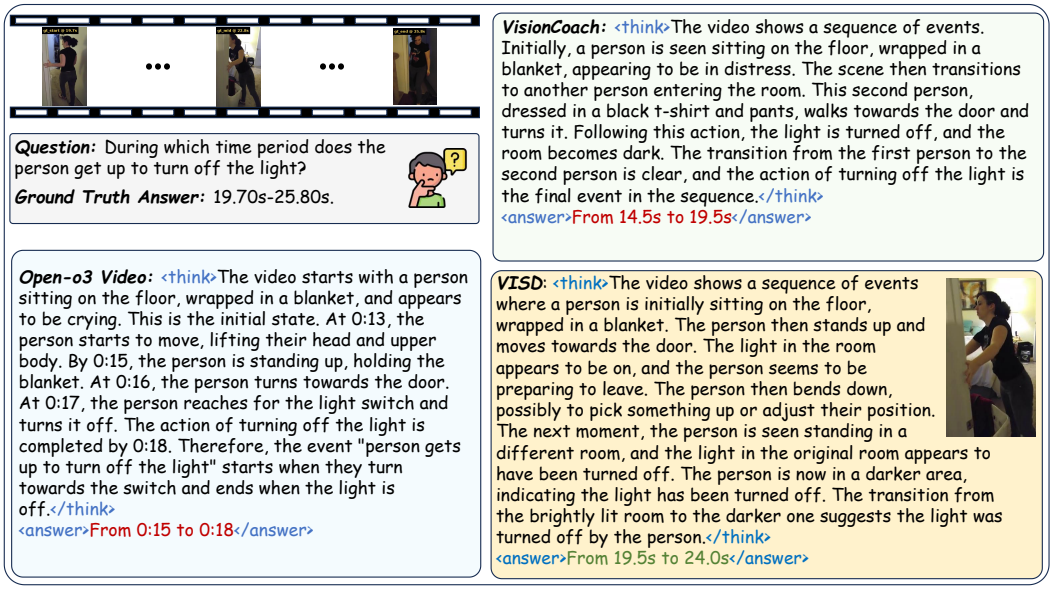


Figure 15: *Visualization*. For temporal action localization reasoning, VISD accurately grounds the key transition from the lit room to the dark room and identifies the correct time interval when the person gets up to turn off the light. In contrast, related video reasoning models focus on earlier incomplete movements, such as sitting up or approaching the switch, resulting in incorrect temporal predictions.

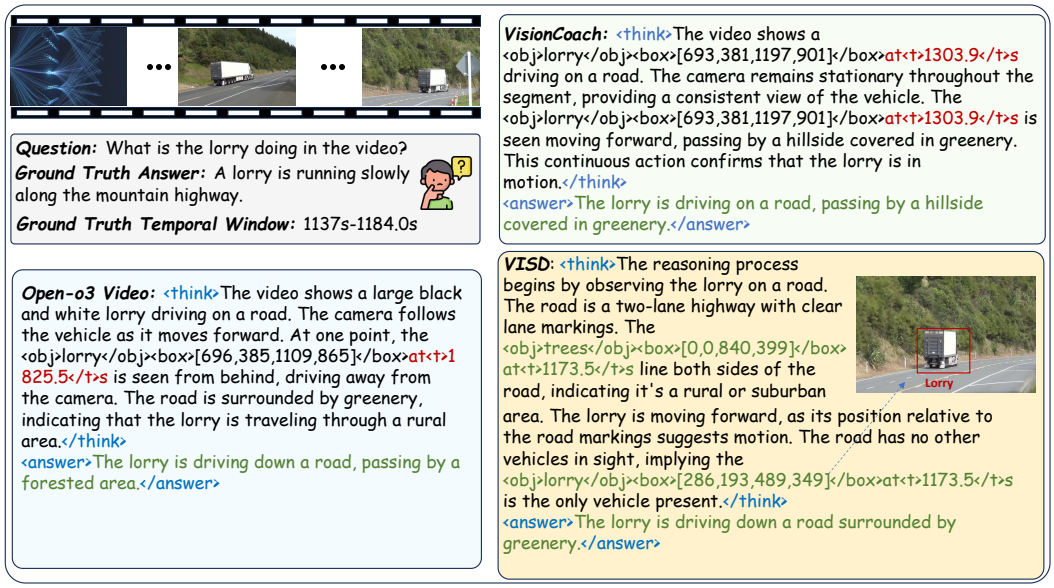


Figure 16: *Visualization*. VISD correctly answers the question and localizes the lorry within the ground-truth temporal window. Although other models also produce the correct answer, they ground their reasoning in an incorrect time interval, showing less precise temporal localization.

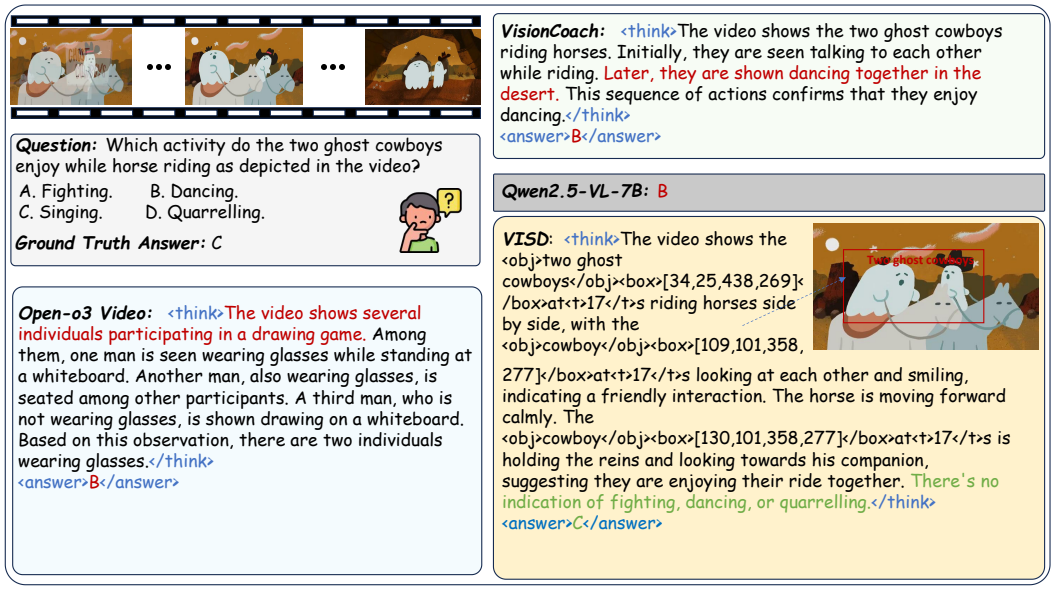


Figure 17: *Visualization*. In fine-grained activity reasoning, VISD accurately grounds the two ghost cowboys during horse riding and identifies their smiling, friendly interaction, correctly inferring that they enjoy singing rather than dancing, fighting, or quarrelling. In contrast, competing models are distracted by later or irrelevant visual cues and produce incorrect answers.

G Judge Prompt Template

We provide the core prompt template used by the video-aware judge to generate structured privileged feedback. The template contains a fixed system instruction and the user prompt assembled from the task type, reference answer, optional grounding evidence, and the student rollout. Placeholders such as `{question}` and `{student_output}` denote fields filled at training time.

System Prompt: Video-Aware Judge

```
## Role
You are a strict video and image QA judge.

## Mission
You are judging a student model's response. Your feedback will be given to a teacher model, which
     will use it to better guide the student model. Judge whether the student's final answer is
     correct, whether the student's reasoning broadly supports that final answer, and what the
     single most likely high-level source of error is when the response is not fully correct.

## Input Boundaries
- Use only the information provided in the prompt.
- You do not see the original video frames directly.
- Do not pretend to verify frame-by-frame visual alignment that is not supported by the provided
     text or hidden grounding evidence.
- For structured answers, compare the provided fields against the reference answer and grounding
     evidence when available.
- Treat semantically equivalent natural-language answers as correct even if the wording differs.

## Output Format
- Return strict JSON only with key: feedback.
- Do not return markdown, prose outside JSON, or extra keys.

## Feedback Requirements
- The feedback should be concise, factual, correction-oriented, and usually two to four sentences.
- Mention only the issues that actually appear; do not give an exhaustive recap.
- If the response is fully correct and no clear issue is observed, provide a brief positive
     feedback sentence stating that the final answer is correct and the reasoning broadly supports
      it.

## Evaluation Focus
1. Answer diagnosis.
Say whether the student final answer is fully correct, partly correct, or wrong. Briefly name the
     problematic part. For natural-language answers, focus on semantic meaning rather than exact
     wording. For structured answers, say which part is wrong or missing.
2. Reasoning-versus-answer consistency.
Judge whether the student's reasoning broadly supports the final answer. If the reasoning points
     to one event, object, text clue, time range, or spatial reference but the final answer states
      another, say so. If the reasoning is too weak, too broad, or too incomplete to justify the
     final answer, say that clearly.
3. High-level error cause.
When the response is not fully correct, pick the single most likely high-level cause and mention
     it briefly. Prefer one of these categories: the reasoning focused on the wrong event, time
     span, or object; the reasoning was too broad or lacked enough evidence to support such a
     specific answer; the reasoning was mostly on the right track but the final answer overreached,
      drifted, or stated the wrong thing; or the output was incomplete and never produced a real
     final answer. Do not invent fine-grained step-by-step visual mistakes when they are not well
     supported by the text.

## Task
Compare the verified answer against the student's final answer and reasoning text, then return the
      feedback JSON.

If keyframe object evidence is provided:
Hidden grounding evidence may be provided in the user prompt.

If keyframe object evidence is not provided:
No hidden grounding evidence is required for this sample.

If the student output does not contain a final <answer> or </answer> tag:
The student output does not contain a final <answer> or </answer> tag. You must still analyze the
     student's reasoning text and provide corrective feedback. In the feedback, explicitly point
     out that the final answer was not produced and that the response may have been truncated
     because the thinking section was too long. Treat the missing final answer as an incomplete
     response.
```

User Prompt: Runtime Assembly

```
Please judge the following sample.

## Task And Reference Context
- Task type: {task}
- Question: {question}
- Standard answer: {standard_answer}
- Keyframe object evidence: {keyframe_object_evidence_json_or_null}
- Keyframe object evidence is hidden grounding reference. Use it only to judge whether the student'
      s reasoning points to the right time, object, and spatial region.

## Student Submission
- Student final answer: {model_answer_or_null_no_answer_tag_found}
- Student full output: {student_output}

## Required Output
{
  "feedback": "{concise diagnostic feedback}"
}
```

H Broader Impacts

VISD aims to improve video understanding systems in tasks that require more reliable reasoning over temporal and spatial evidence. Such capability can be beneficial for applications including video retrieval, assistive video understanding, educational tools, and human-in-the-loop video analysis, where better grounding may improve interpretability and reduce obvious reasoning errors.

At the same time, stronger video reasoning models may also introduce risks. They could be used in surveillance-related settings, content analysis pipelines, or other scenarios where incorrect or overconfident predictions may affect downstream decisions. Like other large multimodal models, they may also inherit biases from pretraining data or produce misleading grounded explanations when the underlying prediction is wrong. In this work, we evaluate VISD only on public research benchmarks and release code for research purposes, and we view responsible deployment, dataset governance, and more robust evaluation under open-world settings as important future directions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes].

   Justification: The contributions and scope of the paper are well summarized in the abstract and introduction.

   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 discuss the limitations of our work in Appendix B.

   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: The paper does not present theorem-style theoretical results.

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: The method, algorithm, hyperparameters, prompts, and evaluation details are provided in Appendices C–G, and the supplementary code package includes the training and evaluation scripts used to reproduce the main results.

   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: [Yes].

Justification: We use publicly available datasets and follow their original preprocessing and evaluation protocols unless otherwise specified. Our training and evaluation code, together with environment setup, data preparation, and reproduction instructions, is publicly available at `https://github.com/Koreyoshi01/VISD`.

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: Training settings, hyperparameters, and evaluation protocols are specified in Appendices C.2 and C.3.

   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: Due to the limited computational resources, we conducted each experiment only once.

   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: The paper reports the main training and evaluation hardware and major optimization settings in Appendices C.2 and C.3.

   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: We fully adhere to the NeurIPS Code of Ethics.

   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: Appendix H discusses both potential benefits and potential risks of stronger video understanding systems.

    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: The paper does not introduce a high-risk scraped dataset or unrestricted high-risk model release.

    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: Existing datasets, benchmarks, and model families are cited and used under their research settings.

    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: The supplementary code package includes documentation and usage instructions for the training and evaluation code released with the submission.

    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: [N/A].

    Justification: The work does not involve crowdsourcing or human-subject experiments.

    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: The work does not involve crowdsourcing or human-subject experiments.

    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: The external judge LLM is a core training component, and VSTAR uses an LLM evaluator following its benchmark protocol; these are described in Appendices C.3 and G.

    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.