Watching Itself Watch: Self-Auditing Visual Reliance for Video Reasoning

Rohan Dugad Mi Luo Kristen Grauman
The University of Texas at Austin

Abstract

Video reasoning hinges on a small number of visually decisive tokens embedded in otherwise largely language-driven chains of thought. Yet prevailing post-training methods such as GRPO overlook this structure, assigning coarse trajectory-level rewards while requiring multiple rollouts and often costly external judges. We introduce **Vi**sually-Aware **O**n-Policy **S**elf-Distillation (VIOS), a post-training framework that identifies where a model under-utilizes visual evidence and concentrates learning at those positions. Our key insight is that a model can *audit its own visual reliance*: the divergence between its next-token distributions with and without the video reveals how strongly each prediction depends on the visual input. Measuring this reliance for both a student and an answer-privileged teacher exposes a per-token *reliance gap*, identifying positions where the teacher exploits visual evidence that the student overlooks. VIOS translates this signal into dense, targeted supervision through three complementary components: an exponential moving average (EMA) teacher that co-evolves with the student, a contrastive objective that encourages predictions to depend on the video, and a reliance-gap gate that focuses learning on visually under-utilized tokens. Crucially, VIOS requires only a single on-policy rollout from a single model, without external rewards, judges, or larger teachers. Across 12 benchmarks spanning general, temporal, spatial, knowledge-intensive, and long-video understanding, VIOS consistently improves three model backbones and outperforms contemporary video-reasoning systems, while training in over $9\times$ fewer steps, $60\times$ fewer FLOPs, and $111\times$ fewer generated tokens.[1]

1 Introduction

Consider a video in which deciding whether one event caused another rests on a brief transition across two frames. The narration around that transition does not describe it and the prior context does not imply it. Video reasoning in multimodal large language models (MLLMs) is characterized by exactly this kind of dependency [7, 38, 27], where a correct continuation hinges on specific moments of visual evidence that no textual fluency can substitute for. A model must track entities through occlusion, relate causes to outcomes that are mere seconds apart, and revise its working hypothesis as new frames arrive [35, 21, 41]. What distinguishes this regime from language-only reasoning is not the length of the chain of thought, but the fact that particular positions in that chain are answerable only by careful attention to the visual input.

Despite the importance of visually decisive moments, existing video post-training methods provide coarse supervision at substantial computational cost. Supervised fine-tuning is off-policy, trains the student on prefixes it never produces itself, and tends to memorize surface patterns of the demonstrations [1, 48, 4, 51]. Group Relative Policy Optimization (GRPO) [34, 11] closes the on-policy gap by sampling multiple rollouts per prompt and rewarding those that score above the group mean while penalizing those below it. This estimator typically requires eight or more rollouts per prompt [34, 11, 46], wastes gradient on groups that collapse to identical rewards [46, 30, 20], and

[1] Project page: https://vision.cs.utexas.edu/projects/VIOS/

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

applies a single scalar reward across the trajectory even though only a small minority of tokens determine its outcome [39, 37, 16]. Recent video methods compound this cost by relying on substantially larger judge models to assign rewards online [23, 27, 26]. Ultimately, these methods remain costly while failing to identify where the model under-utilizes visual evidence during reasoning.

Our central observation is that this missing structure can be recovered from the model itself. We introduce **Vi**sually-Aware **O**n-Policy **S**elf-Distillation (**VIOS**), a novel post-training procedure that recovers dense, per-token, vision-aware supervision from a single rollout and a single model.

Our framework consists of a student and a teacher, both initialized from the same base model. Both receive the video and question, while the teacher additionally sees the ground-truth answer. The student generates an on-policy rollout, and the teacher provides next-token distributions for the student to match at each position. The teacher is updated through an exponential moving average (EMA) of the student, yielding a co-evolving supervision target. However, matching the teacher alone does not ensure that the student actually uses the video. To address this, we measure the student's *visual reliance* at each token by comparing its next-token distributions with and without the video in context. A larger difference indicates that the prediction is more sensitive to visual input. We introduce a contrastive objective that encourages this sensitivity, pushing the student to rely on the video rather than textual cues alone. We then measure the teacher's visual reliance in the same way and compute the *vision-reliance gap*: the difference between the teacher's and student's reliance at each token. A large positive gap identifies positions where the teacher uses visual information more strongly than the student. By using this gap to weight both learning objectives, we focus training on tokens where the visual input is under-utilized. In this way, our method enables the model to "watch itself watch," to identify and improve where its predictions depend on what it sees.

Our main contribution is the novel VIOS post-training paradigm, an efficient framework for video reasoning that combines: a co-evolving EMA teacher that provides an ever-improving privileged target; a contrastive objective that ensures the student policy is visually aware; and a vision reliance gap gate that concentrates learning on the positions where the visual input is under-utilized.

We evaluate VIOS across 12 popular benchmarks spanning general video understanding, temporal reasoning, spatial reasoning, knowledge, and long-video understanding. VIOS consistently outperforms its base models and contemporary video-reasoning systems, while also reaching this performance in over $9\times$ fewer training steps, $60\times$ fewer FLOPs, and $111\times$ fewer generated tokens. Our results suggest that in video reasoning, the path forward is not scaling external supervision or rollouts but teaching models to recognize, on their own, where their reasoning should depend on what they see.

2 Related Work

Video Reasoning in Multimodal Large Language Models. RL with verifiable rewards has driven recent progress in LLM reasoning [11, 34, 18], where Group Relative Policy Optimization (GRPO) and related objectives elicit strong multi-step reasoning in math and code. Recent efforts extend these ideas to video reasoning in MLLMs [7, 23, 44, 38, 27, 26, 43]. Video-R1 [7], VideoChat-R1 [23], and Video-R2 [27] augment GRPO with spatio-temporal rewards and objectives, while VideoRFT [38] and Video-VER [26] design rewards that ground reasoning in visual evidence; VideoChat-R1.5 [44] instead interleaves iterative perception with GRPO. However, these works are limited by GRPO's sample-inefficiency and sparse reward signal, and some works [23, 27, 26] also rely on substantially larger or more capable models as judges during GRPO training. Post-training approaches that move beyond GRPO remain underexplored in this setting. VIOS departs from this line of work by obtaining dense, per-token supervision recovered from a single rollout of a single model, which drastically improves sample-efficiency, enabling less computation and faster training time.

Visual-Input Sensitivity in Multimodal Models. Prior work has explored the sensitivity of MLLM predictions to visual perturbations as a means of guiding decoding and preference optimization. Visual Contrastive Decoding [19] contrasts next-token logits from an original image and its diffusion-noised counterpart to adjust decoding without training. ConVis [31] similarly performs training-free contrastive decoding, but constructs the contrasting image using an external text-to-image model. Token Preference Optimization [10] instead uses image-corruption sensitivity to weight token-level preferences within a DPO-style objective. However, these methods use visual sensitivity to adjust decoding or weight off-policy preference optimization, rather than directly addressing visual under-utilization during on-policy reasoning. VIOS advances this direction by integrating a visual-awareness

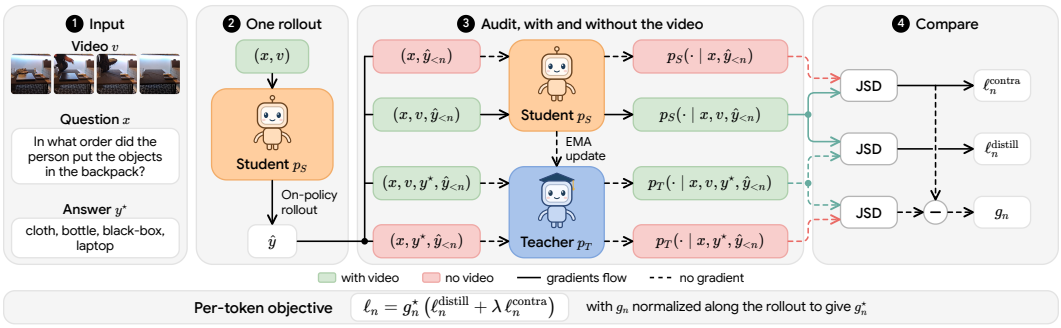


Figure 1: Overview of VIOS. A single on-policy rollout provides per-token, vision-aware supervision through three JSD comparisons between the student p_S and EMA teacher p_T: distillation toward the teacher, contrastive separation from the student's no-vision counterpart, and a reliance gap g_n that emphasizes tokens where the teacher uses vision and the student does not.

contrastive objective with privileged on-policy self-distillation, using the relative teacher-student reliance gap to concentrate dense supervision where the student under-utilizes video evidence.

On-policy Self-Distillation for Reasoning. Knowledge distillation has long been used to transfer a teacher's behavior to a student via soft targets [12, 17, 33], but off-policy distillation from static teacher trajectories suffers from distribution shift since the student is never exposed to its own mistakes. On-policy distillation [1, 25, 42] mitigates this by sampling trajectories from the student and grading them with dense, token-level teacher supervision, combining the on-policy relevance of RL with a non-sparse learning signal, though it typically requires a separate and more capable teacher model. On-policy self-distillation [48] (OPSD) takes a step toward using a single model as both teacher and student: the student samples on-policy rollouts, while the teacher, conditioned additionally on a privileged ground-truth chain of thought, supplies per-token targets using a fixed-direction KL objective. However, OPSD requires ground-truth reasoning traces, freezes the teacher at the base model, and remains limited to text-only reasoning. VIOS addresses these limitations with a co-evolving EMA teacher conditioned only on the final answer and trained via a symmetric, bounded JSD objective, a visual-awareness contrastive objective that encourages video-dependent predictions, and a per-token vision-reliance gap gate that focuses supervision where vision is under-utilized. Together, these remove the need for ground-truth rationales, lift the rationalization ceiling of a stationary teacher, and explicitly target visual under-utilization in multimodal reasoning.

3 Method

A model can audit its own visual reliance: comparing its next-token distribution with and without the vision input at any position measures how much that prediction depends on what was seen. We introduce **VIOS** (**Vi**sually-Aware **O**n-Policy **S**elf-Distillation), turning this into a post-training procedure. Auditing the evolving student and a privileged teacher exposes a per-token *reliance gap* that reveals where the teacher exploits visual evidence and the student does not. From this we recover dense, vision-aware supervision out of a single on-policy rollout of a single model, with no external reward, judge, or teacher. We instantiate the idea through three components: a self-distillation term against a co-evolving EMA teacher, a contrastive term that forces predictions to depend on v, and a reliance-gap gate that focuses both on visually-sensitive tokens.

Notation. We consider a video question answering (VQA) dataset $\mathcal{D} = \{(x_i, v_i, y_i^\star)\}_{i=1}^N$, where each example consists of a text query x, a video v, and a ground-truth final answer $y^\star$. All policies are instantiated from a single vision-language model with parameters θ.

- *Student* $p_S(\cdot \mid x, v)$ and *teacher* $p_T(\cdot \mid x, v, y^\star)$ share the same architecture. The teacher additionally conditions on the privileged ground-truth answer $y^\star$ and is parameterized by an EMA copy $\bar{\theta}$ of the student.
- $\hat{y} = (\hat{y}_1, \ldots, \hat{y}_{|\hat{y}|}) \sim p_S(\cdot \mid x, v)$ denotes an on-policy student rollout, and $p(\cdot \mid \ldots, \hat{y}_{<n})$ the next-token distribution at position n along it.
- $\mathrm{JSD}(p\|q) = \frac{1}{2}\mathrm{KL}(p\|m) + \frac{1}{2}\mathrm{KL}(q\|m)$ with $m = \frac{1}{2}(p+q)$, bounded in $[0, \log 2]$.
- $\mathrm{sg}[\cdot]$ denotes stop-gradient.

3.1 Self-Distillation Term

The student samples a rollout $\hat{y} \sim p_S(\cdot \mid x, v)$, and the teacher scores every prefix. The teacher sees the same inputs *plus* the ground-truth answer $y^\star$, so its next-token distribution reflects what the trajectory should look like given the correct answer. Aligning the student to the teacher position-by-position along its own rollout converts privileged knowledge of $y^\star$ into a dense, on-policy signal.

EMA teacher. Rather than freezing the teacher at initialization, we set $\bar{\theta}$ to be an exponential moving average of the student, updated after each optimizer step,

$$\bar{\theta} \leftarrow \alpha\, \bar{\theta} + (1 - \alpha)\, \theta, \qquad \alpha \in (0, 1). \tag{1}$$

A fixed teacher caps the attainable performance at the initial model's rationalization ability. An EMA teacher instead co-evolves with the student, so privileged-context supervision improves as the student's underlying capability improves. The slow averaging preserves the stability of a non-updating teacher (gradients still only flow through the student) while lifting the ceiling imposed by a stationary target.

JSD objective. We measure teacher-student disagreement with the Jensen-Shannon divergence. JSD is symmetric, bounded, and sits between forward KL (mode-covering) and reverse KL (mode-seeking). In video reasoning, many phrasings describe the same content equally well; mode-covering all of them leaves the policy weak on any one, while mode-seeking collapses onto a single rationalized continuation. JSD's balance keeps the student committed to specific phrasings while permitting plausible alternatives. The distillation loss is

$$\mathcal{L}_{\text{distill}}(\theta) = \mathbb{E}_{(x,v,y^\star)\sim\mathcal{D}}\, \mathbb{E}_{\hat{y}\sim p_S(\cdot|x,v)} \left[\frac{1}{|\hat{y}|} \sum_{n=1}^{|\hat{y}|} \mathrm{JSD}\Big(p_T(\cdot \mid x, v, y^\star, \hat{y}_{<n}) \,\Big\|\, p_S(\cdot \mid x, v, \hat{y}_{<n}) \Big) \right]. \tag{2}$$

This is the backbone: at every token, the privileged teacher provides a target and the student is pulled toward it. What it does not yet do is ensure the student's predictions actually depend on v, nor distinguish positions where vision is decisive from positions where it is irrelevant. The next two subsections supply each piece.

3.2 Visual Awareness Contrastive Term

Eq. (2) admits a shortcut solution. The teacher shares the student's architecture and is itself prone to vision under-utilization. With the privileged answer $y^\star$ in context, it is especially well-positioned to produce plausible supervision from text alone, without consulting v. A student that matches such fabricated supervision can minimize $\mathcal{L}_{\text{distill}}$ with continuations that are textually coherent but under-utilize the video. For example, in Figure 1's sample, "In what order did the person put the objects in the backpack?", the teacher could confidently encourage the student to generate an ordering independent of the video, instead relying on common packing patterns such as placing a laptop last.

We counteract this shortcut with a contrastive term that pushes the student's with-vision distribution *away* from its counterpart that is not conditioned on v:

$$\mathcal{L}_{\text{contra}}(\theta) = -\,\mathbb{E} \left[\frac{1}{|\hat{y}|} \sum_{n=1}^{|\hat{y}|} \mathrm{JSD}\Big(p_S(\cdot \mid x, v, \hat{y}_{<n}) \,\Big\|\, \mathrm{sg}\big[p_S(\cdot \mid x, \hat{y}_{<n})\big] \Big) \right]. \tag{3}$$

Stop-gradient on the vision-ablated branch ensures that we shape the with-vision predictions rather than degrading the no-vision branch. The minus sign expresses that we *maximize* the divergence, so the student's predictions become sensitive to v. Intuitively, $\mathcal{L}_{\text{distill}}$ specifies *what* the student should predict, while $\mathcal{L}_{\text{contra}}$ specifies that those predictions must be generated by *utilizing* the video.

3.3 Vision Reliance Gap Gate

Finally, our approach distinguishes visually decisive positions. Many token positions do not need supervision: stylistic or purely linguistic choices ("therefore" vs. "thus"), boilerplate scaffolding, and any stretch of reasoning that follows from x and $\hat{y}_{<n}$ without necessitating v. Applying divergence

pressure on these tokens destabilizes distributions that do not require adjustment, so we prefer to concentrate supervision on positions that actually require the video. As a result, we introduce a gate that directly localizes the signal to visually-sensitive positions.

Per-token vision reliance. At each position n, we measure how much each policy's prediction changes when vision is removed from its context:

$$r_n^T = \mathrm{sg}\Big[\mathrm{JSD}\Big(p_T(\cdot \mid x, v, y^\star, \hat{y}_{<n}) \,\Big\|\, p_T(\cdot \mid x, y^\star, \hat{y}_{<n})\Big)\Big], \tag{4}$$

$$r_n^S = \mathrm{sg}\Big[\mathrm{JSD}\Big(p_S(\cdot \mid x, v, \hat{y}_{<n}) \,\Big\|\, p_S(\cdot \mid x, \hat{y}_{<n})\Big)\Big]. \tag{5}$$

Both quantities are detached from the computation graph. The proposed *vision reliance gap* at token n is

$$g_n = r_n^T - r_n^S. \tag{6}$$

A large positive g_n means the teacher's prediction at this position genuinely depends on the video while the student's does not. These are precisely the tokens where the student is under-utilizing the vision input and where our objective should apply the strongest pressure. Tokens for which $g_n \approx 0$ are ones where teacher and student agree on how much vision matters, whether because the position is stylistic, text-only derivable, or the student is correctly using the video, and are less emphasized.

Two-stage normalization. Raw JSD values over a large vocabulary are numerically small and concentrate near zero, so multiplying the per-token losses by g_n directly would prevent propagation of the training signal. We therefore apply a two-stage normalization along each trajectory. First, min-max rescaling stretches g_n to the unit interval,

$$\tilde{g}_n = \frac{g_n - \min_k g_k}{\max_k g_k - \min_k g_k}, \tag{7}$$

which recovers the relative structure of visually-sensitive positions independent of the absolute JSD magnitude. Second, we rescale so the trajectory mean of the gate is one,

$$g_n^\star = \frac{\tilde{g}_n}{\frac{1}{|\hat{y}|}\sum_{k=1}^{|\hat{y}|} \tilde{g}_k}. \tag{8}$$

This keeps the per-trajectory loss magnitude in the same regime as the ungated baseline and prevents the gate from affecting the learning rate.

Gated objective. The gate is applied element-wise to both loss terms. Writing ℓ_n^{distill} and ℓ_n^{contra} for the per-position integrands of Eqs. (2) and (3), the full objective is

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{1}{|\hat{y}|}\sum_{n=1}^{|\hat{y}|} g_n^\star\big(\ell_n^{\text{distill}} + \lambda\, \ell_n^{\text{contra}}\big)\right], \tag{9}$$

with $\lambda > 0$ controlling the vision-aware strength and $g_n^\star$ treated as a constant during backpropagation. The gate increases the distillation and awareness signals on the tokens where the teacher exploits visual input and the student does not.

4 Experiments

We evaluate VIOS on 12 video understanding benchmarks against contemporary video reasoning systems and across 3 base models, spanning model families and sizes, then examine training efficiency and ablate each component of the method. Section 4.1 describes the setup. Section 4.2 reports the main comparison. Section 4.3 examines training efficiency. Section 4.4 ablates each component independently.

4.1 Setup

Training Strategy. We apply VIOS to Qwen2.5-VL-7B [2], InternVL2.5-8B, and InternVL2.5-4B [3] using the Video-R1-260k corpus [7]. Because VIOS distills along on-policy student rollouts, the base policy must be able to produce a chain of thought. Therefore, for Qwen2.5-VL-7B we initialize from Qwen2.5-VL-7B-SFT, the SFT-tuned base model released with Video-R1 [7].

Model	General			Knowledge				Temporal		Long Video		Spatial	
	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Video-R1 [7]	64.42	58.37	30.76	63.04	**49.22**	23.57	24.50	71.58	50.81	55.42	**63.11**	35.25	49.17
VideoRFT [38]	62.25	58.26	28.26	64.96	47.44	24.76	24.10	72.65	**53.35**	54.08	61.82	35.10	48.92
VideoChat-R1 [23]	62.40	57.07	29.47	63.68	47.67	23.33	23.30	72.43	52.16	53.85	57.82	33.08	48.02
VideoChat-R1.5 [44]	62.17	56.89	28.56	63.20	**49.22**	25.00	23.20	72.28	48.40	54.82	58.05	31.95	47.81
Video-R2 [27]	62.08	56.04	27.42	64.80	48.33	20.00	**25.40**	72.64	51.68	50.19	58.00	32.56	47.43
Qwen2.5-VL-SFT [2, 7]	60.72	55.15	29.55	59.36	47.22	23.10	22.40	67.25	49.82	52.51	55.70	33.44	46.35
+ VIOS (Ours)	**65.50**	**59.59**	**31.36**	**65.76**	47.44	**25.48**	24.90	**72.77**	52.24	**55.87**	62.74	**35.73**	**49.95**
InternVL2.5-4B [3]	61.27	53.19	27.12	52.96	34.89	19.52	15.90	62.25	52.40	43.68	57.64	33.72	42.88
+ VIOS (Ours)	**68.42**	**57.15**	**30.91**	**57.44**	**40.44**	**21.19**	**19.80**	**68.01**	**57.43**	**51.61**	**63.57**	**34.14**	**47.51**
InternVL2.5-8B [3]	62.42	57.93	30.38	56.48	43.56	19.29	17.90	70.12	57.11	51.16	61.96	**38.04**	47.20
+ VIOS (Ours)	**69.95**	**61.52**	**33.18**	**60.96**	**45.89**	**23.81**	**22.50**	**72.21**	**61.74**	**58.71**	**66.79**	36.33	**51.13**

Table 1: Main results across 12 video understanding benchmarks. VIOS consistently improves all three backbone models, achieving average gains of 3.60, 3.93, and 4.63 points. On Qwen2.5-VL [2, 7], VIOS ranks first or second on 11 of 12 benchmarks while requiring over $9\times$ fewer training steps than the baselines (e.g. Video-R1 [7] requires 1,000 training steps).

Benchmarks. We evaluate on 12 public video understanding benchmarks spanning five categories, chosen to cover the full breadth of capabilities required of a modern video MLLM.[2] For *general video understanding* we use MVBench [22], Video-MME [8], and MINERVA [29]. For *knowledge-intensive reasoning* we use MMVU [49], Video-MMMU [14], VideoMathQA [32], and SciVideoBench [6]. For *temporal reasoning* we use TempCompass [24] and TVBench [5]. For *long video understanding* we use LongVideoBench [40] and MLVU [50]. For *spatial reasoning* we use VSI-Bench [45]. Together these benchmarks probe short and long horizons, abstract knowledge and concrete perception, and a wide range of question formats. They form a broad and widely studied cross section of video understanding, so consistent gains across them are intended to validate VIOS as a fundamental improvement to post-training for modern video models. Following Video-R1, we use the multiple-choice subset of MMVU. Subtitles are excluded from Video-MME.

Implementation. During training we cap the number of video frames at 16 and the number of vision tokens per frame at 128. At inference these are doubled to 32 frames and 256 tokens per frame with temperature set to 0.01 and top_p set to 0.001. The contrastive term in Eq. (9) uses $\lambda = 0.5$. The EMA teacher uses decay $\alpha = 0.99$. Gate normalization and stop-gradient placement follow Section 3. Each model is trained for 100 VIOS steps on 8 NVIDIA H200 GPUs. See Appendix B for hyperparameters, Appendix C for λ and α ablations, and Appendix D for evaluation.

Baselines. We compare VIOS against the corresponding base model across Qwen2.5-VL-7B-SFT [2, 7], InternVL2.5-8B, and InternVL2.5-4B [3] to isolate the contribution of VIOS post-training. Qwen2.5-VL-7B is the backbone most widely adopted by contemporary video reasoning work, so anchoring our study to the same starting point ensures a fair comparison against recent work, while varying the backbone additionally shows the generality of our idea. We compare against contemporary video reasoning models that span the design space of recent work. Video-R1 [7], VideoChat-R1 [23], and Video-R2 [27] augment GRPO with spatio-temporal rewards and objectives. VideoRFT [38] designs rewards that ground reasoning in visual evidence. VideoChat-R1.5 [44] instead interleaves iterative perception with GRPO. Together these methods cover the dominant approaches to post-training video reasoners and provide a representative point of reference for evaluating VIOS.

4.2 Main Results

Table 1 reports performance across the 12 benchmarks. On Qwen2.5-VL [2, 7], VIOS improves the average by 3.60 points, ranking first on 8 benchmarks (MVBench [22], Video-MME [8], MIN-

[2]We also evaluate on four static image benchmarks covering perception, charts, and knowledge [9, 28, 36, 47] (Appendix E).

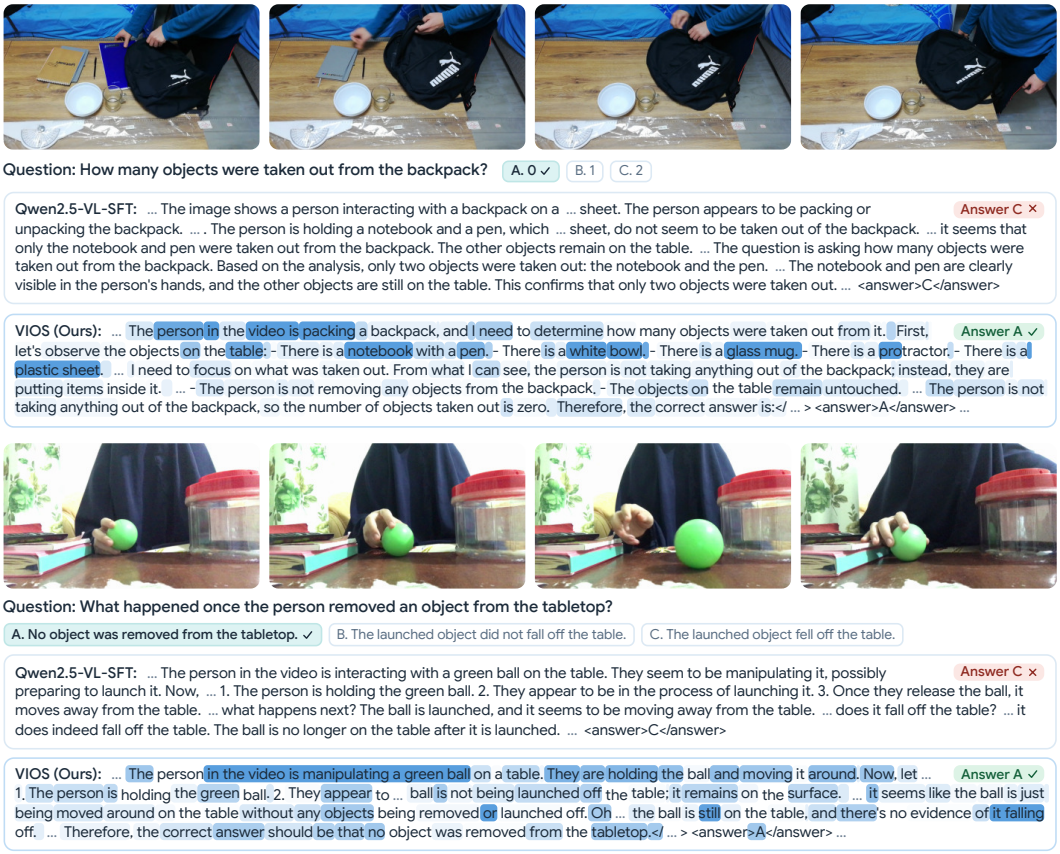


Figure 2: Comparison of reasoning traces between base model and VIOS. The VIOS rollout is highlighted representing the per-token loss at that position, with darker indicating higher loss.

ERVA [29], MMVU [49], VideoMathQA [32], TempCompass [24], LongVideoBench [40], and VSI-Bench [45]) and second on 3 more (TVBench [5], MLVU [50], and SciVideoBench [6]). The largest gains over the base model are on MLVU (+7.0), MMVU (+6.4), TempCompass (+5.5), MVBench (+4.8), and Video-MME (+4.4). These improvements generalize across model families, with VIOS boosting the average performance of InternVL2.5-4B and InternVL2.5-8B [3] by 4.63 and 3.93 points, respectively. Notably, VIOS achieves these results using a single on-policy rollout per example, no external judge model, and over $9\times$ fewer training steps than GRPO-based systems (e.g. Video-R1 [7], which requires 1,000 training steps).

Figure 2 presents qualitative examples of the thinking chains generated by the base model, Qwen2.5-VL-SFT [2, 7], and its VIOS post-trained counterpart. The loss correctly captures the tokens that are relevant to the video and to answering the questions. In the first example, the loss highlights each of the items on the table and notes that the individual is packing, and in the second example the loss highlights the individual's action upon the ball and identifies where the student mentioned no evidence of the ball falling off. It is important to note that high loss may mean the teacher assigned a higher probability to that token than the student, not that VIOS necessarily thinks those tokens are incorrect.

4.3 Efficiency Comparison

We next quantify VIOS's training cost relative to GRPO-based baselines, specifically vanilla GRPO and Video-R1 [7] since we utilize the same training corpus. For a fair comparison, all experiments in this section are initialized from Qwen2.5-VL-SFT [2, 7], the base model used by Video-R1. We rerun VIOS with the same per-step training configuration used by Video-R1 (4 GPUs, per-device batch size 1, learning rate 1×10^{-6}). Since this reduces the samples processed per step, we run 110 iterations rather than 100. Both baselines are obtained identically following Video-R1's publicly

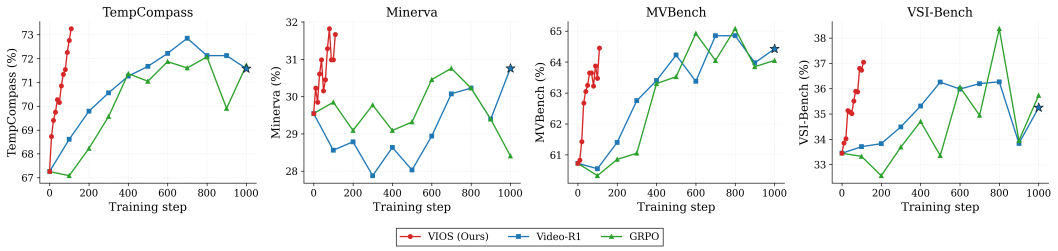


Figure 3: Accuracy across training on four representative benchmarks. VIOS matches or exceeds Video-R1's final accuracy within roughly 100 steps, compared with approximately 1000 steps for Video-R1 and GRPO. Stars indicate the performance of the public Video-R1 checkpoint.

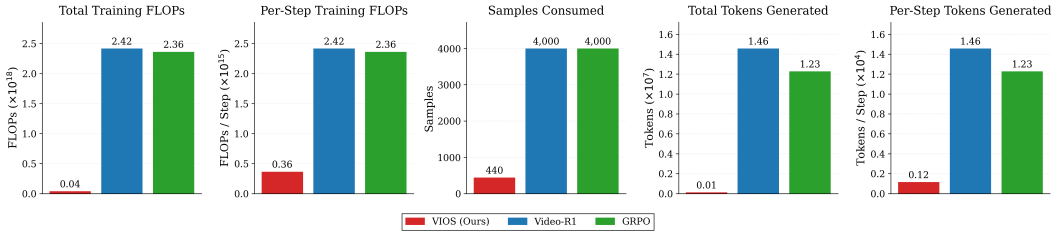


Figure 4: Training cost comparison of VIOS, Video-R1, and GRPO. VIOS substantially reduces total and per-step compute, sample consumption, and token generation through a single on-policy rollout per step. See Appendix A for FLOP calculations.

		General			Knowledge				Temporal		Long Video		Spatial	
Model	**Steps**	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Video-R1	1000	64.42	58.37	30.76	63.04	**49.22**	23.57	24.50	71.58	50.81	**55.42**	**63.11**	35.25	49.17
VIOS† (Ours)	**110**	**64.45**	**58.63**	**31.67**	**65.12**	46.33	**23.60**	**25.24**	**73.25**	**52.00**	54.00	61.68	**37.05**	**49.42**

Table 2: Under Video-R1's training configuration, VIOS achieves higher average accuracy and outperforms Video-R1 on 9 of 12 benchmarks in over 9× fewer steps.

available training scripts. For vanilla GRPO, we disable all of Video-R1's GRPO augmentations. Even with the slight increase in steps, VIOS still trains for over 9× fewer steps than Video-R1.

Table 2 reports results under this configuration. VIOS matches or exceeds Video-R1 [7] on 9 out of 12 benchmarks and outperforms on average. Figure 3 traces accuracy as a function of training step on four representative benchmarks for VIOS, Video-R1, and a GRPO baseline trained from the same starting point. VIOS gains performance at a consistent and very accelerated pace compared to the two baselines that require close to 1000 steps to reach comparable accuracy. Figure 4 reports five complementary measures of training cost. VIOS consumes a small fraction of the total compute, samples, and generation of either baseline. Specifically, it uses 60× fewer total training FLOPs, 6.7× fewer per-step training FLOPs, 111× fewer total tokens generated, and 12× fewer per-step tokens generated than Video-R1 [7].

4.4 Ablation Study

We ablate the three components of VIOS independently: the EMA teacher, the visual awareness contrastive objective, and the visual reliance gap gate. All variants stem from the main Qwen2.5-VL-SFT [2, 7] configuration detailed in Section 4.1.

Teacher update rule. Table 3 compares an EMA-updated teacher with a frozen teacher held at the base model. Both improve substantially over SFT, but the EMA teacher consistently outperforms the frozen one, by 0.7 points on average and on nine of twelve benchmarks. The largest gains occur on reasoning-intensive benchmarks (VideoMathQA, Video-MME, MINERVA, MLVU), where the frozen teacher's rationalization ability imposes a fixed ceiling. By co-evolving with the student, the

Variant	General			Knowledge				Temporal		Long Video		Spatial	
	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Qwen2.5-VL-SFT	60.72	55.15	29.55	59.36	47.22	23.10	22.40	67.25	49.82	52.51	55.70	33.44	46.35
Frozen Teacher	65.07	58.15	29.92	65.60	**47.78**	23.81	24.10	**72.92**	**52.71**	54.90	61.50	34.90	49.28
EMA Teacher (Ours)	**65.50**	**59.59**	**31.36**	**65.76**	47.44	**25.48**	**24.90**	72.77	52.24	**55.87**	**62.74**	**35.73**	**49.95**

Table 3: Ablation on the teacher update rule. The EMA teacher outperforms the frozen teacher on 9 of 12 benchmarks, with the largest gaps on reasoning-heavy benchmarks.

Variant	General			Knowledge				Temporal		Long Video		Spatial	
	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Qwen2.5-VL-SFT	60.72	55.15	29.55	59.36	47.22	23.10	22.40	67.25	49.82	52.51	55.70	33.44	46.35
No Gate	64.60	58.11	30.45	64.96	47.00	**27.86**	**25.10**	72.19	**52.28**	55.80	60.95	33.13	49.37
Gate (Ours)	**65.50**	**59.59**	**31.36**	**65.76**	**47.44**	25.48	24.90	**72.77**	52.24	**55.87**	**62.74**	**35.73**	**49.95**

Table 4: Ablation on the selection gate. The reliance gap gate outperforms ungated distillation on 9 of 12 benchmarks.

Variant	General			Knowledge				Temporal		Long Video		Spatial	
	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Qwen2.5-VL-SFT	60.72	55.15	29.55	59.36	47.22	23.10	22.40	67.25	49.82	52.51	55.70	33.44	46.35
Distill.	64.23	56.96	31.06	62.56	**48.00**	24.29	24.50	69.88	49.98	53.25	59.57	34.23	48.21
Distill. + Gate	63.32	57.74	30.15	64.00	46.11	23.10	24.70	70.93	**52.55**	54.23	61.09	**36.41**	48.69
Distill. + Gate + Contra. (Ours)	**65.50**	**59.59**	**31.36**	**65.76**	47.44	**25.48**	**24.90**	**72.77**	52.24	**55.87**	**62.74**	35.73	**49.95**

Table 5: Ablation on training-objective components. Adding distillation, gating, and contrastive learning progressively improves average accuracy, demonstrating the importance of each component.

EMA teacher avoids this ceiling in principle. The empirical gap can be attributed to our training budget of 100 iterations with EMA decay $\alpha = 0.99$ which limits the compounding of co-evolution.

Visual reliance gap gate. Table 4 compares our visual reliance gap gate against an ungated distillation loss. The gated configuration wins on 9 of 12 benchmarks and improves the average by 0.6 points, with the largest gains on VSI-Bench [45] (+2.6), MLVU [50] (+1.8), and Video-MME [8] (+1.5). These results demonstrate the benefit of selectively weighting supervision rather than treating all token positions equally. By emphasizing positions where the teacher and student differ in visual reliance, the gate directs learning toward predictions where the student may be under-utilizing vision.

Training-objective components. Table 5 adds the visual reliance gap gate and the visual awareness contrastive objective cumulatively over distillation. Each component contributes a positive increment that compounds. Distillation alone improves the base model by 1.9 points by giving the student dense per-token targets. The gate adds another 0.5 points by focusing those targets on visually-sensitive positions. The contrastive term then adds a further 1.3 points by penalizing vision under-utilization to reach full VIOS performance. Its gains span benchmark categories (MVBench [22] +2.2, VideoMathQA [32] +2.4, Video-MME [8] +1.9, MMVU [49] +1.8, MLVU [50] +1.7), reflecting that a privileged-answer teacher can produce plausible supervision from text alone across many task types and that forcing the student's distribution to depend on the video closes that loophole.

5 Conclusion

We presented VIOS, a post-training procedure that recovers dense, per-token, vision-aware supervision from a single on-policy rollout of a model, with no external reward, judge, or larger teacher. The method follows from a single observation: a model can audit its own visual reliance by comparing its next-token distribution with and without the video in context, and the gap between this quantity for a privileged teacher and a student rollout localizes the positions where the student under-utilizes what it saw. Across 12 video understanding benchmarks, VIOS consistently improves three different backbones, while requiring over $9\times$ fewer training steps, $60\times$ fewer training FLOPs, and $111\times$ fewer tokens generated than contemporary video reasoning systems. These results suggest that the supervision needed for ensuring vision utilization is available in the model itself, recoverable by having it watch itself watch. Extending this principle of self-auditing input reliance beyond video, to other modalities and to multimodal agents, is a promising direction for future work.

6 Limitations and Future Work

Scale and initialization. Our experiments are conducted on Qwen2.5-VL-7B-SFT [2, 7], InternVL2.5-4B, and InternVL2.5-8B [3] and the Video-R1-260k [7] training corpus. While this matches the setup used by recent video reasoning systems and isolates the contribution of post-training, it leaves open how VIOS scales to substantially larger models. VIOS also requires a base model that can already produce a coherent chain of thought, which is why for Qwen2.5-VL we initialize from an SFT checkpoint. Testing larger models and less reasoning-ready initializations is a natural next step to assess whether the per-step compute savings continue to compound and how broadly VIOS applies.

Training cost and supervision. Computing the reliance gap requires evaluating the student and teacher both with and without the video at every position, roughly doubling the per-step forward cost relative to standard distillation, although this overhead is more than offset by the elimination of multi-rollout RL (Section 4.3). Our setting also assumes a privileged ground-truth final answer is available for each training example, which restricts VIOS to VQA-style data with ground-truth answers. Future work could make reliance-gap estimation more efficient and relax this answer requirement to extend VIOS to open-ended video understanding.

Grounding and correctness. On the objective itself, the contrastive term promotes visual utilization but does not enforce grounding to specific frames or regions, so hallucinations consistent with the video's general content can still occur. Correctness enters only implicitly through the privileged-answer teacher rather than as explicit supervision on the final answer, so VIOS inherits whatever rationalization biases the teacher exhibits at any given training step. Addressing these limitations is a promising direction for future work, whether by adding an explicit grounding objective tied to specific frames or regions or incorporating a verifiable correctness signal alongside the privileged-answer teacher.

Broader extensions and diagnostics. More broadly, the self-auditing principle behind VIOS—measuring how a prediction shifts when a particular input is removed—is not specific to video. Applying it to other modalities such as audio, depth, and structured sensor streams, as well as to multimodal agents whose decisions must depend on what they perceive, is a direct extension. The reliance gap could also serve as a diagnostic in its own right, providing a token-level audit of whether existing video reasoners actually use their visual input and complementing coarse benchmark accuracy. Exploring these extensions could help establish how broadly self-auditing input reliance generalizes beyond video reasoning.

Acknowledgements

This research has been supported by computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at the University of Texas at Austin. UT Austin is supported in part by the IFML NSF AI Institute and Lyda Hill Philanthropies. We thank the UT Vision group for their valuable feedback and insightful discussions. The authors declare no competing interests.

References

[1] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

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

[3] 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.

[4] Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[5] Daniel Cores, Michael Dorkenwald, Manuel Mucientes, Cees G. M. Snoek, and Yuki M. Asano. TVBench: Redesigning video-language evaluation. *arXiv preprint arXiv:2410.07752*, 2024.

[6] Andong Deng, Taojiannan Yang, Shoubin Yu, Lincoln Spencer, Mohit Bansal, Chen Chen, Serena Yeung-Levy, and Xiaohan Wang. SciVideoBench: Benchmarking scientific video reasoning in large multimodal models. *arXiv preprint arXiv:2510.08559*, 2025.

[7] 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. In *Advances in Neural Information Processing Systems*, 2025.

[8] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[9] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision (ECCV)*, 2024.

[10] Jihao Gu, Yingyao Wang, Meng Cao, Pi Bu, Jun Song, Bo Zheng, Yancheng He, and Shilong Li. Token preference optimization with self-calibrated visual-anchored rewards for hallucination mitigation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

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

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[13] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[14] 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.

[15] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[16] Mykola Khandoga, Rui Yuan, and Vinay Kumar Sankarapu. Beyond uniform credit: Causal credit assignment for policy optimization. *arXiv preprint arXiv:2602.09331*, 2026.

[17] Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[18] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

[19] Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[20] Chen Li, Nazhou Liu, and Kai Yang. Adaptive group policy optimization: Towards stable training and token-efficient reasoning. *arXiv preprint arXiv:2503.15952*, 2025.

[21] Jingyao Li, Jingyun Wang, Molin Tan, Haochen Wang, Cilin Yan, Likun Shi, Jiayin Cai, Xiaolong Jiang, and Yao Hu. CrossVid: A comprehensive benchmark for evaluating cross-video reasoning in multimodal large language models. In *AAAI Conference on Artificial Intelligence*, 2026.

[22] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[23] 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. In *Advances in Neural Information Processing Systems*, 2025.

[24] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. TempCompass: Do video LLMs really understand videos? In *Findings of the Association for Computational Linguistics: ACL*, 2024.

[25] Kevin Lu and Thinking Machines Lab. On-policy distillation. Thinking Machines Lab: Connectionism, October 2025. https://thinkingmachines.ai/blog/on-policy-distillation.

[26] Mi Luo, Zihui Xue, Alex Dimakis, and Kristen Grauman. When thinking drifts: Evidential grounding for robust video reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[27] Muhammad Maaz, Hanoona Rasheed, Fahad Shahbaz Khan, and Salman Khan. Video-R2: Reinforcing consistent and grounded reasoning in multimodal language models. *arXiv preprint arXiv:2511.23478*, 2025.

[28] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

[29] Arsha Nagrani, Sachit Menon, Ahmet Iscen, Shyamal Buch, Ramin Mehran, Nilpa Jha, Anja Hauth, Yukun Zhu, Carl Vondrick, Mikhail Sirotenko, Cordelia Schmid, and Tobias Weyand. MINERVA: Evaluating complex video reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[30] Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, and Xingzhong Xu. NGRPO: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.

[31] Yeji Park, Deokyeong Lee, Junsuk Choe, and Buru Chang. ConVis: Contrastive decoding with hallucination visualization for mitigating hallucinations in multimodal large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[32] Hanoona Rasheed, Abdelrahman Shaker, Anqi Tang, Muhammad Maaz, Ming-Hsuan Yang, Salman Khan, and Fahad Shahbaz Khan. VideoMathQA: Benchmarking mathematical reasoning via multimodal understanding in videos. *arXiv preprint arXiv:2506.05349*, 2025.

[33] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[34] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[35] Yunlong Tang, Jing Bi, Pinxin Liu, Zhenyu Pan, Zhangyun Tan, Qianxiang Shen, Jiani Liu, Hang Hua, Junjia Guo, Yunzhong Xiao, Chao Huang, Zhiyuan Wang, Susan Liang, Xinyi Liu, Yizhi Song, Yuhe Nie, Jia-Xing Zhong, Bozheng Li, Daiqing Qi, Ziyun Zeng, Ali Vosoughi, Luchuan Song, Zeliang Zhang, Daiki Shimada, Han Liu, Jiebo Luo, and Chenliang Xu. Video-LMM post-training: A deep dive into video reasoning with large multimodal models. *arXiv preprint arXiv:2510.05034*, 2025.

[36] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Information Processing Systems*, 2024.

[37] Hieu Tran, Zonghai Yao, and Hong Yu. Exploiting tree structure for credit assignment in RL training of LLMs. *arXiv preprint arXiv:2509.18314*, 2025.

[38] Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. VideoRFT: Incentivizing video reasoning capability in MLLMs via reinforced fine-tuning. In *Advances in Neural Information Processing Systems*, 2025.

[39] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

[40] Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. LongVideoBench: A benchmark for long-context interleaved video-language understanding. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024.

[41] Peng Xu, Shengwu Xiong, Jiajun Zhang, Yaxiong Chen, Bowen Zhou, Chen Change Loy, David Clifton, Kyoung Mu Lee, Luc Van Gool, et al. MARS2 2025 challenge on multimodal reasoning: Datasets, methods, results, discussion, and outlook. In *ICCV Workshops*, 2025.

[42] Wenda Xu, Rujun Han, Zifeng Wang, Long T. Le, Dhruv Madeka, Lei Li, William Yang Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. *arXiv preprint arXiv:2410.11325*, 2024.

[43] Zihui Xue, Mi Luo, and Kristen Grauman. Seeing the arrow of time in large multimodal models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[44] 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.

[45] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. *arXiv preprint arXiv:2412.14171*, 2024.

[46] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[47] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[48] 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.

[49] Yilun Zhao, Lujing Xie, Haowei Zhang, Guo Gan, Yitao Long, Zhiyuan Hu, Tongyan Hu, Weiyuan Chen, Chuhan Li, Junyang Song, Zhijian Xu, Chengye Wang, Weifeng Pan, Ziyao Shangguan, Xiangru Tang, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. MMVU: Measuring expert-level multi-discipline video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[50] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: A comprehensive benchmark for multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

[51] He Zhu, Junyou Su, Peng Lai, Ren Ma, Wenjia Zhang, Linyi Yang, and Guanhua Chen. Anchored supervised fine-tuning. *arXiv preprint arXiv:2509.23753*, 2025.

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 abstract and introduction clearly outline the key objectives and outcomes of the research, including the problem being addressed, the proposed approach or methodology, and the significance of the findings.

   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: Yes, we discussed the limitations and potential failure cases of our work in Section 6 and Appendix G.

   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: Not applicable, as this paper does not include any 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: We provide all the necessary information to reproduce the main experimental results in Section 4.1 and Appendices B, D, and J.

   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: The code and data will be released.

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: The paper includes comprehensive information on key training configurations.

   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: [Yes].

   Justification: We conducted multi-run evaluations for a subset of the big experiments and observed no significant variance in the results. The stability observed across seeds supports the reliability and consistency of the reported results (Appendix F).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide all the necessary information on compute resources to reproduce the main experimental results in Section 4.1 and Appendices B and D.

   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 have reviewed the NeurIPS Code of Ethics and confirm that the research conducted in this paper adheres to its guidelines.

   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: Please refer to the discussion provided in Appendix I.

    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: [Yes]

    Justification: We take responsible release seriously and ensure that any code or evaluation data we release avoids including sensitive content or misuse-enabling functionality. Our work is intended for academic benchmarking and insight into multimodal large language model reasoning.

    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: Any pretrained models referenced are publicly available and subject to their original licenses and usage policies.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

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

13. **New assets**

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

    Answer: [Yes]

    Justification: We will provide guideline documentation as part of the release.

    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: This study does not involve any human subjects.

    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: This study does not involve any human subjects.

    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: [N/A].

    Justification: LLM is used only for writing, editing, or formatting purposes.

    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.

A Training FLOP Accounting

We estimate the per-step FLOPs in Figure 4 using the standard $2N/6N$ per-token approximation [13, 15]: $2N$ for forward-only passes and $6N$ for forward-and-backward passes, where N excludes embedding parameters. Since MLLMs route language and vision tokens through separate modules, we count their FLOPs independently, using $N_{\mathrm{LLM}} = 6{,}525{,}621{,}760$ and $N_{\mathrm{ViT}} = 675{,}044{,}864$ for Qwen2.5-VL-7B [2]. Let τ_{L}^{p} and τ_{V}^{p} denote the total language and vision tokens processed in pass p, summed over all relevant trajectories. The per-step cost is

$$\begin{aligned}\mathrm{FLOPs} = \underbrace{2\big(N_{\mathrm{LLM}}\tau_{\mathrm{L}}^{\mathrm{pre}} + N_{\mathrm{ViT}}\tau_{\mathrm{V}}^{\mathrm{pre}}\big)}_{\text{rollout prefill}} + \underbrace{2N_{\mathrm{LLM}}\tau_{\mathrm{L}}^{\mathrm{dec}}}_{\text{rollout decode}} \\ + \underbrace{6\big(N_{\mathrm{LLM}}\tau_{\mathrm{L}}^{\mathrm{pol}} + N_{\mathrm{ViT}}\tau_{\mathrm{V}}^{\mathrm{pol}}\big)}_{\text{policy update}} + \underbrace{2\big(N_{\mathrm{LLM}}\tau_{\mathrm{L}}^{\mathrm{aux}} + N_{\mathrm{ViT}}\tau_{\mathrm{V}}^{\mathrm{aux}}\big)}_{\text{additional forwards}}.\end{aligned}$$

Here, $\tau^{\mathrm{aux}} = \tau^{\mathrm{tch}} + \tau^{\mathrm{con}}$ for VIOS and $\tau^{\mathrm{aux}} = \tau^{\mathrm{ref}}$ for the GRPO [34] baselines (for both token types). Autoregressive decoding is language-only because the visual prefix is cached.

Pass	VIOS	Vanilla GRPO	Video-R1
Rollout prefill	Prompt and video	Shared prompt and video	Shared policy prefill; 4 temporal-reward prefills
Rollout decode	1 trajectory	8 trajectories	8 policy + 4 temporal-reward trajectories
Policy update	1 trajectory	8 trajectories	8 policy trajectories
Additional forwards	EMA teacher with/without video; student no-video contrastive pass	Frozen reference model for KL on 8 trajectories	Frozen reference model for KL on 8 policy trajectories

Table 6: Passes included in the per-step FLOP estimate. All token counters sum over the relevant trajectories; rollout counts are not additional multipliers.

Both GRPO baselines use $G = 8$ trajectories for policy training. Video-R1 [7] additionally samples $T = 4$ trajectories to compute its temporal reward. These extra trajectories contribute to generation FLOPs (prefill and decode), but not to the policy update or reference-model KL pass. The teacher, contrastive, and reference passes are forward-only. Both GRPO baselines use the publicly available Video-R1 training pipeline, modified for vLLM colocation and updated to the same dependency versions as VIOS for fair comparison.

B Training and Generation Hyperparameters

Table 7 provides the full configuration used for VIOS training rollouts and optimization. The maximum completion length is model-dependent: 768 tokens for Qwen2.5-VL [2, 7] and 4096 tokens for InternVL2.5 [3].

Hyperparameter	Value
Sampling temperature	1.1
Top-k	20
Top-p	0.95
Training seed	42
Per-device batch size	8
Learning rate	2×10^{-6}
LR scheduler	Cosine (1-epoch horizon)
Weight decay	0.01
Warmup ratio	0.0
EMA decay α	0.99 (constant)
Contrastive weight λ	0.5

Table 7: VIOS training-rollout and optimization hyperparameters. Maximum completion length differs between Qwen2.5-VL and InternVL; all other settings are shared.

Training seed. The publicly released checkpoints of the video-reasoning baselines [7, 23, 27, 38, 44] were trained by their respective authors using seed 42, the default in Hugging Face TRL, the framework on which our implementation is also based. For a consistent comparison, we likewise report VIOS trained using seed 42.

C Contrastive Weight and EMA Decay Ablations

Variant	General			Knowledge				Temporal		Long Video		Spatial	
	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Qwen2.5-VL-SFT	60.72	55.15	29.55	59.36	47.22	23.10	22.40	67.25	49.82	52.51	55.70	33.44	46.35
$\lambda = 0$	63.32	57.74	30.15	64.00	46.11	23.10	24.70	70.93	**52.55**	54.23	61.09	**36.41**	48.69
$\lambda = 0.25$	65.57	58.85	30.98	**66.24**	48.00	23.10	**25.30**	72.32	51.88	55.20	61.96	35.71	49.59
$\lambda = 0.5$ (Ours)	65.50	**59.59**	**31.36**	65.76	47.44	**25.48**	24.90	72.77	52.24	55.87	**62.74**	35.73	**49.95**
$\lambda = 0.75$	**65.72**	58.15	30.98	62.40	**49.89**	23.10	23.40	**73.82**	52.16	**56.54**	62.65	36.23	49.59
$\lambda = 1.0$	50.42	52.04	25.83	56.64	45.33	19.52	19.80	72.64	38.30	48.99	54.55	28.46	42.71

Table 8: We sweep λ while keeping all other training configurations identical. The $\lambda = 0$ setting corresponds to the Distill. + Gate variant in Table 5.

Contrastive Weight. Between $\lambda = 0.25$ and $\lambda = 0.75$ the method is stable: the three averages lie within 0.4 points of one another, no benchmark moves by more than a few points, and no single setting dominates per-benchmark, so the reported result does not rest on a finely tuned λ. Performance declines substantially on average at $\lambda = 1.0$: once the contrastive term is weighted equally to distillation, divergence maximization dominates the objective and the student drifts from the teacher rather than being pulled toward it. The operative constraint is therefore that neither distillation nor contrasting should dominate, not a precise value of λ.

Variant	General			Knowledge				Temporal		Long Video		Spatial	
	MVBench	Video-MME	MINERVA	MMVU	Video-MMMU	VideoMathQA	SciVideoBench	TempCompass	TVBench	LongVideoBench	MLVU	VSI-Bench	Average
Qwen2.5-VL-SFT	60.72	55.15	29.55	59.36	47.22	23.10	22.40	67.25	49.82	52.51	55.70	33.44	46.35
$\alpha = 0.90$	65.62	58.56	30.23	64.00	48.11	21.67	**26.10**	**73.33**	**52.75**	56.32	62.65	35.22	49.55
$\alpha = 0.99$ (Ours)	65.50	**59.59**	**31.36**	**65.76**	47.44	25.48	24.90	72.77	52.24	55.87	**62.74**	**35.73**	**49.95**
$\alpha = 0.995$	65.50	59.00	31.29	64.64	47.56	22.86	22.90	73.16	52.71	55.95	61.96	33.64	49.26
$\alpha = 0.999$	**65.75**	59.04	30.08	65.28	**48.78**	**26.43**	25.70	73.17	52.51	**56.47**	61.22	34.43	49.91
Fixed ($\alpha \to 1$)	65.07	58.15	29.92	65.60	47.78	23.81	24.10	72.92	52.71	54.90	61.50	34.90	49.28

Table 9: We sweep α while keeping all other training configurations identical. The $\alpha = 1.0$ setting corresponds to the Frozen Teacher variant in Table 3.

EMA decay. All EMA settings improve the average performance substantially over the base model, with gains on nearly every benchmark. The choice of α matters less than the choice to update the teacher at all: the frozen teacher is not the best setting on any of the twelve benchmarks and is beaten on average by three of the four EMA values, while differences among the EMA settings themselves are mostly within one to two points per benchmark.

D Evaluation Protocol and Baseline Comparison

Unified evaluation protocol. We evaluate every model in Table 1 ourselves under a single protocol. Specifically, we adopt the publicly released evaluation setup of Video-R1 [7] and VideoRFT [38] for all methods, including VIOS: at most 32 video frames, a per-frame pixel budget of $256 \times 28 \times 28$, temperature 0.01, top-p 0.001, and at most 1,024 generated tokens. All methods use the same prompt and evaluation pipeline, with InternVL-specific answer extraction. We evaluate the public checkpoints without retraining any baseline, so each retains the training configuration chosen by its authors. Separately, Table 2 matches Video-R1's per-step training configuration directly (Section 4.3).

Changes to the reference implementation. Our evaluation differs from the original Video-R1/VideoRFT implementation in four respects. First, we update dependencies to their current

versions: `vLLM` 0.7.x to 0.16.x, `torch` 2.5 to 2.10, and `transformers` 4.49 to 5.3. Second, we correct an issue in Video-R1's MVBench evaluator that silently omitted the video input for a subset of samples; the correction is applied to every model. Third, we preprocess benchmark videos into the model's input format before evaluation to reduce wall-clock time, without altering the inputs presented to the models or their outputs. Fourth, because InternVL2.5 models do not natively use `<answer>` tags, we add an InternVL-only fallback that extracts explicit final answers (e.g., "the answer is B" or "Answer: (b)") or, failing that, the last standalone uppercase option letter. The same extraction procedure is used for both InternVL base models and their VIOS checkpoints. The deviation from baselines' originally published numbers therefore stems from the baselines' own evaluation configurations, which differ substantially from one another and from our Video-R1/VideoRFT setup.

E Evaluation on Static Image Benchmarks

To further assess the generality of VIOS across visual modalities, we evaluate the same VIOS-trained Qwen2.5-VL checkpoint on four static image understanding benchmarks. As shown in Table 10, VIOS improves average accuracy by 2.64 points over the SFT baseline. The largest gains occur on BLINK [9] (+4.74) and CV-Bench [36] (+4.47), which emphasize direct visual perception, while MMMU [47] improves by 1.44 points, which is more knowledge-driven. ChartQA [28] performance remains essentially unchanged (-0.08). These results demonstrate that VIOS's visual-reliance supervision generalizes beyond video to static image understanding.

Model	ChartQA	BLINK	CV-Bench	MMMU	Average
Qwen2.5-VL-SFT	**84.64**	51.18	71.76	51.78	64.84
+ VIOS (Ours)	84.56	**55.92**	**76.23**	**53.22**	**67.48**

Table 10: Evaluation on four static image understanding benchmarks. VIOS improves average accuracy by 2.64 points without image-specific modifications, demonstrating generalization beyond video reasoning.

F Training Stability and Seed Robustness

We assess VIOS training stability against vanilla GRPO across three seeds (42–44), using common diagnostics. Both methods start from Qwen2.5-VL-SFT, with VIOS trained for 100 steps and GRPO for 1,000 steps. During experimentation only standard diagnostics were logged for faster training. Gate-weight and entropy statistics were additionally logged for seeds 43 and 44.

Reliance-gap gate distribution. Table 11 examines whether the gate progressively concentrates or disperses its weights during training.

	Seed 43		Seed 44	
Statistic	**Start**	**End**	**Start**	**End**
Mean	1.000	1.000	1.000	1.000
Std. dev.	0.148	0.148	0.150	0.148
p_{05}	0.774	0.742	0.772	0.741
p_{25}	0.992	0.995	0.993	0.996
p_{50}	1.025	1.034	1.025	1.034
p_{75}	1.053	1.064	1.053	1.064
p_{95}	1.114	1.114	1.117	1.113

Table 11: Distribution of normalized VIOS gate weights at training start and end across two seeds.

The trajectory-wise normalization (Eqs. 7–8) fixes the mean at one, while the nearly unchanged standard deviation and quantiles show that the gate neither sharpens nor flattens appreciably. Both seeds exhibit nearly identical distributions, indicating stable token weighting throughout VIOS training.

Gradient magnitudes and clipping. Table 12 compares pre-clipping gradient norms under the shared maximum gradient norm of 5.

GRPO exhibits heavy-tailed gradients, with occasional large spikes requiring clipping on approximately 4.5% of steps. In contrast, VIOS maintains substantially smaller and less variable gradients, with no clipping across any seed. This indicates more controlled optimization under VIOS.

Method	Seed	Mean	Std.	p95	p99	Max.	Clipped (%)
GRPO	42	2.45	3.90	4.76	10.8	112.4	4.4
GRPO	43	2.44	2.84	4.81	16.1	44.3	4.5
GRPO	44	2.46	2.85	4.84	10.9	57.2	4.7
VIOS	42	0.298	0.218	0.69	1.36	1.46	0.0
VIOS	43	0.309	0.306	0.61	1.06	2.89	0.0
VIOS	44	0.292	0.305	0.55	0.87	3.01	0.0

Table 12: Pre-clipping gradient-norm statistics across three seeds. Both methods use a clipping threshold of 5.

KL drift and normalized token entropy. Table 13 reports KL divergence from the base policy and token entropy measured at temperature 1 for normalization.

			Normalized token entropy			
Method	Seed	KL to base	Start	End	Δ	Min.
GRPO	42	0.0735	–	–	–	–
GRPO	43	0.0609	0.714	0.928	+0.214	0.359
GRPO	44	0.0681	0.711	0.914	+0.203	0.397
VIOS	42	0.0354	–	–	–	–
VIOS	43	0.0349	0.721	0.694	−0.027	0.620
VIOS	44	0.0335	0.715	0.701	−0.014	0.618

Table 13: KL divergence from the base policy and normalized token entropy. Entropy statistics are available for seeds 43 and 44; dashes denote unavailable measurements.

GRPO undergoes larger changes in token uncertainty, reaching low intermediate entropy before finishing substantially above its initial level. VIOS shows only mild entropy decreases and consistently maintains roughly half the KL divergence of GRPO. Together, these results suggest more stable policy evolution under VIOS.

Completion lengths and homogeneous reward groups. Table 14 compares generation lengths and, for GRPO, the fraction of groups with identical rewards and hence zero group-relative advantage.

		Completion length (tokens)			Homog.
Method	Seed	Start	End	Δ	group frac.
GRPO	42	290.6	376.5	+85.9	0.299
GRPO	43	275.5	404.1	+128.6	0.319
GRPO	44	282.9	384.3	+101.4	0.292
VIOS	42	294.1	297.2	+3.1	–
VIOS	43	295.1	293.9	−1.2	–
VIOS	44	293.0	296.3	+3.3	–

Table 14: Completion-length evolution and fraction of homogeneous-reward GRPO groups across three seeds. Group homogeneity is not applicable to VIOS.

GRPO completions grow substantially, consistent with its explicit length reward favoring 320–512 tokens, while approximately 30% of its sampled groups provide no group-relative learning signal. VIOS avoids this group-estimation inefficiency and maintains nearly unchanged completion lengths across all seeds, further demonstrating consistent training behavior.

G Reliance-Gap Gate Limitations

Although the reliance-gap gate improves performance, we identify a rare failure mode arising from noisy reliance estimates at the final answer token. Occasionally, the gate assigns a large weight to this position, amplifying both the distillation and contrastive gradients.

For the contrastive term, the answer is already text-derivable from the completed chain of thought, so the no-vision student may already place substantial probability on the correct option. Pushing the with-vision distribution away from its no-vision counterpart can therefore perturb an already-correct prediction. An analogous failure can arise in distillation: because the EMA teacher is a lagged average of the student, it may disagree with a student that has already improved onto the correct option, pulling its prediction away from that answer.

While such perturbations can occur elsewhere, the answer token is particularly fragile: its distribution is concentrated on very few tokens, and its position at the end of the output makes errors nonrecoverable. In rare cases, this produces a mismatch between the answer reached by the chain of thought and the option emitted between the answer tags.

H Additional Qualitative Results



Figure 5: Additional reasoning traces from Qwen2.5-VL-SFT and VIOS on three video understanding tasks. VIOS tokens are shaded by per-token loss, with darker meaning higher loss.

Figure 5 compares examples from Qwen2.5-VL-SFT and VIOS. In the top example, the base model stops at three objects, concluding "I don't see any more distinct objects." The VIOS loss is highest on scene and object descriptors ("unpacking," "glass-top table," "small box," "white object," "bottle") and on the counting cues "Finally" and "more," where the model commits to an item or revises its tally. It then recounts, separates out the laptop that was "already on the table," and settles on 4.

The middle example shows the loss exposing inconsistent reasoning. VIOS correctly identifies "a white bottle" and "a clear bottle," both highly weighted, but then claims "There are two containers that were incorrectly matched," which contradicts the ground truth and is not among the answer options. The loss is elevated on the visual claims behind this error ("white" cap, "black" cap). Because the answer-privileged teacher would not produce this trajectory, VIOS flags the flawed step. Although the final answer (A) is correct, the signal points to the incorrect reasoning that training should correct.

In the bottom example, the base model sees only three containers and states, "No, I don't see any additional containers or lids." VIOS names all four, and the loss falls on their distinguishing attributes ("bowl," "white container," "purple" lid, "clear jar," "silver" lid) and on the self-correction "Oh" just before the model confirms "there are indeed four containers." Across all three examples, VIOS concentrates supervision on the few tokens that depend on the video, such as object identities, attributes, and count updates, rather than spreading it evenly over the chain of thought.

I Societal Impact

VIOS reduces the compute required to post-train a competitive video reasoner by roughly $60\times$ in total training FLOPs and $111\times$ in tokens generated relative to the baselines (Section 4.3), broadening access to research groups without large GPU budgets, lowering the energy footprint of post-training, and supporting beneficial downstream applications such as assistive video description, educational content understanding, and analysis of long-form media. These capabilities are dual-use, however, and the same improvements lower the cost of automated surveillance and large-scale video classification of individuals. VIOS also inherits the biases of its base model, since it is a post-training procedure that introduces no new data or modalities. By making strong video reasoning attainable at a fraction of the usual training cost, VIOS shifts the frontier toward broader and more sustainable participation in multimodal research.

J Prompt Templates

Student prompt

<video frames>
[Question with options]
Please think about this question as if you were a human pondering deeply.
Engage in an internal dialogue using expressions such as 'let me think',
'wait', 'Hmm', 'oh, I see', 'let's break it down', etc, or other natural
language thought expressions. It's encouraged to include self-reflection
or verification in the reasoning process.
Provide your detailed reasoning between the <think> </think> tags, and
then give your final answer between the <answer> </answer> tags.
[Type-specific instruction]

Teacher prompt

<video frames>
[Question with options]
Please think about this question as if you were a human pondering deeply.
Engage in an internal dialogue using expressions such as 'let me think',
'wait', 'Hmm', 'oh, I see', 'let's break it down', etc, or other natural
language thought expressions. It's encouraged to include self-reflection
or verification in the reasoning process.
Here is the correct answer to this problem: [Answer]
Now articulate your own step-by-step reasoning that derives the same final
answer to the problem below:
Provide your detailed reasoning between the <think> </think> tags, and
then give your final answer between the <answer> </answer> tags.
[Type-specific instruction]

Figure 6: The privileged ground-truth answer is highlighted in the teacher prompt. Type-specific instruction refers to the question type. For example, for multiple-choice questions the instruction is to respond with either A, B, C, D, or E.

We utilize the same prompt template as Video-R1 for the student and include the highlighted section for the teacher prompt.