Video-HopChain: Multi-Hop Questions and Confidence-Gated Exploration for Video Reasoning Models

Nguyen Quang Trung[1] Yuhao Dong[1] Shuo Sun[2] Shuai Liu[1] Shulin Tian[1]
Kim-Hui Yap[3] Ziwei Liu[1] ✉

[1]S-Lab, Nanyang Technological University (NTU) [2]Johns Hopkins University [3]NTU

✉ Corresponding author

Project page · Code · Dataset and models

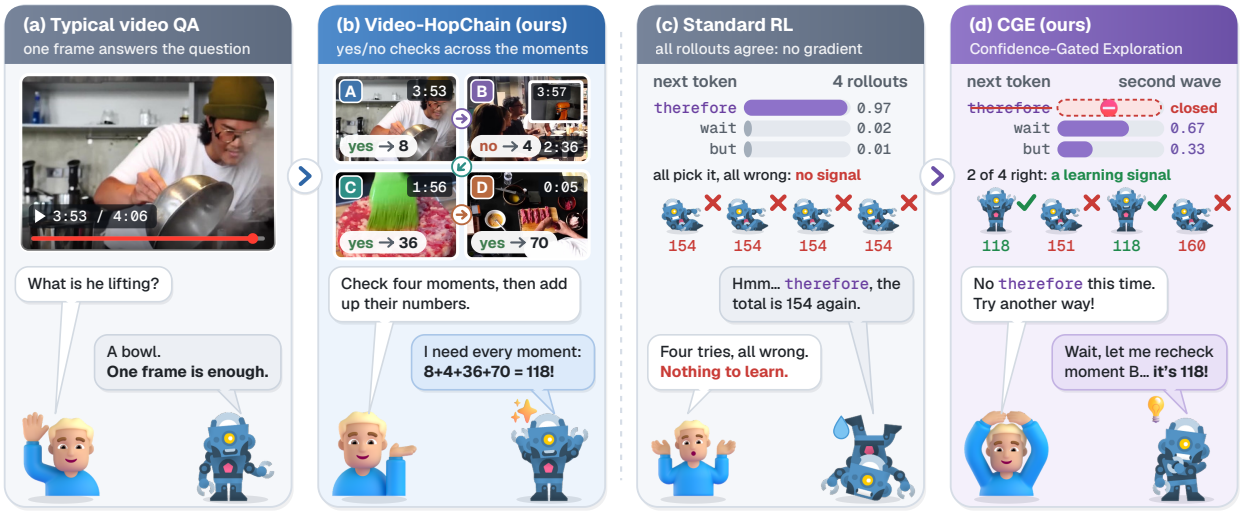


Figure 1. Video-HopChain and CGE at a glance. (a, b) A typical video question needs one frame, while a Video-HopChain question chains yes/no checks across the moments of a video and sums the numbers they select. (c, d) When every rollout of a group fails in the same way, the group gives no gradient. In the second wave, CGE closes the token that the policy is sure of, so the rollouts take new paths and the rewards differ.

Abstract

HopChain has shown on still images that multi-hop data synthesis improves vision-language reasoning, because long chain-of-thought reasoning exposes errors that compound across steps, while most data used for reinforcement learning with verifiable rewards (RLVR) rarely demands a chain of visual evidence, so these weaknesses are likely to stay unexposed. We observe the same problem in video, where this framework has not yet been explored. We therefore build Video-HopChain, a dataset of 22,550 multi-hop video questions over 13,378 videos, together with a held-out benchmark of 1,000 questions. Each question chains three to six yes/no questions about moments in one video, and each of them yields one of two integers depending on its answer. The final answer is the sum of these integers, so an exact match on that sum gives the verifiable reward that RLVR needs. We first train Qwen3-VL-8B with GRPO on a standard video dataset, and a second stage on Video-HopChain then raises the mean over eight video understanding and reasoning benchmarks from 55.4 to 57.9 and improves every one of them. Training on such a dataset, however, also exposes a known limitation of GRPO: its learning signal comes from the reward variance within a group, so hard questions whose rollouts are all incorrect and easy questions whose rollouts are all correct both leave the group with no gradient. To recover these groups at the same compute budget, we introduce Confidence-Gated Exploration (CGE). With 8 rollouts per question, CGE samples the first 4 rollouts as usual. If these 4 rollouts are either all correct or all incorrect, it then

samples the last 4 rollouts with the policy's most confident token masked inside the reasoning span, and all 8 rollouts then enter the advantage. With CGE, the mean rises further to 59.3. We release our model, Video-HopChain-8B, together with the dataset, the data generation code, and the training code.

1. Introduction

Rule-based reinforcement learning with verifiable rewards (RLVR) has become the standard route to a reasoning model, first in mathematics and code [14, 51, 73] and then in images [61, 79]. Video is the natural next domain, and a growing number of systems now train video question answering models with Group Relative Policy Optimization (GRPO) against a verifiable answer [10, 17, 18, 36, 57]. However, what such a model learns depends on the questions it is trained on. HopChain [60] made this point for still images: long chain-of-thought reasoning exposes perception, reasoning, knowledge and hallucination errors that compound across intermediate steps, yet most RLVR data asks for no reasoning chain that relies on visual evidence throughout, so these weaknesses stay largely unexposed during training. We observe the same problem in video. Much of the existing video RLVR data is already verifiable, because it is multiple choice or numeric, but the questions that would require a chain of visual evidence are often conceptual, and they are therefore hard to verify. Prior audits also report that video models often answer without looking at the video at all [31, 83], and that a model can encode what it sees and still answer from its priors [49]. We therefore design the training data so that the model must look at the video, and look at it several times, before it answers.

HopChain addresses this weakness on images with multi-hop data synthesis. It builds each question as a chain of dependent hops, in which earlier hops fix the objects and conditions that later hops need, and it ends the chain in one unambiguous number, so a wrong step changes the final answer. In addition, training on such data yields reasoning that generalizes across general understanding and reasoning benchmarks. We expect such chains to matter most in video, because the evidence a chain must revisit is spread over time rather than held within a single frame.

For this reason, we adapt this framework to video and build Video-HopChain, in which every question asks about the moments of one video rather than about the regions of one image. Each question chains three to six hops, and every hop is a yes/no question about one or two moments of the video that yields one of two integers depending on its answer, so the final answer is the sum of these integers and is verifiable by exact match. Every question includes at least one hop on the order of two moments, so a single frame does not carry the evidence that a chain needs. In addition, some hops act as selectors, where an earlier answer decides which moment a later hop examines. To generate the questions, we first caption each shot of a video with a vision-language model, and then we use a text-only language model to write the multi-hop question from those captions. The resulting dataset holds 22,550 training questions over 13,378 videos, plus 1,000 held-out videos with one question each as a benchmark.

We expect that training on this dataset improves general video understanding and reasoning, even though the dataset targets one specific reasoning type. To test this, we train Qwen3-VL-8B-Instruct with GRPO and evaluate on eight general-purpose video benchmarks. Table 1 shows the result. Training further on Video-HopChain raises the mean over the eight benchmarks from 55.4 to 57.9 and improves every one of them.

Training with GRPO, however, has a known limitation. GRPO takes its learning signal from the reward variance within a group of G rollouts of one question, so a group whose rollouts all earn the same reward carries zero advantage and contributes nothing to the update. The literature calls this failure advantage collapse [23, 81], and we call such a group a zero-variance group. Prior work handles this limitation either by discarding the group and sampling new prompts until the batch is full [73], which spends more rollouts, or by reshaping the advantage of the group it already has [34, 44].

Instead of drawing more groups, we change how each group is drawn: with $G = 8$ rollouts per question, we sample the first 4 rollouts normally. If these 4 rollouts are either all correct or all incorrect, we sample the last 4 rollouts under a top-token mask inside the reasoning span. Wherever the policy places more than $\tau = 0.95$ of its mass on one token, we drop that token and renormalize over the rest, so the model continues from a token that it samples rarely on its own. All 8 rollouts then enter the group advantage, so the intervention adds no rollouts. We call this Confidence-Gated Exploration (CGE). On top of Video-HopChain, it raises the mean over the eight benchmarks from 57.9 to 59.3. Our model, Video-HopChain-8B, combines standard RL, Video-HopChain and CGE.

In summary, we make three contributions.

- **The Video-HopChain framework.** We propose a framework that generates multi-hop questions for training video reasoning models (Section 3.2).
- **The Video-HopChain dataset and benchmark.** We release 22,550 training questions over 13,378 videos, together with 1,000 questions on 1,000 held-out videos as a benchmark built from the framework.
- **Confidence-Gated Exploration (CGE).** We introduce a simple and lightweight method that recovers zero-variance groups, which improves the results of GRPO training on Video-HopChain.

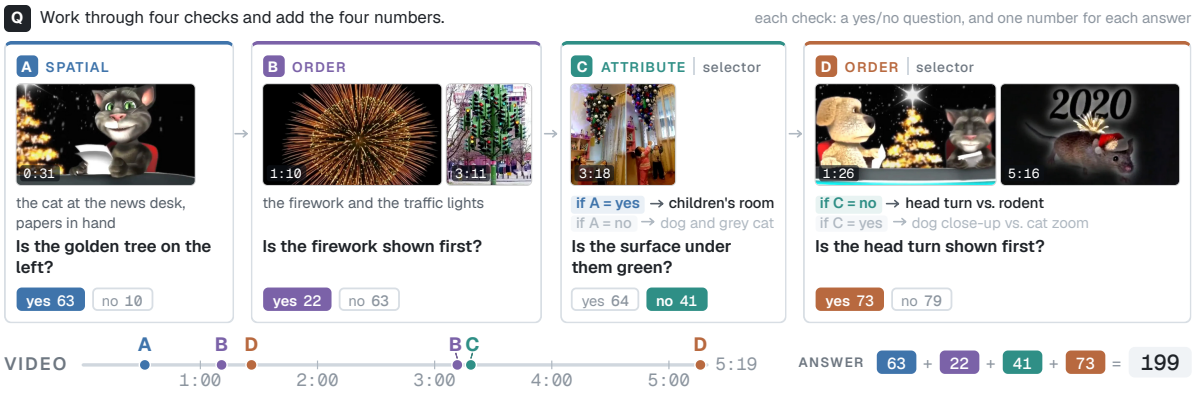


Figure 2. A Video-HopChain question. Appendix F of the supplementary gives the question text.

2. Related work

Training data for multimodal reinforcement learning. Reinforcement learning with verifiable rewards needs questions with one correct answer, so most multimodal datasets pool public benchmarks and instruction sets [17, 18]. However, audits find that a model answers a large share of long-video questions from text alone [83], shortcut-aware benchmarks measure when it answers from its priors [31], and probes show that an omnimodal model can encode a premise-perception mismatch that it never acts on [49]. The usual response removes such shortcuts after the fact, with a text-only solver [83], inside a caption-grounded synthesis loop [76], or by reducing text bias in the options [32]. In contrast, we build every question around a multi-hop structure that ties every hop to visual evidence, which prior work applies to text [28, 55] and to video reasoning [21, 53], and which recent video benchmarks measure [72]. HopChain applies it to still images [60], and we adapt it to video, where a hop becomes a yes/no question about a moment in time.

Reinforcement learning for video reasoning. Video-R1 applied GRPO to video with a temporal contrast term [17], and later systems extend this recipe to spatio-temporal grounding [36], reasoning traces and thinking with video [57, 58], key-token attribution [64], long video [10], and joint image-and-video training [18]. Almost all of them draw every group from the unmodified policy, and STRIVE changes only what a group is drawn over [2]. CGE instead changes the sampler itself, so it is orthogonal to these recipes.

Zero-variance groups. This shared sampler leaves in place the zero-variance limitation of Section 4.2, which recent work calls advantage collapse [23]. Objective-level variants reshape the estimator [78, 85], whereas other methods spend more compute, because they resample the group [73], reallocate rollouts across prompts [37, 68], branch a rollout [56], or filter by difficulty during training [1]. A third family reuses the group it already has, by reshaping the advantage [34, 44] or by recovering signal from all-incorrect groups [19], in some designs with a teacher model [22]. CGE also reuses the group, but it samples the second half of a zero-variance group differently, so that the group can regain variance.

Exploration and the loss at the token level. Our work also relates to methods that make the rollout more exploratory. Many reweight or restrict the gradient at high-entropy positions [11, 59] or at the tokens whose log-probability covaries most with the advantage [13], while others act on low-probability tokens [25, 40], on the probability mass after the rollout [27], on noise in prompt or parameter space [3, 26], or on the surprisal of the sampled token [42, 43]. Closer to the sampler, FR3E explores from high-uncertainty decision points [86], EDGE-GRPO corrects the rollout errors that drive the advantage to zero [81], and CaSP acts in the loss on the top-1 probability that we use as our trigger [46]. The closest methods perturb the rollout as we do: EEPO regenerates the second half of a group after a transient weight update [8], and REFT resamples the first reasoning token from the top-N candidates of the policy [30]. In contrast, CGE acts only on zero-variance groups, perturbs one token at a time in the sampler, and leaves the answer tokens untouched. Such a perturbation makes the rollout off-policy, as a mismatch between training and inference engines does [87], which prior work handles by masking the discrepant tokens [54] or by correcting their importance weight [15, 84]. For a deliberate perturbation [38], prefix-guided methods remove the injected tokens from the loss [5], whereas LUFFY reshapes their gradient [69]. We follow the importance-weight correction, because the sampling engine scores the masked token under the distribution it samples from, so the ratio of the clipped objective already weights that token (Section 4.2 and Appendix I of the supplementary). The tokens after a mask need no correction, since the unmodified policy samples them.

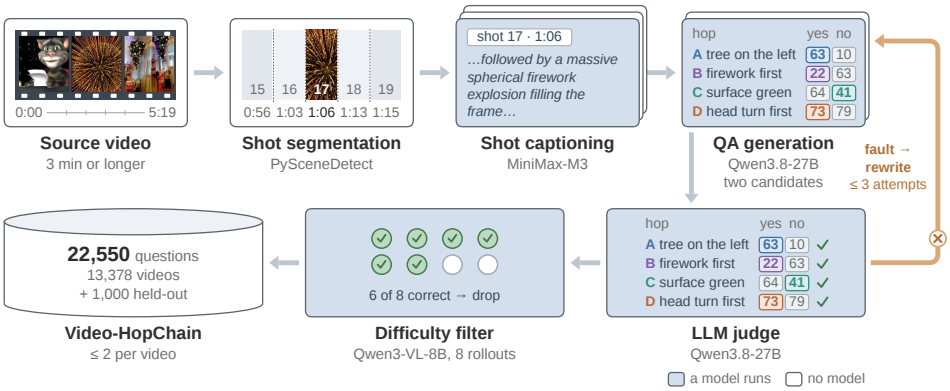


Figure 3. The Video-HopChain generator.

3. Video-HopChain

3.1. Question design

Video-HopChain is a dataset of multi-hop video questions, in which every question chains n hops over one video, with n between three and six. Every hop carries two numbers, of which the first one counts when the answer to the hop is yes and the second one when the answer is no. The model then adds the numbers of all the hops, which makes this sum the answer to the question. Figure 2 presents a sample of Video-HopChain hop by hop.

For every hop we assign the two numbers by sampling integers at random in the range 1 to 80. We then resample them until no two combinations of answers give the same total. In addition, we keep the arithmetic a plain sum, so that the model only adds the numbers that its answers select. With this design, we intend the difficulty to come from the video, not the calculation.

A hop is a yes/no question that the video settles, and we use four main categories of question for each hop, each over a different kind of visual evidence. The video settles three of these categories at one moment, whereas it settles the order category across two moments. We choose these four categories because a caption records this kind of information.

- An *order* hop asks about the temporal order of two events, so it spans two moments.
- A *spatial* hop asks how two things are arranged inside one frame, such as left, right or middle, in terms anchored to the frame rather than to the body of the person shown.
- An *action* hop asks which physical action an agent performs, such as pour or lift.
- An *attribute* hop asks about a visible property of a named object, namely its colour.

Beyond the category of each hop, we link the hops in one of two ways. In a *flat* question, which we choose with probability 0.30, every hop names the moment it asks about and does not chain to another hop, so the model can answer the hops in any order. By contrast, in a *selector* question, which we choose with probability 0.70, the answer to an earlier hop decides which of two moments a later hop asks about, so the model has to work through the hops in order.

3.2. Generation pipeline

Figure 3 shows the pipeline that turns raw video into questions. The source videos come from FineVideo [16], from LongVILA [9] as released with the OneThinker training data [18], and from the long split of Vript [70].

From video to captions. From these sources, we keep a source video only when it lasts at least three minutes, because we want the questions to stay hard and to force the model to reason over moments that lie far apart in time. We then cut the video into shots with PySceneDetect [6], and we caption every shot once with the MiniMax-M3 model [33].

Generating the question. Before we write a question from these captions, we fix its specification. To keep the questions diverse, we sample the hop count and the hop types with fixed probabilities. Table 7 reports the hop counts of the corpus that these draws produced. This specification also fixes the way the hops link and the two numbers of each hop. We then use Qwen3.8-27B as a question generator that writes two candidate questions per video from the captions and this specification.

Verifying and assembling. Once the generator writes the candidates, the same model re-examines every one of them against the same captions, and it passes a question only when it faults no hop. When the judge faults a hop, we send the question back to the generator, which writes it again from the same captions and the same specification. The judge then examines the new question. We repeat this loop until the judge reports no fault, but we drop the question when it still faults after three attempts. The pipeline then ends with a

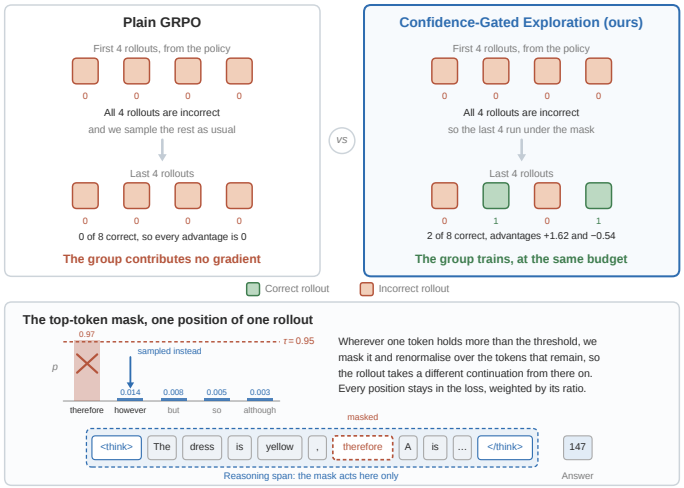


Figure 4. Confidence-Gated Exploration on one group of G rollouts.

difficulty filter, which removes every question that the base model already solves in 6 of 8 rollouts.

3.3. Statistics and split

Together, these stages yield 22,550 training questions over 13,378 videos and a held-out split of 1,000 questions over 1,000 further videos. We split by video, so no video appears on both sides. In addition, we draw the held-out side from the longer and more varied questions, stratified by hop count, video length and source.

4. Confidence-Gated Exploration

4.1. Preliminaries

We train with Group Relative Policy Optimization (GRPO) [14, 51]. For a prompt q, which here is a video and a question, the policy π_θ samples a group of G rollouts $\{o_i\}_{i=1}^{G}$, and a verifier then scores each one with a reward r_i. Reasoning work commonly rewards the answer and the response format together. We use this reward in this paper:

$$r_i = 0.8\, a_i + 0.2\, f_i, \tag{1}$$

where a_i is 1 when the answer matches the reference and f_i is 1 when the response follows the format that the system prompt asks for, namely a reasoning span inside the think tags and then an answer that carries the final integer in a boxed expression. Appendix F of the supplementary gives that system prompt in full. GRPO needs no value model, because the advantage $\hat{A}_i$ of a rollout is its reward standardized over the rewards of its own group. Every token of o_i then carries that one value. With the importance ratio $\rho_{i,t}(\theta)$ between the current policy and the policy that sampled the rollout, we maximize the token-level clipped objective:

$$\mathcal{J}(\theta) = \mathbb{E}_{q,\{o_i\}}\Bigg[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\Big(\rho_{i,t}(\theta)\,\hat{A}_i,\; \mathrm{clip}\left(\rho_{i,t}(\theta),\,1-\epsilon_{\text{low}},\,1+\epsilon_{\text{high}}\right)\hat{A}_i\Big)\Bigg]. \tag{2}$$

The clip range is asymmetric, following Yu et al. [73]. Throughout, we use $G = 8$, $\epsilon_{\text{low}} = 0.2$, $\epsilon_{\text{high}} = 0.3$ and no KL penalty.

4.2. Confidence-Gated Exploration

The advantage of GRPO has a direct consequence. When every rollout of a group earns the same reward, the group has no reward variance, so every advantage is zero. We roll out and score such a group in full, and it then contributes no gradient. Following Le et al. [34], we call such a group a *zero-variance* group. These groups appear from both directions, because a question the model cannot solve returns all-incorrect rollouts, whereas a question it has learned returns all-correct ones. Among the methods that answer this problem, the closest design to ours is EEPO [8], which also regenerates part of a group after an intervention. CGE instead keeps the group and samples its second half differently, as Figure 4 summarizes and as Algorithm 1 of the supplementary states.

When we encourage the model to explore. We draw each group in two waves of equal size, so the first wave takes $G_1 = 4$ of the 8 rollouts from π_θ, and we then score them. If those 4 rollouts do not all earn the same accuracy, the group already carries variance, so we draw the second wave

from π_θ unchanged. If they are all correct or all incorrect, however, the first wave is zero-variance, so we draw the second wave under the mask below instead. This check reads the accuracy a_i of Equation 1 rather than the full reward r_i, so the format term f_i alone never makes a group appear to carry variance. We intervene on both kinds of zero-variance group, all correct and all incorrect. To test this choice, we also ablate the alternative that intervenes on all-incorrect groups alone. This alternative is intuitive, because exploration appears most necessary on the questions that the model fails to solve, yet it performs worse, as Section 5.3 reports.

Top-token mask. When the first wave is zero-variance, the second wave samples differently. Let $y_t^\star$ denote the token the policy finds most likely at position t and $p_t^\star$ its probability. Wherever $p_t^\star$ exceeds τ and the position lies inside the reasoning span, between the think tags, we drop that token and share its probability over the rest:

$$\tilde{\pi}_\theta(y_t^\star) = 0, \qquad \tilde{\pi}_\theta(y) = \frac{\pi_\theta(y \mid q, o_{i,<t})}{1 - p_t^\star} \qquad (3)$$

for every other token y.

Here π_θ is the policy we train, q is the prompt, y is a candidate token at position t, and $o_{i,<t}$ is the part of rollout i that the policy has already produced. Everywhere else the second wave samples from π_θ unchanged. Because we renormalize the remaining mass, the sampler draws from the policy's own alternatives in proportion to their probability, so the rollout continues from an alternative token that the policy samples rarely at that position. We set $\tau = 0.95$ (Section 5.3 ablates τ). We name the method after this gate, because the second wave explores exactly at the positions where the policy is most confident. Because the mask acts only inside the reasoning span, it never changes the answer tokens that the reward reads. The think tags that bound that span lie outside it, so the mask never removes them. Most token-level exploration methods trigger on the entropy or the surprisal of the next token, or on a statistic that also needs the advantage [13, 42, 43, 59]. We follow this direction, but we use $p_t^\star$ as the trigger instead, because the mask acts on the top-1 token, so $p_t^\star$ measures exactly the confidence that the mask removes. A fixed top-1 probability still allows many different entropy values, so a trigger on $p_t^\star$ and a trigger on the entropy select different positions.

Forced tokens in the loss. We keep every position of every rollout in the sum of Equation 2, including the masked positions, which are the positions where the mask forces a token other than the top one. The advantage $\hat{A}_i$ still covers all G rollouts of both waves together, so each rollout of the second wave contributes in the same way as a rollout of the first wave. A forced token is off-policy, and the importance ratio weights it without a special rule. Because the sampling engine scores the token under the masked and renormalized distribution, the ratio in Equation 2 falls to $1 - p_t^\star \le 0.05$ at the first update:

$$\rho_{i,t} = \frac{\pi_\theta(o_{i,t} \mid \cdot)}{\tilde{\pi}_\theta(o_{i,t} \mid \cdot)} = 1 - p_t^\star \;\le\; 1 - \tau = 0.05. \qquad (4)$$

This value is far below the lower clip bound, so the clipped objective treats the forced token according to the sign of its advantage. For a positive advantage, it keeps the token at a small weight, so an alternative that leads to a better outcome gains a little probability. For a negative advantage, the clipped term is constant, so the token receives no gradient. If the second wave changes the outcome of a zero-variance group, every rollout in the group now receives a gradient. Otherwise, the group stays zero-variance and costs nothing more than before.

Cost. CGE adds only one barrier per prompt, since the second wave cannot start before we score the first wave. In our asynchronous trainer, however, we measure no significant change in throughput, as Appendix K of the supplementary reports.

5. Experiments

5.1. Setup

Training data. The *general* dataset is a 105,993-row mixture of public multiple-choice video question answering data that prior video reasoning work assembled. Specifically, it draws on LLaVA-Video [82] (72,421 rows), STAR [65] (11,455), CLEVRER [71] (8,220), NExT-QA [67] (7,549) and PerceptionTest [47] (6,348). The *multi-hop* dataset, in contrast, is Video-HopChain with 22,550 rows. We release Video-HopChain[1], our model Video-HopChain-8B[2], the standard-RL checkpoint[3], and the code that generates the dataset and trains the model[4].

Model and training. On both datasets, every run starts from Qwen3-VL-8B-Instruct [4]. From this base model, we train on 4 nodes of 8 H100 80GB GPUs, split as 2 rollout nodes and 2 trainer nodes, with a fully asynchronous GRPO trainer built on verl [52]. For the video input, we sample frames evenly over the whole video, namely 140 frames for

[1] https://huggingface.co/datasets/ngqtrung/Video-HopChain

[2] https://huggingface.co/ngqtrung/Video-HopChain-8B

[3] https://huggingface.co/ngqtrung/Video-HopChain-8B-Standard-RL

[4] https://github.com/ngquangtrung57/video-hopchain

model	VMME	PComp	VMMMU	Holmes	VCR	MMR-V	LVR	VRB	mean	in-domain
Qwen3-VL-8B-Instruct	64.0	28.1	63.5	40.7	32.0	42.7	72.6	74.7	52.3	13.4
+ standard RL	65.8	34.3	63.0	47.4	35.9	43.4	76.1	77.6	55.4	13.4
+ standard RL + CGE	67.9	34.4	62.5	48.5	35.1	46.4	78.8	79.5	56.6	16.4
+ standard RL + Video-HopChain	68.6	36.2	64.7	**48.6**	42.1	45.9	77.8	79.4	57.9	21.2
+ standard RL + Video-HopChain + CGE	**69.2**	**37.4**	**67.5**	**48.6**	**44.9**	**46.6**	**78.9**	**81.0**	**59.3**	**23.2**

Table 1. Main results with the best value of each column in bold. The shaded row is our model, Video-HopChain-8B.

model	base	VMME	PComp	VMMMU	Holmes	VCR	MMR-V	LVR	VRB
base model									
Qwen3-VL-8B-Instruct†	–	64.0	28.1	63.5	40.7	32.0	42.7	72.6	74.7
open-source video reasoning models									
Video-R1 [17]	Qwen2.5-VL-7B	61.4	26.3	52.4	36.5	48.0	36.3	68.1	69.5
VideoChat-R1 [36]	Qwen2.5-VL-7B	60.0	28.6	46.4	33.0	48.2	36.1	67.2	61.5
VideoRFT [57]	Qwen2.5-VL-7B	59.8	–	51.1	–	–	–	–	–
Video-RTS [63]	Qwen2.5-VL-7B	63.0	–	52.7	40.7	–	–	–	–
Video-KTR [64]	Qwen2.5-VL-7B	62.5	–	53.1	42.7	–	–	–	–
Video-Thinker [58]	Qwen2.5-VL-7B	–	–	–	43.2	–	–	–	80.7
Video-o3 [75]	Qwen2.5-VL-7B	66.5	–	51.7	46.5	–	44.7	–	–
Conan [45]	Qwen2.5-VL-7B	–	–	–	44.6	**51.0**	42.7	72.8	**81.0**
LongVILA-R1 [10]	LongVILA-7B	65.1	–	51.0	–	–	–	72.0	–
OneThinker [18]	Qwen3-VL-8B	66.5	–	66.2	**48.7**	–	–	**79.2**	–
Video-HopChain-8B (ours)	Qwen3-VL-8B	**69.2**	**37.4**	**67.5**	48.6	44.9	**46.6**	78.9	**81.0**

Table 2. Comparison with open-source video reasoning models. A dash marks an unreported number, and the best value of each column is bold with the second best underlined. A dagger marks a result that we reproduce under our own setting.

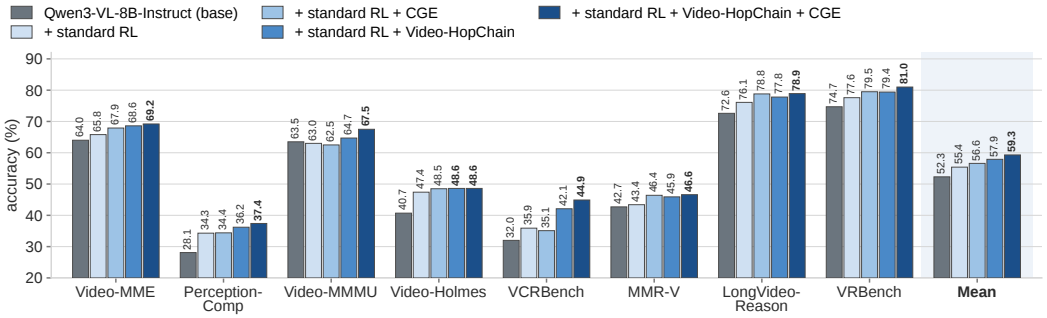


Figure 5. Main result. Accuracy on the eight public video benchmarks and their mean, for the base model and for our four training runs. The darkest bars are our model, Video-HopChain-8B.

Video-HopChain and 24 frames for the general dataset. For CGE, we also use $\tau = 0.95$ and a first wave of $G_1 = 4$ rollouts.

Training stages. We run the following stages.

- **First stage, on the general dataset.** We train the base model on the general dataset. We then keep the checkpoint at which its accuracy peaks. This checkpoint is the warm start for the longer and more complicated reasoning traces that Video-HopChain asks for. We call this run *standard RL*.
- **Second stage, on Video-HopChain with plain GRPO.** We start this run from the single checkpoint of the first stage, and we then train on Video-HopChain with plain GRPO.
- **Second stage, on Video-HopChain with CGE.** We start this run from the same checkpoint, but we train on Video-HopChain with CGE instead of plain GRPO. This run gives our model, Video-HopChain-8B.
- **CGE on the general dataset.** We start this run from the base model and enable the method from the first step.

Benchmarks. To measure these runs, we report eight public video benchmarks, namely Video-MME [20], PerceptionComp [35], Video-MMMU [24], Video-Holmes [12], VCRBench [50], MMR-V [88], LongVideo-Reason [10] and

run	τ	best mean
plain GRPO	—	66.8
CGE	0.80	65.0†
CGE	0.85	64.9†
CGE	0.90	**72.6**
CGE	0.95	72.5
CGE, all-incorrect only	0.95	66.7

Table 3. Ablations on OpenMMReasoner, as the best mean over the six image benchmarks of OpenMMReasoner. The last row intervenes on all-incorrect groups alone, whereas every other CGE run intervenes on all-correct groups as well. A dagger marks a run that never learns the answer format.

VRBench [72]. We report VCRBench on its multiple-choice subset. We evaluate every benchmark with the lmms-eval framework [77], under the same settings for every run, and we give the model the same system prompt that it sees during training. For every benchmark, we sample 100 frames evenly over the whole video, and we decode with at most 501,760 pixels per frame, a context of 33,792 tokens and at most 16,384 generated tokens. We report the accuracy on the 1,000 held-out Video-HopChain questions.

5.2. Main results

The dataset improves the model, and the method improves it further. On these eight public benchmarks, a second stage on Video-HopChain raises the mean and improves every one of them. Enabling CGE on top of that dataset raises the mean again. In addition, our model holds the best value of every column, where it ties the Video-HopChain run on Video-Holmes.

Our model outperforms open-source video reasoning models. Table 2 places our model beside ten open-source video reasoning models on the same eight benchmarks. It gives the best value on most of these benchmarks, ties with Conan on VRBench, and comes within 0.3 points of One-Thinker on Video-Holmes and on LongVideo-Reason. We compare released systems rather than recipes, because the ten models differ in base model, training data and compute.

5.3. Ablations

We ablate the two choices of Section 4.2, namely the threshold τ and the groups we intervene on. We run every one of them on the OpenMMReasoner RL dataset [79], a 74K-sample image dataset. CGE makes no assumption about the modality of the prompt and therefore applies to an image dataset as it does to video, and every run is a cold start from the same base model under the same reward and the same compute budget, so we change only the training dataset. We score every run on the six public image benchmarks that OpenMMReasoner evaluates on, namely CharXiv [62], DynaMath [89], MathVerse [80], MathVista [41], MMMU [74] and WeMath [48], and Table 3 reports the best mean over those six benchmarks.

The threshold needs no tuning above 0.90. At $\tau = 0.90$ and $\tau = 0.95$, CGE reaches 72.6 and 72.5, which is more than five points above plain GRPO at the same compute budget, and the two thresholds stay within 0.1 points of each other. At $\tau = 0.85$ and $\tau = 0.80$, by contrast, the model never learns to emit the answer format, the response length grows from 380 to over 13,000 tokens, and the fraction of clipped tokens rises from zero to 0.69. We read these two failures as one mechanism, because a lower threshold makes the mask eligible at positions of ordinary uncertainty rather than only where the policy is nearly certain, so the second wave fires at many more positions of the same rollout and the reasoning span keeps growing instead of closing. We therefore apply the mask only in the near-certain regime, under which the reasoning still converges.

Both kinds of zero-variance group need the intervention. A natural alternative to Section 4.2 is to intervene only when the first wave is all incorrect, because a group that the model already solves appears to need no exploration. However, most zero-variance groups are all correct, so this alternative lowers the share of intervened groups from 0.79 to 0.29, and the run then peaks at 66.7, which is 5.8 points below CGE at the same threshold and no better than plain GRPO. For this reason we intervene on all-correct and all-incorrect groups alike, and Appendix D of the supplementary reports both ablations in full.

6. Analysis

CGE recovers one zero-variance group in six. The first panel of Figure 6 tracks the share of the intervened groups whose second wave restores the variance of the group. This share grows over training, from about one group in seven to one in five. Almost every intervened group is a question the model cannot yet solve, because on Video-HopChain the first wave comes back all incorrect far more often than all correct. However, the mask changes the outcome of an all-correct group much more often than that of an all-incorrect one, so we attribute the rise to the growing share of all-correct groups. The restored variance also comes from new solutions rather than from noise, because the second wave solves questions that the first wave never solves, and it answers as accurately as the first wave does. When we count both waves, therefore, CGE gives about half again as many groups with a gradient at the same compute budget. Appendix G of the supplementary gives the full numbers.

CGE holds the policy entropy above plain GRPO. While the first panel counts groups, the third panel tracks the token entropy of the policy on its own rollouts, where CGE stays a few tenths of a nat above plain GRPO for the whole run, although both runs reach the same training reward. This margin matters, because reinforcement learning for reason-

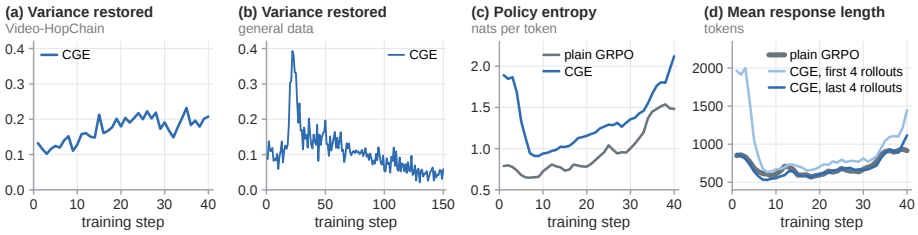


Figure 6. Four training metrics of our runs, over the training steps.

ing models tends to lose entropy as the reward rises, until the policy stops exploring and the reward saturates, a failure that prior work calls entropy collapse [13]. CGE keeps more room to explore. We attribute this margin to the masked positions and to the tokens that follow them, because one alternative token carries a rollout along a path the policy samples rarely.

The dataset and the method fit together. Because the mask acts only inside the reasoning span, the number of positions it can change grows with the length of that span, and a multi-hop question of Video-HopChain therefore offers it many more positions than a multiple-choice question does. On the general dataset, the second wave restores the variance of far fewer intervened groups than on Video-HopChain.

7. Conclusion

We presented Video-HopChain, a dataset of multi-hop questions over videos. A second stage of training on this dataset improves general video understanding and reasoning beyond what standard RL on general video data reaches. Because such training also produces zero-variance groups, we then introduced Confidence-Gated Exploration, which resamples the second half of a zero-variance group under a top-token mask. The method therefore recovers these groups without additional rollouts, and it improves the results further. We hope that both contributions help future work on video reasoning.

LLM Usage Statement

We used large language models in three parts of this work. First, they are part of the method that builds the dataset: MiniMax-M3 captions every shot of a video, and Qwen3.8-27B writes each question from those captions and then checks it (Section 3.2). Every Video-HopChain question is therefore synthetic text. Second, we used Claude (Anthropic), through the Claude Code agent, for implementation and coding and to create the figures. Third, we used the same tool to polish the writing of the paper. The authors designed the study, checked every output of these tools, and take full responsibility for the content of the paper.

References

[1] Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*, 2025. 3

[2] Emad Bahrami, Olga Zatsarynna, Parth Pathak, Sunando Sengupta, Juergen Gall, and Mohsen Fayyaz. STRIVE: Structured spatiotemporal exploration for reinforcement learning in video question answering. *arXiv preprint arXiv:2604.01824*, 2026. 3

[3] Bizhe Bai, Xinyue Wang, Peng Ye, and Tao Chen. Learning to explore with Parameter-Space noise: A deep dive into Parameter-Space noise for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2602.02555*, 2026. 3

[4] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. 6

[5] Vladislav Beliaev. Max out GRPO signal: Adaptive trace prefix control for hard reasoning problems. *arXiv preprint arXiv:2607.07674*, 2026. 3

[6] Brandon Castellano. PySceneDetect: Video scene cut detection and analysis tool. https://github.com/Breakthrough/PySceneDetect, 2026. Software, version 0.7. 4

[7] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, 2024. 17

[8] Liang Chen, Xueting Han, Qizhou Wang, Bo Han, Jing Bai, Hinrich Schütze, and Kam-Fai Wong. EEPO: Exploration-Enhanced policy optimization via Sample-Then-Forget. *arXiv preprint arXiv:2510.05837*, 2025. 3, 5

[9] Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. LongVILA: Scaling Long-Context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024. 4

[10] Yukang Chen, Wei Huang, Baifeng Shi, Qinghao Hu, Hanrong Ye, Ligeng Zhu, Zhijian Liu, Pavlo Molchanov, Jan Kautz, Xiaojuan Qi, Sifei Liu, Hongxu Yin, Yao Lu, and Song Han. Scaling RL to long videos. *arXiv preprint arXiv:2507.07966*, 2025. 2, 3, 7

[11] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025. 3

[12] Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. Video-Holmes: Can MLLM think like Holmes for complex video reasoning? *arXiv preprint arXiv:2505.21374*, 2025. 7

[13] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. 3, 6, 9

[14] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. 2, 5

[15] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed Deep-RL with importance weighted Actor-Learner architectures. *arXiv preprint arXiv:1802.01561*, 2018. 3

[16] Miquel Farré, Andi Marafioti, Lewis Tunstall, Leandro Von Werra, and Thomas Wolf. FineVideo. https://huggingface.co/datasets/HuggingFaceFV/finevideo, 2024. 4

[17] Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. *arXiv preprint arXiv:2503.21776*, 2025. 2, 3, 7

[18] Kaituo Feng, Manyuan Zhang, Hongyu Li, Kaixuan Fan, Shuang Chen, Yilei Jiang, Dian Zheng, Peiwen Sun, Yiyuan Zhang, Haoze Sun, Yan Feng, Peng Pei, Xunliang Cai, and Xiangyu Yue. OneThinker: All-in-one reasoning model for image and video. *arXiv preprint arXiv:2512.03043*, 2025. 2, 3, 4, 7

[19] Yunzhen Feng, Parag Jain, Anthony Hartshorn, Yaqi Duan, and Julia Kempe. Don't waste mistakes: Leveraging negative RL-Groups via confidence reweighting. *arXiv preprint arXiv:2510.08696*, 2025. 3

[20] 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. *arXiv preprint arXiv:2405.21075*, 2024. 7

[21] Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. AGQA: A benchmark for compositional Spatio-Temporal reasoning. *arXiv preprint arXiv:2103.16002*, 2021. 3

[22] Zhuowen Han, Jinwei Xiao, Zhengxi Lu, Renren Jin, Zhiyuan Yao, Yuxin Liu, Hongyan Hao, Yueqing Sun, Yu Yang, Qi GU, Xunliang Cai, and Deyi Xiong. Distill where you fail: Recovering learning signals of negative RL-Groups from adaptive teacher guidance. *arXiv preprint arXiv:2608.00782*, 2026. 3

[23] Xixiang He, Qiyao Sun, Ao Cheng, Xingming Li, Xuanyu Ji, Hailun Lu, Runke Huang, and Qingyong Hu. Advantage collapse in group relative policy optimization: Diagnosis and mitigation. *arXiv preprint arXiv:2605.21125*, 2026. 2, 3

[24] 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. 7

[25] Guanhua Huang, Tingqiang Xu, Mingze Wang, Qi Yi, Xue Gong, Siheng Li, Ruibin Xiong, Kejiao Li, Yuhao Jiang, and Bo Zhou. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. *arXiv preprint arXiv:2510.03222*, 2025. 3

[26] Langlin Huang, Chengsong Huang, Jinyuan Li, Donghong Cai, Yuyi Yang, and Jiaxin Huang. Nonsense helps: Prompt space perturbation broadens reasoning exploration. *arXiv preprint arXiv:2605.05566*, 2026. 3

[27] Yifu Huo, Chenglong Wang, Ziming Zhu, Shunjie Xing, Peinan Feng, Tongran Liu, Qiaozhi He, Tianhua Zhou, Xiaojia Chang, Jingbo Zhu, Zhengtao Yu, and Tong Xiao. SPS: Steering probability squeezing for better exploration in reinforcement learning for large language models. *arXiv preprint arXiv:2604.16995*, 2026. 3

[28] Anmol Kabra, Yilun Yin, Albert Gong, Kamilė Stankevičiūtė, Dongyoung Go, Johann Lee, Katie Z. Luo, Carla P. Gomes, and Kilian Q. Weinberger. Learning from synthetic data improves multi-hop reasoning. *arXiv preprint arXiv:2603.02091*, 2026. 3

[29] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Proceedings of the European Conference on Computer Vision*, 2016. 17

[30] Soeun Kim and Albert No. Diversifying RLVR rollouts via First-Token exploration. *arXiv preprint arXiv:2605.28295*, 2026. 3

[31] Benno Krojer, Mojtaba Komeili, Candace Ross, Quentin Garrido, Koustuv Sinha, Nicolas Ballas, and Mahmoud Assran. A shortcut-aware Video-QA benchmark for physical understanding via minimal video pairs. *arXiv preprint arXiv:2506.09987*, 2025. 2, 3

[32] Sutej Kulgod, Sean Ye, Sanchit Tanwar, and Christoffer Heckman. Reducing text bias in synthetically generated MCQAs for VLMs in autonomous driving. *arXiv preprint arXiv:2602.17677*, 2026. 3

[33] Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, Jinkai Hu, Jiayao Li, Rui Gao, Zekun Li, Songquan Zhu, Jingkai Zhou, and Pengyu Zhao. MiniMax sparse attention. *arXiv preprint arXiv:2606.13392*, 2026. 4

[34] Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. No prompt left behind: Exploiting Zero-Variance prompts in LLM reinforcement learning via Entropy-Guided advantage shaping. *arXiv preprint arXiv:2509.21880*, 2025. 2, 3, 5

[35] Shaoxuan Li, Zhixuan Zhao, Hanze Deng, Zirun Ma, Shulin Tian, Zuyan Liu, Yushi Hu, Haoning Wu, Yuhao Dong, Benlin Liu, Ziwei Liu, and Ranjay Krishna. PerceptionComp: A video benchmark for complex Perception-Centric reasoning. *arXiv preprint arXiv:2603.26653*, 2026. 7

[36] Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing Spatio-Temporal perception via reinforcement Fine-Tuning. *arXiv preprint arXiv:2504.06958*, 2025. 2, 3, 7

[37] Ziniu Li, Congliang Chen, Tianyun Yang, Tian Ding, Ruoyu Sun, Ge Zhang, Wenhao Huang, and Zhi-Quan Luo. Knapsack RL: Unlocking exploration of LLMs via optimizing budget allocation. *arXiv preprint arXiv:2509.25849*, 2025. 3

[38] Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. NoisyRollout: Reinforcing visual reasoning with data augmentation. *arXiv preprint arXiv:2504.13055*, 2025. 3

[39] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *Proceedings of the European Conference on Computer Vision*, 2024. 17

[40] Xiuyi Lou, Zicheng Xu, Yu-Neng Chuang, Hoang Anh Duy Le, Zhaozhuo Xu, Guanchu Wang, and Vladimir Braverman. When implausible tokens get reinforced: Tail-Aware credit calibration for LLM reinforcement learning. *arXiv preprint arXiv:2607.07976*, 2026. 3

[41] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, 2024. 8, 17

[42] Haipeng Luo, Qingfeng Sun, Songli Wu, Can Xu, Wenfeng Deng, Han Hu, and Yansong Tang. STARE: Surprisal-Guided Token-Level advantage reweighting for policy entropy stability. *arXiv preprint arXiv:2606.19236*, 2026. 3, 6

[43] Outongyi Lv, Yanzhao Zheng, Yuanwei Zhang, Zhenghao Huang, Xingjun Wang, Baohua Dong, Hangcheng Zhu, and Yingda Chen. Which tokens matter? adaptive token selection for RLVR with the relative surprisal index. *arXiv preprint arXiv:2606.31575*, 2026. 3, 6

[44] 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. 2, 3

[45] Kun Ouyang, Yuanxin Liu, Linli Yao, Yishuo Cai, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Conan: Progressive learning to reason like a detective over multi-scale visual evidence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 41089–41099, 2026. 7

[46] Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. Beyond the sampled token: Preserving candidate support in RLVR. *arXiv preprint arXiv:2510.14807*, 2025. 3

[47] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Per-

ception test: A diagnostic benchmark for multimodal video models. *arXiv preprint arXiv:2305.13786*, 2023. 6

[48] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, Yifan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. We-Math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070, 2025. 8, 17

[49] Trung Nguyen Quang, Yiming Gao, Fanyi Pu, Kaichen Zhang, Shuo Sun, and Ziwei Liu. Senses wide shut: A representation-action gap in omnimodal LLMs. *arXiv preprint arXiv:2605.13737*, 2026. 2, 3

[50] Pritam Sarkar and Ali Etemad. VCRBench: Exploring Long-form causal reasoning capabilities of large video language models. *arXiv preprint arXiv:2505.08455*, 2025. 7

[51] 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. 2, 5

[52] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024. 6

[53] Junyoung Sung, Seungwoo Lyu, Minjun Kim, Sumin An, Arsha Nagrani, and Paul Hongsuck Seo. CRIT: Graph-Based automatic data synthesis to enhance Cross-Modal Multi-Hop reasoning. *arXiv preprint arXiv:2604.01634*, 2026. 3

[54] Ling Team, Anqi Shen, Baihui Li, Bin Hu, Bin Jing, Cai Chen, Chao Huang, Chao Zhang, Chaokun Yang, Cheng Lin, Chengyao Wen, Congqi Li, Deng Zhao, Dingbo Yuan, Donghai You, Fagui Mao, Fanzhuang Meng, Feng Xu, Guojie Li, Guowei Wang, Hao Dai, Haonan Zheng, Hong Liu, Jia Guo, Jiaming Liu, Jian Liu, Jianhao Fu, Jiannan Shi, Jianwen Wang, Jianxin Lai, Jin Yang, Jun Mei, Jun Zhou, Junbo Zhao, Junping Zhao, Kuan Xu, Le Su, Lei Chen, Li Tang, Liang Jiang, Liangcheng Fu, Lianhao Xu, Linfeng Shi, Lisha Liao, Longfei Zheng, Meng Li, Mingchun Chen, Qi Zuo, Qiang Cheng, Qianggang Cao, Qitao Shi, Quanrui Guo, Senlin Zhu, Shaofei Wang, Shaomian Zheng, Shuaicheng Li, Shuwei Gu, Siba Chen, Tao Wu, Tao Zhang, Tianyu Zhang, Tianyu Zhou, Tiwei Bie, Tongkai Yang, Wang Hong, Wang Ren, Weihua Chen, Wenbo Yu, Wengang Zheng, Xiangchun Wang, Xiaodong Yan, Xiaopei Wan, Xin Zhao, Xinyu Kong, Xinyu Tang, Xudong Han, Xudong Wang, Xuemin Yang, Xueyu Hu, Yalin Zhang, Yan Sun, Yicheng Shan, Yilong Wang, Yingying Xu, Yongkang Liu, Yongzhen Guo, Yuanyuan Wang, Yuchen Yan, Yuefan Wang, Yuhong Guo, Zehuan Li, Zhankai Xu, Zhe Li, Zhenduo Zhang, Zhengke Gui, Zhenxuan Pan, Zhenyu Huang, Zhenzhong Lan, Zhiqiang Ding, Zhiqiang Zhang, Zhixun Li, Zhizhen Liu, Zihao Wang, and Zujie Wen. Every step evolves: Scaling reinforcement learning for Trillion-Scale thinking model. *arXiv preprint arXiv:2510.18855*, 2025. 3

[55] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *arXiv preprint arXiv:2108.00573*, 2021. 3

[56] Hongcheng Wang, Yinuo Huang, Sukai Wang, Guanghui Ren, and Hao Dong. Why Tree-Style branching matters for thought advantage estimation in GRPO. *arXiv preprint arXiv:2509.24494*, 2025. 3

[57] Qi Wang, Yanrui Yu, Ye Yuan, Rui Mao, and Tianfei Zhou. VideoRFT: Incentivizing video reasoning capability in MLLMs via reinforced Fine-Tuning. *arXiv preprint arXiv:2505.12434*, 2025. 2, 3, 7

[58] Shijian Wang, Jiarui Jin, Xingjian Wang, Linxin Song, Runhao Fu, Hecheng Wang, Zongyuan Ge, Yuan Lu, and Xuelian Cheng. Video-Thinker: Sparking "thinking with videos" via reinforcement learning. *arXiv preprint arXiv:2510.23473*, 2025. 3, 7

[59] 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. 3, 6

[60] Shenzhi Wang, Shixuan Liu, Jing Zhou, Chang Gao, Xiong-Hui Chen, Binghai Wang, An Yang, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. HopChain: Multi-Hop data synthesis for generalizable Vision-Language reasoning. *arXiv preprint arXiv:2603.17024*, 2026. 2, 3

[61] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. SoTA with less: MCTS-Guided sample selection for Data-Efficient visual reasoning Self-Improvement. *arXiv preprint arXiv:2504.07934*, 2025. 2

[62] Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. In *Advances in Neural Information Processing Systems*, pages 113569–113697, 2024. 8, 17

[63] Ziyang Wang, Jaehong Yoon, Shoubin Yu, Md Mohaiminul Islam, Gedas Bertasius, and Mohit Bansal. Video-RTS: Rethinking reinforcement learning and Test-Time scaling for efficient and enhanced video reasoning. *arXiv preprint arXiv:2507.06485*, 2025. 7

[64] Ziyue Wang, Sheng Jin, Zhongrong Zuo, Jiawei Wu, Han Qiu, Qi She, Hao Zhang, and Xudong Jiang. Video-KTR: Reinforcing video reasoning via key token attribution. *arXiv preprint arXiv:2601.19686*, 2026. 3, 7

[65] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. STAR: A benchmark for situated reasoning in Real-World videos. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021. 6

[66] xAI. RealWorldQA. Dataset release accompanying Grok-1.5 Vision, 2024. 17

[67] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of Question-Answering to explaining

temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 6

[68] Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. Reinforce-Ada: An adaptive sampling framework under non-linear RL objectives. *arXiv preprint arXiv:2510.04996*, 2025. 3

[69] Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under Off-Policy guidance. *arXiv preprint arXiv:2504.14945*, 2025. 3

[70] Dongjie Yang, Suyuan Huang, Chengqiang Lu, Xiaodong Han, Haoxin Zhang, Yan Gao, Yao Hu, and Hai Zhao. Vript: A video is worth thousands of words. *arXiv preprint arXiv:2406.06040*, 2024. 4

[71] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: CoLlision events for video REpresentation and reasoning. In *International Conference on Learning Representations*, 2020. 6

[72] Jiashuo Yu, Yue Wu, Meng Chu, Zhifei Ren, Zizheng Huang, Pei Chu, Ruijie Zhang, Yinan He, Qirui Li, Songze Li, Zhenxiang Li, Zhongying Tu, Conghui He, Yu Qiao, Yali Wang, Yi Wang, and Limin Wang. VRBench: A benchmark for Multi-Step reasoning in long narrative videos. *arXiv preprint arXiv:2506.10857*, 2025. 3, 8

[73] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. 2, 3, 5

[74] 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*, 2024. 8, 17

[75] Xiangyu Zeng, Zhiqiu Zhang, Yuhan Zhu, Xinhao Li, Zikang Wang, Changlian Ma, Qingyu Zhang, Zizheng Huang, Kun Ouyang, Tianxiang Jiang, Ziang Yan, Yi Wang, Hongjie Zhang, Yali Wang, and Limin Wang. Video-o3: Native interleaved clue seeking for long video Multi-Hop reasoning. *arXiv preprint arXiv:2601.23224*, 2026. 7

[76] Congzhi Zhang, Zhibin Wang, Yinchao Ma, Jiawei Peng, Yihan Wang, Qiang Zhou, Jun Song, and Bo Zheng. ReWatch-R1: Boosting complex video reasoning in large Vision-Language models through agentic data synthesis. *arXiv preprint arXiv:2509.23652*, 2025. 3

[77] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*, 2024. 8

[78] Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. GVPO: Group variance policy optimization for large language model Post-Training. *arXiv preprint arXiv:2504.19599*, 2025. 3

[79] Kaichen Zhang, Keming Wu, Zuhao Yang, Bo Li, Kairui Hu, Bin Wang, Ziwei Liu, Xingxuan Li, and Lidong Bing. OpenMMReasoner: Pushing the frontiers for multimodal reasoning with an open and general recipe. *arXiv preprint arXiv:2511.16334*, 2025. 2, 8, 16

[80] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? In *Proceedings of the European Conference on Computer Vision*, pages 169–186. Springer, 2024. 8, 17

[81] Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. EDGE-GRPO: Entropy-Driven GRPO with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025. 2, 3

[82] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024. 6

[83] Yuxuan Zhang, EunJeong Hwang, Huaisong Zhang, Penghui Du, Yiming Jia, Dongfu Jiang, Xuan He, Shenhui Zhang, Ping Nie, Peter West, and Kelsey R. Allen. Watch before you answer: Learning from visually grounded Post-Training. *arXiv preprint arXiv:2604.05117*, 2026. 2, 3

[84] Yuheng Zhang, Chenlu Ye, Shuowei Jin, Changlong Yu, Wei Xiong, Saurabh Sahu, and Nan Jiang. Rethinking importance sampling in LLM policy optimization: A cumulative token perspective. *arXiv preprint arXiv:2605.07331*, 2026. 3

[85] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025. 3

[86] Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, Qian Liu, Ge Zhang, and Zejun Ma. First return, Entropy-Eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025. 3

[87] Tianle Zhong, Neiwen Ling, Yifan Pi, Zijun Wei, Tianshu Yu, Geoffrey Fox, Peng Wu, and Xiao Yu. Diagnosing training inference mismatch in LLM reinforcement learning via a zero-mismatch reference. *arXiv preprint arXiv:2605.14220*, 2026. 3

[88] Kejian Zhu, Zhuoran Jin, Hongbang Yuan, Jiachun Li, Shangqing Tu, Pengfei Cao, Yubo Chen, Kang Liu, and Jun Zhao. MMR-V: What's left unsaid? A benchmark for multimodal deep reasoning in videos. *arXiv preprint arXiv:2506.04141*, 2025. 7

[89] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. DynaMath: A dynamic visual benchmark

for evaluating mathematical reasoning robustness of vision language models. In *International Conference on Learning Representations*, 2025. 8, 17

Video-HopChain: Multi-Hop Questions and Confidence-Gated Exploration for Video Reasoning Models

Supplementary Material

Appendix contents

- **Appendix A, Algorithm.** CGE for one prompt, step by step.
- **Appendix B, Limitations.** What this study does not show.
- **Appendix C, Configuration.** Every hyperparameter of the two runs.
- **Appendix D, Ablations.** The threshold τ and the groups we intervene on, measured on image data.
- **Appendix E, Image benchmark results.** The effect of this training on general image ability.
- **Appendix F, Dataset details.** Statistics, row format, system prompt and the held-out split.
- **Appendix G, Group-level analysis of the second wave.** Which groups the second wave returns a gradient to.
- **Appendix H, Token-level analysis of the mask.** Where the mask acts and what it removes.
- **Appendix I, Derivations.** The entropy bounds and the importance ratio at a masked position.
- **Appendix J, Effect of the second wave on the update.** What the second wave changes in the update.
- **Appendix K, Computational cost and response length.** The cost of the second wave, and its effect on response length.

A. Algorithm

Algorithm 1 Confidence-Gated Exploration for one prompt

Require: prompt q, policy π_θ, group size G, threshold τ

sample $o_1, \dots, o_{G/2} \sim \pi_\theta(\cdot \mid q)$ and score $r_1, \dots, r_{G/2}$ with accuracies $a_1, \dots, a_{G/2}$ ▷ first wave

$g \leftarrow 1$ if $a_1 = \cdots = a_{G/2}$, else 0 ▷ is the first wave all correct or all incorrect?

for $i = G/2 + 1, \dots, G$ **do** ▷ second wave

  for $t = 1, 2, \dots$ until end of response **do**

    $p_t^\star \leftarrow \max_y \pi_\theta(y \mid q, o_{i,<t})$

    if $g = 1$ and $p_t^\star > \tau$ and t is inside the reasoning span **then**

      sample $o_{i,t} \sim \tilde{\pi}_\theta(\cdot \mid q, o_{i,<t})$ ▷ top-token mask, Eq. 3

    else

      sample $o_{i,t} \sim \pi_\theta(\cdot \mid q, o_{i,<t})$

    end if

  end for

  score r_i

end for

compute $\hat{A}_1, \dots, \hat{A}_G$ over all G responses ▷ standardized within the group

return $\{(o_i, \hat{A}_i)\}_{i=1}^{G}$ ▷ every position enters the loss

B. Limitations

We train one base model at one scale, so we do not test whether our results hold at other scales. In addition, we generate and verify Video-HopChain automatically, and although this dataset improves both the in-domain measure and the public benchmarks, a manually labelled and manually verified dataset could give different results. Finally, we do not run a few ablations, namely the first-wave size G_1, the restriction of the mask to the reasoning span, and an adaptive τ.

C. Configuration

Table 4 lists the settings of the two video runs, read from their resolved configurations. In this table, the default column reports the plain GRPO run, whereas the CGE column reports the run that enables the mask, and these two runs differ only in the exploration settings.

D. Ablations

Scope of the ablations. We ablate two choices of Section 4.2, namely the threshold τ and the groups we intervene

setting	default	CGE
data		
maximum prompt length	5,376	5,376
maximum response length	16,384	16,384
optimizer		
learning rate	1×10^{-6}	1×10^{-6}
warmup steps	25	25
weight decay	0.1	0.1
PPO mini-batch size	16	16
micro-batch size per GPU	dynamic	dynamic
maximum tokens per actor micro-batch	26,004	26,004
PPO epochs per batch	1	1
objective		
advantage estimator	GRPO	GRPO
advantage normalized by group std.	yes	yes
clip range, low (Eq. 2)	0.2	0.2
clip range, high	0.3	0.3
KL loss	off	off
KL coefficient	0	0
KL term in the reward	off	off
entropy coefficient	0	0
loss aggregation	token-mean	token-mean
old log-probabilities (Eq. 4)	from the engine	from the engine
cross-engine correction	none	none
rollout		
group size G	8	8
sampling temperature	1.0	1.0
top-p	1.0	1.0
top-k	-1, off	-1, off
tensor parallel size	2	2
GPU memory fraction	0.8	0.8
exploration		
exploration enabled	no	yes
second wave enabled	no	yes
first-wave fraction	—	0.5, $G_1 = 4$
first-wave reward variance	—	0
reward the check uses	—	accuracy
intervene on all-correct groups too	—	yes
trigger mode	—	high
mask threshold τ (Eq. 3)	—	0.95
tokens masked per position	—	1
positions eligible	—	reasoning span
cap on masks per response	—	16,384
forced tokens kept in the loss	—	yes
schedule and hardware		
epochs over the dataset	2	2
trainer nodes $\times$ GPUs	2×8	2×8
rollout nodes $\times$ GPUs	2×8	2×8
staleness threshold	0.5	0.5
parameter-sync interval	4 steps	4 steps

Table 4. Configuration of the default run and the CGE run.

on. We run every one of them on the OpenMMReasoner RL dataset [79], a 74K-sample image dataset, and we hold

everything else fixed, namely the same base model, the same algorithm, the same reward and the same compute budget, so we change only the training dataset. We run them on image data because an image run logs many more validation points than a video run of the same wall-clock budget, so it locates the best accuracy of a run far more precisely. We therefore treat these ablations as evidence about CGE itself rather than as measurements of video accuracy.

Setting. On that image dataset, every run here measures CGE alone, because CGE makes no assumption about the modality of the prompt and therefore applies to an image dataset as it does to video. Every run is a cold start from the base model on the same data, and we then score it on the six public image benchmarks that OpenMMReasoner evaluates on, namely CharXiv [62], DynaMath [89], MathVerse [80], MathVista [41], MMMU [74] and WeMath [48]. Every mean we report on this dataset is the mean over those six benchmarks.

Threshold. Figure 7 gives the best mean accuracy of the threshold sweep, together with plain GRPO as the run without any exploration. At 0.90 and 0.95 both runs learn the answer format and reach more than five points above plain GRPO at the same compute budget. Moreover, the two thresholds stay within 0.1 points of each other, so the threshold needs no tuning inside that range. At 0.85 and 0.80, by contrast, the model never learns to emit the answer format, the response length grows from 380 to over 13,000 tokens, the fraction of clipped tokens rises from zero to 0.69, and the step time grows thirteenfold. The accuracy at $\tau = 0.85$ rises again late in that run, however, and we attribute this rise to length inflation rather than to recovery.

We read these two failures as one mechanism. A lower threshold makes the mask eligible at positions of ordinary uncertainty, not only where the policy is nearly certain, so the second wave fires at many more positions of the same rollout. The model then never settles on a line of reasoning, because at every step where it starts to commit, the mask moves it off the token it was about to write. The reasoning span therefore keeps growing instead of closing, which is what the rise from 380 to over 13,000 tokens measures. The response finally meets the token budget rather than an end of its own, and the fraction of clipped tokens of 0.69 is that collision. A clipped response stops inside the reasoning span, so it never reaches the answer that the format asks for, and the format rate falls with it. Moreover, the mask applies only inside the reasoning span, between the think tags, as Section 4.2 defines it, so a token of the answer block is never eligible for the mask at any threshold. The format degrades because the reasoning never ends, not because the mask deletes the format. We therefore apply the mask only where the policy is nearly certain, since that is the setting under which the reasoning still converges.

Choice of the intervened groups. In Section 4.2, we intervene on all-correct and all-incorrect groups alike. A natural alternative is to intervene only when the first wave is all-incorrect, because a group that the model already solves appears to need no exploration. However, Table 5 shows that this alternative fails. Because most zero-variance groups are all correct, a rule that skips them reduces the share of intervened groups from 0.79 to 0.29, and the run then peaks at 66.7, which is 5.8 points below CGE and no better than plain GRPO, before it collapses. For this reason we intervene on all-correct and all-incorrect groups alike.

E. Image benchmark results

Because every training stage of this paper uses video, we check whether this training lowers general image ability. We therefore evaluate the same checkpoints of Table 1 on six public image benchmarks, namely AI2D [29], MathVista [41], MMBench-EN [39], MMMU [74], MMStar [7] and RealWorldQA [66], for a total of 11,527 questions. We take the multiple-choice split of AI2D, the testmini split of MathVista, the English dev split of MMBench and the validation split of MMMU, and on MMMU we keep the multiple-choice questions only. We score every benchmark by exact match on a multiple-choice letter or on a numeric answer, so we use no model as a judge. Because we otherwise follow the settings of Section 5.1, we measure these image numbers in the same way as the video numbers of Table 1. Table 6 then reports the result.

F. Dataset details

This appendix gives the statistics of the dataset, the row format of the released dataset, the system prompt, and how we draw the held-out split.

Dataset statistics. Table 7 reports the size of the two splits, the hop counts and the range of the ground-truth answer, which is the integer that the hops of a question sum to.

Row format. Each training row of Video-HopChain holds a two-message prompt, namely the system prompt of the dataset, which the released code carries, and a user turn that starts with the video placeholder and continues with the question text. The row also carries the video path with its decode contract, the ground-truth answer as a string, and an extra-information record with the answer, the hop count, the hop types, and the question and video identifiers. We store the hop records, with their moment references, mappings, values and caption quotes, beside the dataset for auditing, but we keep them out of the training rows.

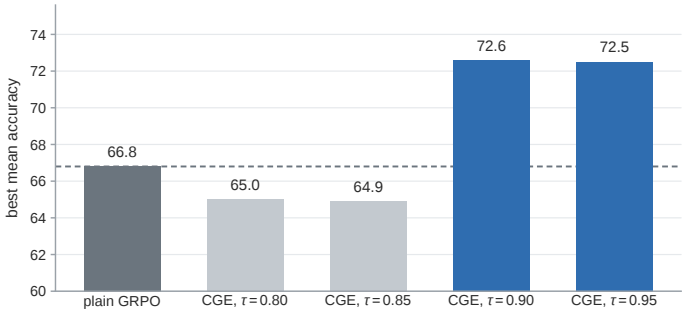


Figure 7. Best mean accuracy over the six image benchmarks of OpenMMReasoner, for plain GRPO and for CGE at four thresholds. The dashed line marks plain GRPO, and the two grey bars are the runs that never learn the answer format.

we intervene on	τ	best mean accuracy (step)	vs. CGE
all-correct and all-incorrect	0.95	72.5 (180)	—
all-incorrect only	0.95	66.7 (50)	−5.8

Table 5. Effect of the choice of intervened groups, measured on OpenMMReasoner.

model	AI2D 3,088	MathVista 998	MMBench 4,329	MMMU 847	MMStar 1,500	RealWorldQA 765	mean
Qwen3-VL-8B-Instruct	79.5	**75.2**	88.3	61.6	<u>67.5</u>	63.3	72.6
+ standard RL	82.4	71.2	89.3	62.3	**67.7**	65.0	73.0
+ standard RL + CGE	**83.3**	70.5	**89.6**	<u>62.8</u>	66.3	**69.8**	<u>73.7</u>
+ standard RL + Video-HopChain	82.7	71.6	<u>89.4</u>	62.2	66.8	66.3	73.2
+ standard RL + Video-HopChain + CGE	<u>82.8</u>	<u>73.0</u>	89.2	**62.9**	<u>67.5</u>	<u>67.3</u>	**73.8**

Table 6. Image understanding, accuracy in percent, with the number of questions under each benchmark, the best value of each column in bold and the second best underlined. The shaded row is our model, Video-HopChain-8B.

	train	held-out
questions	22,550	1,000
videos	13,378	1,000
questions per video	1.69	1.00
questions with 3 / 4 / 5 / 6 hops	595 / 12,604 / 7,527 / 1,824	0 / 302 / 450 / 248
hops by type: order / spatial / action / attribute	44,416 / 22,476 / 21,372 / 12,516	2,082 / 1,141 / 1,072 / 651
ground-truth answer, smallest to largest	20 to 436	52 to 366
ground-truth answer, median	178	198

Table 7. Video-HopChain statistics for the training split and the held-out split.

System prompt. Every training row and every held-out row carries this system prompt, so we evaluate every model we train with the same prompt.

```
You are a careful reasoning assistant. ALWAYS respond in
this EXACT format:

<think>step-by-step reasoning</think>
<answer>\boxed{final_answer}</answer>

Examples:

Q: 7 x 8?
<think>7 x 8 = 56.</think>
<answer>\boxed{56}</answer>

Q: A right triangle has legs of length 3 and 4. What is the
hypotenuse?
<think>By the Pythagorean theorem, c^2 = 3^2 + 4^2 = 9 + 16
= 25, so c = 5.</think>
<answer>\boxed{5}</answer>

For multiple-choice, put the letter, e.g. \boxed{B}.
Always wrap reasoning in <think>...</think> and answer in
<answer>\boxed{...}</answer>. No text outside these tags.
```

The question of Figure 2. The dataset row of this question, held-out question `2Wa_OoZyEk8::1`, carries the full text below, without the video placeholder.

```
Work through five checks and add the five numbers.

First, in the scene where a cat sits at a circular table
wearing a red harness with silver buckles, look at the
position of the golden Christmas tree: if the tree is on the
left side of the frame, let A be 63; otherwise let A be 10.
```

```
Second, compare two moments: the moment a spherical firework
explosion fills the frame, and the moment an outdoor scene
shows a tall green pole supporting a cluster of traffic
lights. If the firework explosion is shown before the
traffic light scene, let B be 22; otherwise let B be 63.

Third, use the first check to choose where to look: if the
tree is on the left side of the frame, look at the room
where two children stand beneath an upside-down Christmas
tree; otherwise look at the scene where a dog and a grey cat
rest on a flat surface with the cat holding one paw raised
near its face. In whichever of those two places you were
sent to, look at the surface beneath the characters: if the
surface is green, let C be 64; otherwise let C be 41.

Fourth, use the third check to choose which pair of moments
to compare: if the surface is green, compare the moment the
dog's face fills the frame in close-up with out-of-focus
colored lights on the left against the moment the camera
zooms in until the grey cat's face fills most of the frame;
if the surface is not green, compare the moment the dog's
head is turned toward the cat with a teal banner at the
bottom of the frame against the moment a small rodent in a
Santa hat is shown with fireworks bursting around it. In
whichever pair you were sent to, if the first-named moment
is shown before the second-named moment, let D be 73;
otherwise let D be 79.

Fifth, in the news desk scene where red illuminated elements
are visible below the desk's cyan rim, look at what the dog
is doing: if the dog is turning its head toward the other
character, let E be 70; if the dog is looking down at its
paper, let E be 42.

Add A, B, C, D, and E.
```

Held-out split. We split the dataset by video with seed 42, and we then draw the held-out split with seed 1234 from the questions with at least four hops and at least two distinct hop types, with exactly one question per video, and we stratify this split on three axes at once. By hop count it holds 302, 450 and 248 questions with 4, 5 and 6 hops. By video length it holds 151 questions on videos under 40 segments, 301 on 40 to 70 segments, 300 on 70 to 110 segments, and 248 on longer videos. By source it holds 467 questions from LongVILA, 358 from FineVideo and 175 from Vript.

G. Group-level analysis of the second wave

Table 8 gives the numbers behind the observations of Section 6, and we measure every row on the run of our model in Table 1, over the same window that Figure 6 plots. We report the mean over the first half of that window, which we label early, and the mean over the second half, which we label later.

The variance the second wave adds. Table 8 counts the groups whose variance the second wave restores, and we also measure how much variance it adds. For a group of G rollouts whose first wave answers a share p of them correctly, an unperturbed second wave would leave the group with a reward variance of $p(1-p)(G-1)/G$ on average, so we subtract that quantity from the variance we measure after the second wave and report the difference, which the middle panel of Figure 8 plots. On an intervened group the first wave is all correct or all incorrect, hence p is 0 or 1, the quantity we subtract is exactly zero, and the whole of the remaining variance follows from the mask. This difference grows from 0.020 early in the run to 0.028 later, whereas on the groups that the mask leaves alone the same difference reaches only 0.005 early and 0.009 later, so the subtraction leaves almost no variance on those groups.

The accuracy of each wave. The right panel of Figure 8 tracks the accuracy of each wave. Over all groups the two waves reach the same accuracy for the whole run, at 0.17 for the first wave and 0.16 for the second, whereas over the intervened groups alone the second wave is more accurate than the first by 0.02 early and by 0.01 later, as the accuracy row of Table 8 reports. We measure the same on the general dataset, where the first wave reaches 0.81 and the second reaches 0.80 over the whole run. Therefore the second wave answers as accurately as the first, and it answers more accurately on the groups the mask acts on, while it follows a path that the policy would otherwise not have taken.

The length of a response. The mask could also change the length of a rollout, and the fourth panel of Figure 6 therefore tracks the mean response length of the two runs on Video-HopChain. Both of them settle between 600 and 900 tokens once the format is stable, and both of them lengthen again late in training. CGE gives the longer responses, at 810 tokens against 708 for plain GRPO. The mask itself is not the source, however, because the second wave that carries it stays shorter than the first, at 696 tokens against 923. We therefore attribute the extra length to the policy that CGE trains.

H. Token-level analysis of the mask

The section above measures the second wave over a whole group, and we now measure it at the level of the token. The top-1 probability exceeds τ at 2.2% of the positions the mask examines, as the left panel of Figure 8 shows, and no response reaches the cap on masks of Table 4. To see which positions those are, we record every token that the mask removes together with the token that the sampler draws in its place, over the CGE run of Table 1. Each of the 8 sampling engines writes the first 20,000 masked positions it produces, which gives 160,000 positions in total, and the sampler resolves every one of them to a replacement. Figure 9 shows the two sides of that substitution, and Table 10 reports where those positions sit and which token the sampler draws in their place.

The two panels of Figure 9, however, do not carry the same kind of word, because the mask removes mostly the answer letters A to E, `boxed` and `answer`, and frequent function words such as `to` and `the`. The mask therefore acts

share of	early	later
what the first wave produces, over all groups		
groups whose 4 rollouts are all incorrect	0.73	0.64
groups whose 4 rollouts are all correct	0.03	0.08
groups that already carry variance	0.24	0.28
what the second wave adds, over the intervened groups		
groups whose variance the second wave restores	0.15	0.20
all-correct groups whose outcome the mask changes	0.70	0.61
all-incorrect groups in which the mask produces a correct rollout	0.13	0.15
groups in which the second wave solves what the first wave never solves	0.12	0.13
accuracy of the second wave minus the first	+0.02	+0.01
what the group contributes to the update		
groups that carry a gradient, first wave alone	0.24	0.28
groups that carry a gradient, both waves together	0.36	0.42
questions solved by at least one of the 8 rollouts	0.37	0.45
questions solved by at least one of the first 4	0.28	0.36

Table 8. The signal that the second wave recovers, measured on Video-HopChain.

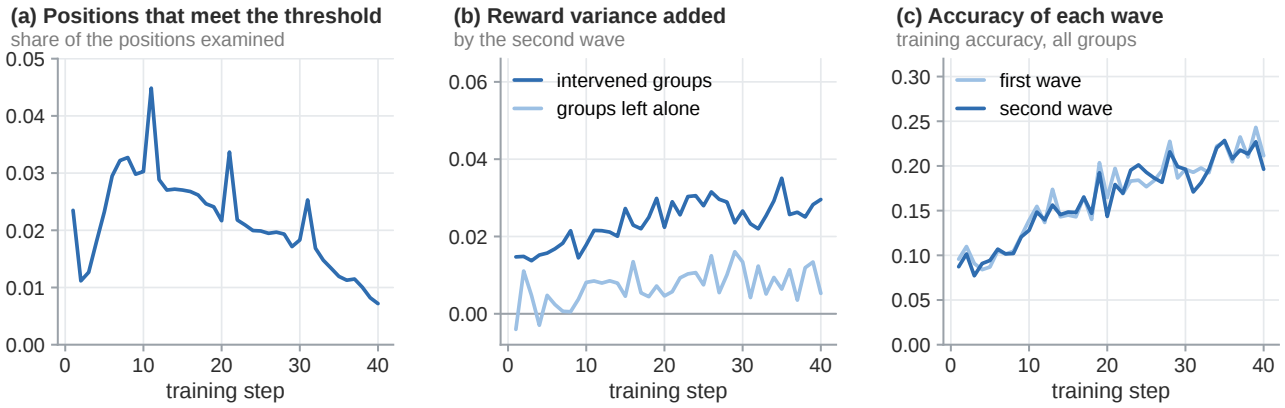


Figure 8. Three measurements of the second wave on Video-HopChain, over the training steps. The left panel gives the share of the positions the mask examines whose top-1 probability exceeds τ, whereas the middle panel gives the reward variance the second wave adds to a group, both on the intervened groups and on the groups the mask leaves alone, which already carry variance in their first wave. The right panel then gives the training accuracy of each wave over all groups.



Figure 9. The tokens that the mask removes, and the tokens that the sampler draws in their place, over the masked positions of the CGE run. Size follows frequency. Both panels fold a token onto its printed form, so a token and its word-boundary variant appear once, and they omit byte pieces that print no full character.

most often at the moment the model commits to an answer inside its reasoning span, which is where a near-certain token sits. The tokens `boxed` and `answer` appear in this list because the model drafts its answer inside the reasoning span before it closes that span. The mask therefore reaches that draft rather than the answer the reward reads, which lies

after the closing think tag and outside the span. The mask never removes the think tags either.

The substitution also reaches the content of a hop, as Table 9 shows. The upper block of that table changes the value a hop reads, because the sampler names a different answer letter, one side of the frame for the other, one vertical half for the other, the opposite direction of an order relation, a different colour, and a different part of a person. These are the kinds of hop that a Video-HopChain question asks about, so the mask reaches the content on which the answer depends and not only the wording that surrounds it. The lower block changes the direction the rest of the response takes, because the sampler alters the relation between two objects, commits to a side that the model had not yet named, weakens the claim the model was about to make, and renames the step of the chain that the model is working on. We therefore treat the second wave as exploration of the answer rather than as noise, and Appendix G supports this at the level of the group, because the mask produces a correct rollout in 0.13 of the all-incorrect groups and gives 0.12 of the groups a solution that the first wave never finds.

Three rows of that table matter for the method. First, the mask acts across the whole reasoning span rather than at its opening, because the median masked position sits at token 242 and about one masked position in three sits beyond token 400. Second, the sampler usually draws the token that the policy itself ranks second, which happens at half of the masked positions, and it draws one of the first three candidates at about two thirds of them, so the second wave follows a continuation that the policy already ranks highly rather than an arbitrary token. Third, the second most likely token holds more than half of the probability that the mask leaves behind at 46% of the positions, hence a masked position is most often a choice between the token the policy would commit to and one alternative.

I. Derivations

Entropy at a fixed top-1 probability. Let V be the vocabulary size and $p^\star$ the probability of the most likely token. The entropy of the distribution is smallest when one token holds all the remaining mass, whereas it is largest when the mass is spread evenly over the other $V-1$ tokens, which gives two bounds:

$$\begin{aligned} H_{\min}(p^\star) &= -p^\star \log p^\star - (1-p^\star)\log(1-p^\star), \\ H_{\max}(p^\star) &= -p^\star \log p^\star \\ &\quad + (1-p^\star)\big(\log(V-1) - \log(1-p^\star)\big). \end{aligned} \tag{5}$$

With $V = 151{,}936$ the interval is $[0.50, 2.89]$ nats at $p^\star = 0.80$, $[0.33, 1.52]$ at 0.90, $[0.20, 0.80]$ at 0.95 and $[0.06, 0.18]$ at 0.99, so the top-1 probability determines the entropy only within a wide range. In the other direction, a fixed entropy leaves a wide range of top-1 probabilities, because at $H = 0.5$ nats the value of $p^\star$ can lie anywhere in $[0.80, 0.97]$.

Entropy after the mask. Because the mask removes the top token and renormalizes the remaining mass, it gives a distribution $q(y) = \pi(y)/(1-p^\star)$ for $y \neq y^\star$, and the entropy of this distribution has a closed form:

$$H_{\text{post}} = \frac{H + p^\star \log p^\star}{1-p^\star} + \log(1-p^\star). \tag{6}$$

Since a trace stores $p^\star$ to five decimals, we evaluate this form only where $1-p^\star > 10^{-3}$, below which the denominator loses too much precision.

The ratio at a masked position. The mask also changes the importance ratio, because the sampling engine returns the log-probability of each generated token under the distribution it sampled from. At a masked position, that distribution is $\tilde{\pi}_\theta$ of Equation 3, whose value at the sampled token is $\pi_\theta(o_t \mid \cdot)/(1-p^\star)$, and the trainer therefore uses this value as the old log-probability. Hence, at the first update, before θ has moved, the ratio is $\pi_\theta(o_t \mid \cdot)\big/\big(\pi_\theta(o_t \mid \cdot)/(1-p^\star)\big) = 1-p^\star$, which is Equation 4. Later updates move the ratio by the same factor as any other token, whereas the policy samples the positions after the mask from π_θ itself, so their ratio is one.

J. Effect of the second wave on the update

The derivation above concerns one masked position, and we also measure what the second wave does to the update as a whole. Table 11 compares the two runs of Table 1 on four quantities that the trainer logs, over the same window as Table 8, and we again report the mean over the first half of that window and the mean over the second half.

Three of these rows carry the result. First, the clip range catches more positions under CGE than under plain GRPO, at 0.023 and 0.021 against 0.005 and 0.009, and we attribute this to the forced tokens of the second wave, because a forced token starts at a ratio of $1-p_t^\star \le 0.05$, so the clip gives it no gradient whenever its advantage is negative (Appendix I). However, that share stays near two positions in a hundred, so the clip absorbs the perturbation rather than removing the rollout that carries it. Second, the magnitude of the advantage is larger under CGE early in training, and we attribute this to the restored groups, because a group whose rewards vary gives a non-zero advantage to every one of its rollouts. Third, the two runs reach the same training reward, at 0.360 against 0.355 in the second half of the window, so CGE obtains the extra groups of Table 8 at the same training reward as plain GRPO. In addition, the gradient norm stays within 0.14 of plain GRPO over the whole window, hence the second wave does not change the scale of the update.

removed	sampled instead	what the substitution changes
the value that a hop reads		
`B`	`C`	the answer the model was about to name
`right`	`left`	the side of the frame that a spatial hop reads
`bottom`	`top`	the vertical half that a spatial hop reads
`before`	`after`	the direction of an order hop
`orange`	`yellow`	the colour that an attribute hop reads
`arms`	`hands`	the part of a person that a hop refers to
where the rest of the response goes		
`on`	`in`	the relation between two objects
`the`	`left`	a side the model had not yet named
`is`	`appears`	how firmly the model commits to what it reports
`Second`	`First`	the step of the chain the model is working on

Table 9. Notable substitutions of the CGE run.

over the 160,000 masked positions	value
where the mask acts	
median position of a masked token in the response	242
share within the first 100 tokens of the response	0.30
share beyond token 400 of the response	0.37
how certain the policy is at a masked position	
mean top-1 probability	0.98
share whose top-1 probability exceeds 0.99	0.42
share where the second most likely token holds more than half of the rest	0.46
which token the sampler draws instead	
share where the sampler draws the second most likely token	0.50
share where it draws the second or the third most likely token	0.65
share where it draws a token outside the 8 that the trace records	0.18

Table 10. The masked positions of the CGE run, and the tokens the sampler draws in their place.

	plain GRPO		CGE	
per training step	early	later	early	later
share of clipped positions	0.005	0.009	0.023	0.021
mean magnitude of the advantage	0.337	0.332	0.393	0.338
gradient norm	1.23	1.30	1.26	1.16
mean training reward	0.302	0.355	0.293	0.360

Table 11. What the second wave changes in the update, on the two runs that continue on Video-HopChain.

per training step	plain GRPO	CGE	difference
step time (s)	2,275	2,285	+0.4%
wall-clock between steps (s)	2,435	2,459	+1.0%
generation (s)	1,097	1,084	−1.2%
actor update (s)	1,167	1,190	+2.0%
advantage and check (s)	0.15	0.58	+0.4 s
tokens per second	281	285	+1.4%
actor model utilization	0.137	0.139	—
trainer idle ratio	0.48	0.47	—

Table 12. Training cost of the two runs that continue on Video-HopChain, averaged over the run, on 4 nodes of 8 H100 GPUs. The wall-clock row is the mean gap between the timestamps of consecutive steps.

K. Computational cost and response length

We measure how much time the second wave costs, and how the mask changes the length of a response, on the two runs of Table 1 that continue on Video-HopChain, namely plain GRPO and CGE, which we train on the same 4 nodes.

Training time. Table 12 reports the timing that the trainer logs, averaged over the run. Every timing row of that table, except the advantage and check step, differs by at most 2.0% between the two runs, and the largest of these differences is the actor update, which the CGE run makes slower because this run produces slightly longer sequences. The advantage and check step itself grows from 0.15 to 0.58 seconds, which is negligible compared with a step of more than 2,000 seconds. In addition, token throughput differs by 1.4%, model utilization is 0.137 against 0.139, and the trainer waits for rollouts during about 48% of its time in both runs, so neither run is limited by the extra computation that CGE adds. This

result follows from the design of CGE, because we keep the compute budget fixed at 8 rollouts per question and we apply the mask inside the sampler as a logits processor, so the second wave adds no forward passes beyond the ones that plain GRPO already spends. Although the second wave waits for the rewards of the first wave, the asynchronous rollouter samples many other questions during that wait, so we measure no cost from this wait in the step time.

Response length. The second wave also changes the response length, because the CGE run gives longer responses on average, at 810 tokens against 708 for plain GRPO. Within the CGE run, however, the second wave gives shorter responses than the first, at 696 tokens against 923, so the mask does not lengthen a rollout by itself. The first wave of the CGE run is nevertheless longer than plain GRPO, at 923 tokens against 708, although that wave carries no mask. The policy that CGE trains therefore reasons for longer than the policy that plain GRPO trains, and it keeps that longer span at evaluation, where we disable the mask for both checkpoints. In addition, fewer than 0.4% of the responses of either run reach the 16,384-token limit, so truncation costs neither run a measurable amount of reward.