HopChain: Multi-Hop Data Synthesis for Generalizable Vision-Language Reasoning

Shenzhi Wang[♣,♠,*], **Shixuan Liu**[♠,*], **Jing Zhou**[♠], **Chang Gao**[♠], **Xiong-Hui Chen**[♠], **Binghai Wang**[♠], **An Yang**[♠], **Shiji Song**[♣], **Bowen Yu**[♠,✉,†], **Gao Huang**[♣,✉], **Junyang Lin**[♠]

[♣]LeapLab, Tsinghua University [♠]Qwen Team, Alibaba Inc.

Abstract

Vision-language models (VLMs) show strong multimodal capabilities but still struggle with fine-grained vision-language reasoning. We find that long chain-of-thought (CoT) reasoning exposes diverse failure modes, including perception, reasoning, knowledge, and hallucination errors, which can compound across intermediate steps. However, most existing data for reinforcement learning with verifiable rewards (RLVR) does not involve complex reasoning chains grounded in visual evidence throughout, leaving these failure modes understressed during training. We therefore propose **HopChain**, a scalable framework that constructs an out-of-distribution (OOD) proxy task: OOD only in query format, while strengthening foundational abilities shared with downstream benchmarks to support *generalizable* gains. Concretely, HopChain synthesizes *multi-hop vision-language reasoning* chains, each a logically dependent chain of instance-grounded hops where earlier hops establish the instances, sets, or conditions needed for later hops, ending in an unambiguous numerical answer for verifiable rewards. In experiments, we train Qwen3.5-35B-A3B and Qwen3.5-397B-A17B under two RLVR settings (with and without HopChain's multi-hop data) and compare them across 24 benchmarks spanning STEM and Puzzle, General VQA, Text Recognition and Document Understanding, and Video Understanding. Although the multi-hop data is not designed for any specific benchmark, it improves 20 of 24 benchmarks on both models, indicating broad and generalizable gains. Consistently, replacing full chained queries with half-multi-hop or single-hop variants reduces the average score across five representative benchmarks from 70.4 to 66.7 and 64.3, respectively. These results indicate that retaining a sufficient number of hops in multi-hop queries is important for preserving the training benefit of multi-hop data. Improvements are particularly substantial in long-CoT and ultra-long-CoT regimes, peaking at more than 50 accuracy points in the ultra-long-CoT regime. These results show that OOD proxy tasks for long-chain visual reasoning are a scalable RLVR supervision source for generalizable VLM reasoning.

1 Introduction

Vision-language models (VLMs) have achieved impressive performance across diverse multimodal benchmarks by integrating visual encoders with large language models (Liu et al., 2023a,b; Bai et al., 2023, 2025b; Chen et al., 2024c; OpenAI, 2023; Gemini Team et al., 2023). Recent advances in reinforcement learning with verifiable rewards (RLVR) have further improved VLMs' reasoning abilities by training them to produce step-by-step chain-of-thought solutions for questions with objectively verifiable answers (Shao et al., 2024; DeepSeek-AI, 2025). Despite this progress, a critical gap remains: VLMs struggle with tasks that require *fine-grained, multi-step vision-language*

*Equal contribution ✉Corresponding authors †Project lead.
Emails: wangshenzhi99@gmail.com, liushixuan66@gmail.com, yubowen.ph@gmail.com, gaohuang@tsinghua.edu.cn.
40th Conference on Neural Information Processing Systems (NeurIPS 2026).

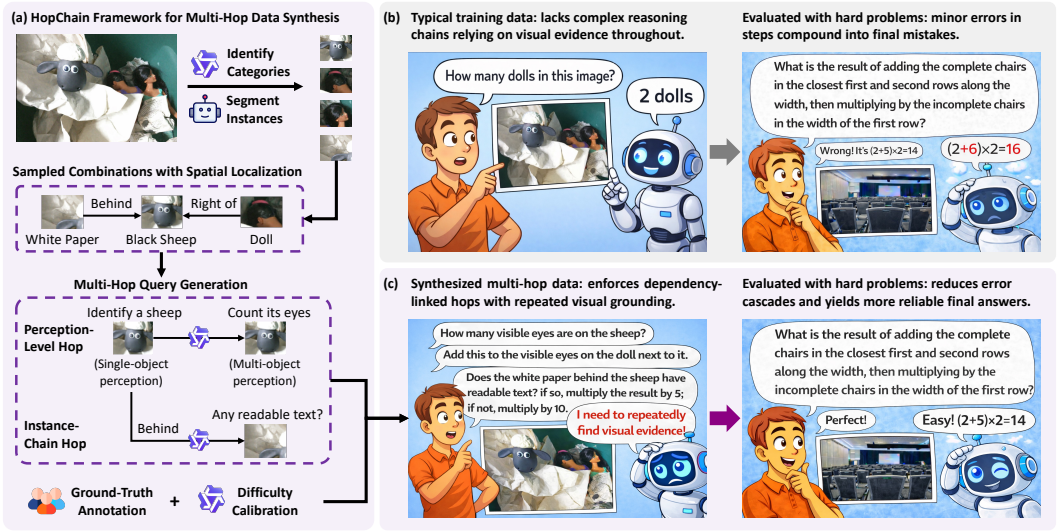


Figure 1: Overview of HopChain. **(a)** Four-stage synthesis pipeline: category identification, instance segmentation, multi-hop query generation, and ground-truth annotation with difficulty calibration. **(b)** Typical vision-language training data rarely involves complex reasoning chains that rely on continuous visual evidence; minor errors at intermediate long-CoT steps can compound into wrong final answers. **(c)** HopChain's multi-hop data forms logically dependent hop chains, where later hops build on the instances, sets, or conditions established by earlier hops and nearly every hop requires fresh visual re-grounding, encouraging continual visual evidence seeking during long CoT and reducing compounding errors.

reasoning, where the correct answer depends on carefully attending to multiple visual elements and their relationships within an image (Bigverdi et al., 2025; Ye et al., 2025; Jiang et al., 2025).

A natural question arises: *what prevents current VLMs from performing robust vision-language reasoning?* In Section 2, we analyze a central challenge: VLMs exhibit **diverse and compounding failure modes during long chain-of-thought (CoT) reasoning**, as longer reasoning chains may cause models to attend less faithfully to the image, execute incorrect intermediate reasoning steps, rely on incomplete knowledge, or produce hallucinated and weakly grounded intermediate content that then compounds across subsequent steps. Related failure patterns have also been reported in prior analyses of amplified multimodal hallucination, evidential drift, object hallucination, visual illusion, and image-context reasoning errors (Liu et al., 2025; Luo et al., 2025; Rohrbach et al., 2018; Guan et al., 2024; Leng et al., 2024). Critically, as depicted in Figure 1(b), most existing vision-language training data does not involve particularly complex reasoning chains that depend on visual evidence throughout the process. As a result, these long-CoT weaknesses remain largely unexposed during training. This observation suggests that relying on, or simply expanding, existing vision-language RLVR training data is insufficient; what is needed is training data that *structurally forces* the model to seek visual evidence at each step of long-CoT reasoning, strengthening step-by-step vision-language reasoning and improving generalization across diverse scenarios.

Motivated by this insight, we propose **HopChain**[1], a scalable framework for synthesizing multi-hop vision-language reasoning data specifically for RLVR training of VLMs. Rather than targeting any specific downstream benchmark, we treat the multi-hop data synthesized by HopChain as an out-of-distribution (OOD) proxy task: only the query format is OOD, while the foundational abilities it strengthens—sustained visual grounding, multi-step instance-chain reasoning, and error control across long chains—are shared with downstream benchmarks at test time. This shared foundation supports broad, *generalizable* gains that we later show transfer across diverse benchmark families. Concretely, each synthesized multi-hop query forms a logically dependent chain of instance-grounded hops, where earlier hops establish the instances, sets, or conditions needed for later hops, forcing repeated visual re-grounding throughout training rather than language-only shortcuts. At the same

[1]Throughout this paper, HopChain refers to our synthesis framework, while *multi-hop vision-language reasoning data*, or simply *multi-hop data*, refers to the training data synthesized by HopChain.

time, each query terminates in a specific, unambiguous numerical answer that is easy to verify for RLVR; moreover, because the hops are logically dependent, obtaining the correct final number usually also requires the intermediate reasoning steps to be correct.

This multi-hop structure consists of two complementary hop types: perception-level hops and instance-chain hops. A *perception-level hop* switches between single-object perception (e.g., read text, identify color, determine position) and multi-object relationship reasoning (e.g., compare sizes, count objects satisfying a condition, determine spatial arrangement), while remaining grounded in the instances, sets, or conditions established by earlier hops. An *instance-chain hop* follows an explicit dependency chain (e.g., instance A $\rightarrow$ B $\rightarrow$ C), where the next instance can be identified only from the instances, sets, or conditions established by earlier hops. We require each query to combine both hop types in a logically dependent chain, where earlier hops establish the instances, sets, or conditions needed for later hops. This design forces fresh visual grounding at every step, blocks language-only shortcuts, and exposes diverse long-CoT failure modes during training, as shown in Figure 1(c).

To construct such multi-hop data at scale, HopChain adopts a scalable four-stage data synthesis pipeline, summarized in Figure 1(a): (1) *category identification* to enumerate semantic categories in each image; (2) *instance segmentation* to localize individual instances for the identified semantic categories; (3) *multi-hop query generation* that constructs logically chained questions over combinations of instances; and (4) *human-in-the-loop verification*, adopting the majority answer when at least three of four model-assisted answers agree and escalating the remaining cases to manual adjudication. The pipeline scales to broad image collections with sufficient detectable instances while maintaining strict quality control.

We train VLMs by applying RLVR with Soft Adaptive Policy Optimization (SAPO) (Gao et al., 2025) on this multi-hop data, and validate the effectiveness on Qwen3.5-35B-A3B and Qwen3.5-397B-A17B (Qwen Team, 2026) across 24 benchmarks spanning STEM and Puzzle, General VQA, Text Recognition and Document Understanding, and Video Understanding. Compared with RLVR on the original RLVR data alone, adding this multi-hop data improves 20 out of 24 benchmarks on both models despite not targeting any specific benchmark, indicating broad, generalizable gains. To demonstrate that full chained queries are important, we replace them with half-multi-hop or single-hop variants on Qwen3.5-35B-A3B, reducing the average score across five representative benchmarks from 70.4 to 66.7 and 64.3, respectively. Multi-hop training yields particularly substantial improvements in long-CoT and ultra-long-CoT regimes, peaking at more than 50 accuracy points in the ultra-long-CoT regime on Qwen3.5-397B-A17B. When we independently sample each synthesized query multiple times on both Qwen3.5-35B-A3B and Qwen3.5-397B-A17B, more than half of the queries are partially solved, while the remaining outcomes span fully incorrect to fully correct and the larger model shifts toward more fully solved queries. This indicates a broad difficulty range suitable for both smaller and larger models. The corrected error types also closely follow the original error distribution before adding this multi-hop data, indicating gains that are broad rather than confined to a single narrow failure mode. These experiments establish HopChain as an effective framework for synthesizing multi-hop data that improves generalizable vision-language reasoning beyond the synthesized training distribution.

Our main contributions are as follows:

- We identify diverse and compounding failure modes during long CoT reasoning as a key barrier to vision-language generalization, and show why relying on or simply expanding existing vision-language RLVR training data is insufficient (§ 2).
- We formalize multi-hop vision-language reasoning as an out-of-distribution (OOD) proxy task that is OOD only in query format while sharing foundational abilities with downstream benchmarks, and build HopChain, a scalable synthesis pipeline whose perception-level and instance-chain hops form logically dependent chains, forcing repeated visual re-grounding while keeping the final answer directly verifiable for RLVR (§ 3).
- Through extensive experiments, we verify that RLVR on the multi-hop data synthesized by HopChain yields broad, generalizable gains: 20 out of 24 benchmarks improve on both Qwen3.5-35B-A3B and Qwen3.5-397B-A17B, while additional analyses show that full chained queries are important, multi-hop data strengthens long-CoT vision-language reasoning, and the synthesized data spans a broad difficulty range while letting trained models correct diverse error types (§ 4).

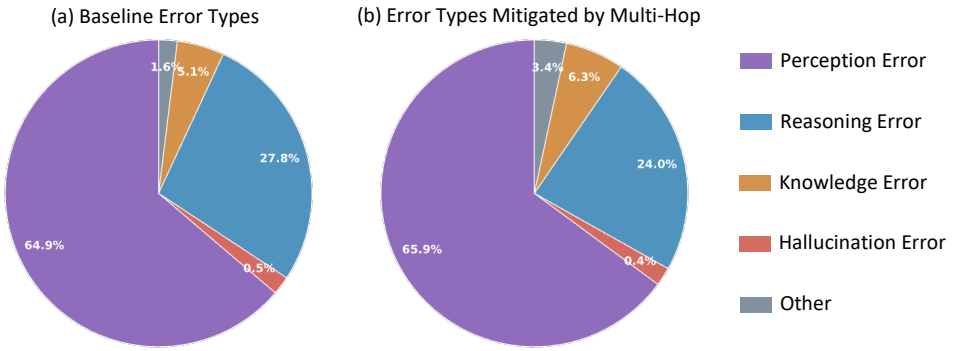


Figure 2: Error-type distributions before and after adding multi-hop data in RLVR. **(a)** Error distribution for `RLVR w/o Multi-Hop` on the 24 benchmarks in Appendix D.1: perception errors form the largest category, reasoning errors come next, and knowledge, hallucination, and other errors form smaller fractions. **(b)** Errors corrected by `RLVR w/ Multi-Hop` track (a), showing broadly distributed gains; see Figure 3 for representative cases.

2 Diverse Failure Modes in Long Chain-of-Thought Reasoning[2]

Long CoT reasoning demands sustained visual grounding at each step, and any intermediate error in perception, reasoning, knowledge, or hallucination can compound into an incorrect final answer (Liu et al., 2025; Luo et al., 2025; Rohrbach et al., 2018; Guan et al., 2024).

On each of the 24 benchmarks in Appendix D.1, annotators classify the failure type of 20 randomly sampled incorrect responses from Qwen3.5-397B-A17B `RLVR w/o Multi-Hop` (RLVR on the original data only; see Section 4). Figure 2(a) shows that long-CoT failures are diverse: perception errors form the largest category, followed by reasoning errors, with knowledge, hallucination, and other errors representing smaller fractions; Figure 3 provides representative cases involving counting, astronomy, parking, and zebra stripe patterns that share the same structure: a faulty intermediate step appears mid-chain and later steps inherit it (detailed analysis in Appendix C). By contrast, `RLVR w/ Multi-Hop` (RLVR with our multi-hop data added) more often recovers the correct visual evidence at each hop and propagates it through subsequent steps, mitigating compounding error chains.

Improving long CoT reasoning thus requires an RLVR data construction method that is applicable to diverse images, forces repeated grounding, and chains reasoning steps so that the result of each step is used to locate, verify, or constrain the visual evidence needed for later steps (Bigverdi et al., 2025; Ye et al., 2025; Jiang et al., 2025), which is exactly the goal of HopChain in the next section.

3 Boosting Vision-Language Generalization by Synthesizing Multi-Hop Data

Motivated by the diverse compounding failure modes identified in Section 2, we synthesize multi-hop vision-language reasoning data as an out-of-distribution (OOD) proxy task that structurally forces the model to seek visual evidence at each step of long-CoT reasoning, while ensuring each query terminates in a specific, unambiguous numerical answer compatible with RLVR. Figure 1(a) summarizes the full synthesis pipeline, and Figure 4 provides representative synthesized examples.

3.1 Multi-Hop Vision-Language Reasoning Definition

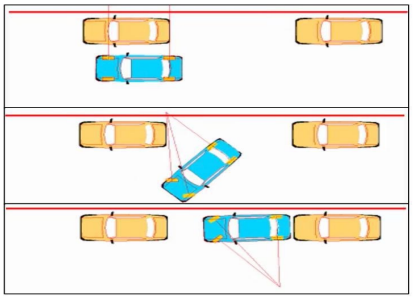
We define target queries through three reasoning levels. **Level 1** is single-object perception, such as reading text or identifying object attributes including color, shape, size, position, and category. **Level 2** is multi-object perception, such as spatial, comparative, or counting relations across objects. **Level 3** chains multiple Level 1 and Level 2 steps into one query.

Within a Level 3 query, consecutive hops can be linked in two complementary dimensions. **Perception-level hop:** the next step changes the kind of perception being performed, for example from a Level 1 single-object judgment to a Level 2 relational judgment, or vice versa, while remaining

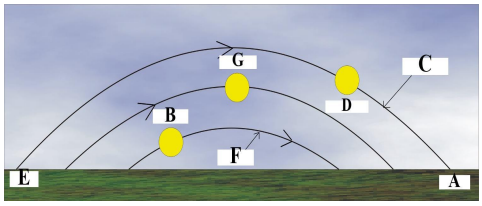
[2]Due to page limits, the preliminaries are deferred to Appendix A and the related work to Appendix B.



How many dots do all ladybirds have together?
Baseline: [...] Top-middle: 4 dots [...] Bottom-left: 6 dots [...] Bottom-right: 7 dots [...] Total: 2 + 4 + 3 + 6 + 7 = 22. ✗
Ours: [...] Top-middle: 3 dots [...] Bottom-left: 5 dots [...] Bottom-right: 6 dots [...] Total: 2 + 3 + 3 + 5 + 6 = 19. ✓
Error analysis: Perception Error (Miscounted ladybird dots: top-middle 3 to 4, bottom-left 5 to 6, bottom-right 6 to 7).



According to the positive sequence of the images, does the blue car attempt to leave the parking space?
Baseline: Top: car parked parallel [...] Middle: wheels turned angling outward [...] Bottom: straightened in lane [...] interprets sequence as leaving → Answer: Yes. ✗
Ours: Top: car positioning alongside [...] Middle: reversing at angle to back in [...] Bottom: straightened and stopped parked [...] interprets sequence as entering → Answer: No. ✓
Error analysis: Reasoning Error (Correctly perceived positions but logically misinterpreted entry as exit).



Which season does C represent?
Baseline: Highest arc = summer [...] Lowest arc = winter [...] Middle arc = equinox/spring [...] Label C points to middle arc [...] → Answer: spring. ✗
Ours: Highest arc (labeled C) = summer solstice [...] Lowest arc = winter [...] Middle arc = spring/fall [...] Label C points to highest arc [...] → Answer: summer. ✓
Error analysis: Perception Error (Misjudged the arrow pointing to the highest arc as pointing to the middle arc).



What percentage of the zebra's stripes are horizontal?
Baseline: Neck/body stripes vertical [...] legs/rump stripes horizontal [...] vertical is dominant pattern [...] percentage of horizontal = 0 → Answer: 0. ✗
Ours: Front body stripes vertical [...] hindquarters/legs stripes horizontal [...] roughly half and half distribution [...] percentage of horizontal = 50 → Answer: 50. ✓
Error analysis: Reasoning Error (Misidentification of dominant as 100%, ignoring horizontal stripes).

Figure 3: Qualitative examples of unreliable visual perception in long CoT reasoning. We show failure cases from the benchmarks in Tables 1 and 2 using "RLVR w/o Multi-Hop" (baseline), alongside correct answers from "RLVR w/ Multi-Hop" (ours). For brevity, unimportant parts of the long CoT are omitted with [. . .]. The baseline error is highlighted in red, and the failure reason is given in the error analysis.

grounded in the instances, sets, or conditions established by earlier hops. **Instance-chain hop:** the next step moves to a new instance along an explicit dependency chain (e.g., A → B → C), identified only from instances, sets, or conditions established by earlier hops.

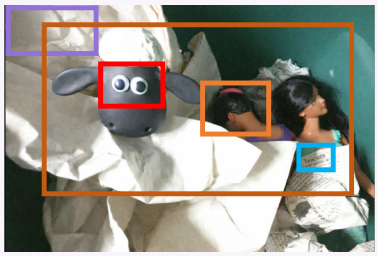
Each query must satisfy three structural conditions: (i) it must be Level 3, (ii) combine both hop types, and (iii) form a logically dependent chain where earlier hops establish the instances, sets, or conditions needed for later hops; we prefer the dependency chains and perception-level transitions to be intertwined as tightly as possible. Each query must also terminate in a specific, unambiguous numerical answer for RLVR, allowing direct verification of the final answer and providing an end-to-end consistency signal for the reasoning chain. This rules out pseudo-multi-hop queries with loosely connected substeps and targets exactly the failures in Figure 3: the model must recover the correct visual evidence at each step, then use it to determine the next object, relation, or computation.

Although the final answer must be numerical for reward verification, the intermediate hops can involve OCR, comparison, and spatial grounding. For example, an unambiguous size comparison can be encoded as 1 if object A is larger than B and 2 if B is larger than A.

Multi-Hop Query #1:

H1: Start by locating the black sheep toy on the left side of the image. Count the number of white eyes visible on this sheep.
H2: Next, inspect the large area of crumpled white paper that forms the background strictly behind and above the sheep; determine if there is any readable text on this specific background paper. If text is present, set a reference value to 5; if no text is visible, set the reference value to 10.
H3: Then, move right to identify the first doll. Count the number of eyes visible on this doll's face.
H4: Continue to the right to find a second doll. Locate the crumpled paper positioned directly in front of this second doll, read the bold title word starting with 'T', and count its letters.
H5: Now, perform the calculation: add the sheep's eye count to the first doll's eye count, multiply this sum by the letter count of the word, and then add the reference value from the background paper check.
H6: Finally, multiply this total by the number of distinct toy figures present in the image. What is the final number?

Ground-Truth Answer #1:





H1: The black sheep has 2 visible eyes. Answer: 2.



H2: No readable text. Answer: 10.



H3: **The doll to the sheep's right has no visible eyes.** Answer: 0.



H4: The bold title word starting with 'T' has 7 letters. Answer: 7.

H5: (H1 + H3) x H4 + H2 = 24. Answer: 24.

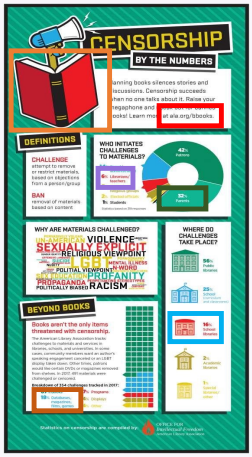


H6: The number of distinct toy figures is 3. Answer: H5 x 3 = 72.

Multi-Hop Query #2:

H1: Start by examining the introductory text paragraph located on the right side of the image, directly below the main 'CENSORSHIP' title banner. Identify the URL mentioned at the very end of this paragraph and count the number of letters in the portion of the URL strictly after the forward slash.
H2: Use this letter count to find the category in the 'WHO INITIATES CHALLENGES TO MATERIALS?' donut chart that has a percentage value exactly equal to this number. Identify the specific color used for this category's slice in the chart.
H3: Next, look at the large illustration in the top-left corner featuring a book and a megaphone. Determine if the primary color of the book's cover matches the color identified from the chart slice in the previous step.
H4: If they match, locate the facility type in the 'WHERE DO CHALLENGES TAKE PLACE?' section (on the right) whose corresponding building icon is drawn in this same color. Note its percentage. If they don't match, the final number is 0.
H5: Then, find the 'BEYOND BOOKS' section in the bottom-left and identify the percentage of challenges that target 'Databases'. Sum this 'Databases' percentage with the percentage from the facility type found earlier.
H6: Finally, return to the 'WHO INITIATES' donut chart, identify the category whose percentage is numerically closest to this calculated sum, and subtract the initial letter count (from the URL step) from this category's percentage. What is the final number?

Ground-Truth Answer #2:





H1: The number of letters strictly after the forward slash is 6. Answer: 6.



H2: The donut chart's 6% segment is red. Answer: red.



H3: The book cover's primary red color matches the chart slice. Answer: yes.



H4: Since H3 answered yes, the red building icon's percentage is 16. Answer: 16.



H5: Databases account for 18% of challenges. Answer: H4 + 18 = 34.



H6: The donut chart percentage closest to H5 is 32. Answer: 32 – H1 = 26.

Figure 4: Examples of synthesized multi-hop data. Purple-highlighted text denotes instance-chain dependencies; colored rectangles mark the corresponding hop-wise grounded instances and answers.

Table 1: Results of Qwen3.5-35B-A3B. `Before RLVR` is the post-SFT model; `RLVR w/o Multi-Hop` uses the original RLVR data; `RLVR w/ Multi-Hop` adds HopChain data.

Benchmark	Before RLVR	RLVR w/o Multi-Hop	RLVR w/ Multi-Hop
STEM and Puzzle			
MathVision	61.97	73.71	**76.05**
MMMU-Pro	66.07	69.25	**70.64**
MMMU	75.89	**78.89**	78.33
MathVista(mini)	82.80	**85.50**	85.00
BabyVision	14.95	21.91	**22.68**
ZeroBench	1	1	**3**
EMMA(mini)	41.88	53.00	**58.00**
LogicVista	63.85	74.66	**75.56**
General VQA			
$\text{MMBench}_{\text{CN-DEV-V1.1}}$	87.46	90.17	**90.48**
$\text{MMBench}_{\text{EN-DEV-V1.1}}$	88.47	90.63	**91.49**
RealWorldQA	75.42	78.17	**79.35**
MMStar	75.20	78.53	**78.60**
HallusionBench	66.49	**66.64**	66.50
AI2D_TEST	88.44	90.87	**91.29**
ERQA	44.50	48.25	**51.38**
Text Recognition and Document Understanding			
CharXiv	61.30	69.00	**73.10**
DocVQA_VAL	95.00	95.13	**95.55**
InfoVQA_VAL	86.81	87.44	**90.17**
Video Understanding			
$\text{VideoMME}_{\text{w/o sub.}}$	73.41	74.63	**75.00**
VideoMMMU	70.67	73.33	**74.78**
MMVU-CoT	63.70	65.80	**68.90**
MVBench	69.18	69.95	**70.73**
LVBench	51.13	**54.49**	53.20
MLVU (M-Avg)	77.92	77.69	**79.53**

3.2 Data Synthesis Pipeline

Given the query definition, we synthesize candidates through a three-stage pipeline; this subsection covers Stages 1–3 and Section 3.3 covers Stage 4.

Stage 1: Category Identification. We use a VLM, Qwen3-VL-235B-A22B-Thinking (Bai et al., 2025a), to identify the semantic categories present in each image (e.g., "car," "person," "sign") without localization, providing the candidate visual entities that later hops can operate over.

Stage 2: Instance Segmentation. For each semantic category, we use SAM3 (Carion et al., 2025) to generate segmentation masks and bounding boxes, resolving categories into concrete spatially-localized instances that hop-based reasoning can anchor to.

Stage 3: Multi-Hop Query Generation. We form combinations of 3–6 instances and use Qwen3-VL-235B-A22B-Thinking to generate multi-hop queries for each combination; the model receives the original image plus cropped patches of each instance (used only at design time to identify appearance and location), and the full prompt is provided in Appendix E. To address the failure modes from Section 2, each synthesized query must (i) involve as many instances as possible while remaining answerable from the original image alone; (ii) describe objects only by spatial, contextual, or visual attributes; (iii) terminate in a specific, unambiguous numerical answer; and (iv) avoid any reference to segmentation masks, bounding boxes, or patch images. Reasoning hops must form a logically dependent chain where earlier hops establish the instances, sets, or conditions needed for later hops, requiring the model to recover and retain intermediate visual evidence at each step rather than rely on language-only heuristics.

Table 2: Results of Qwen3.5-397B-A17B, the larger model counterpart of Table 1. The three settings follow the same definitions as in Table 1.

Benchmark	Before RLVR	RLVR w/o Multi-Hop	RLVR w/ Multi-Hop
STEM and Puzzle			
MathVision	77.38	81.68	**83.71**
MMMU-Pro	73.03	75.06	**76.47**
MMMU	79.78	81.67	**82.89**
MathVista(mini)	87.50	88.30	**89.00**
BabyVision	17.01	28.61	**32.22**
ZeroBench	3	4	**8**
EMMA(mini)	58.13	66.25	**69.00**
LogicVista	75.62	80.69	**81.59**
General VQA			
$\text{MMBench}_{\text{CN-DEV-V1.1}}$	89.47	91.41	**91.72**
$\text{MMBench}_{\text{EN-DEV-V1.1}}$	90.71	**92.49**	91.56
RealWorldQA	79.87	79.87	**81.70**
MMStar	80.00	**81.73**	80.67
HallusionBench	65.76	67.48	**67.86**
AI2D_TEST	91.45	92.81	**92.97**
ERQA	53.75	**60.50**	60.00
Text Recognition and Document Understanding			
CharXiv	70.00	74.60	**77.20**
DocVQA_VAL	95.93	95.98	**96.03**
InfoVQA_VAL	90.42	90.83	**92.20**
Video Understanding			
$\text{VideoMME}_{\text{w/o sub.}}$	76.56	78.30	**80.41**
VideoMMMU	76.67	78.89	**80.00**
MMVU-CoT	70.20	72.30	**72.50**
MVBench	69.63	73.03	**73.31**
LVBench	58.30	**59.13**	59.07
MLVU (M-Avg)	81.46	82.43	**82.52**

3.3 Ground-Truth Annotation and Difficulty Calibration

After Stages 1–3 produce candidate queries, Stage 4 adds two complementary components: *ground-truth annotation* (filters low-quality queries and provides reliable answers via annotator agreement) and *difficulty calibration* (uses these answers to remove overly easy queries).

Stage 4: Ground-Truth Annotation and Difficulty Calibration. Four model-assisted annotators independently produce initial answers for each query, each using an assigned model. If at least three answers agree, the majority answer is adopted as the ground truth; otherwise, the query is escalated to manual adjudication. Queries with ambiguous references or other quality issues are discarded. We then remove overly easy queries using a weaker model and the pre-RLVR model; the weaker-model filter uses eight samples per query and discards queries at 100% accuracy, yielding a final dataset with verified difficulty and reliable annotations.

Together, the pipeline scales to broad image collections with model and human quality control. Compared with typical RLVR-style questions (Figure 3), the synthesized data (Figure 4) enforces hop-wise dependency and repeated visual re-grounding.

4 Experiments

We evaluate HopChain from three angles: (i) whether multi-hop data improves general vision-language reasoning across model scales and benchmark families, (ii) whether the full multi-hop formulation is necessary versus weaker variants, and (iii) whether the gains align with our claims

about long-CoT robustness, difficulty coverage, and broad failure-mode correction; Appendix D details training hyperparameters, image filtering, the 24-benchmark list, and error-type analyses.

We evaluate **Qwen3.5-35B-A3B** and **Qwen3.5-397B-A17B** (Qwen Team, 2026) under 3 settings: `Before RLVR` (the post-SFT model and shared starting point for both RLVR settings), `RLVR w/o Multi-Hop` (RLVR on the original data only), and `RLVR w/ Multi-Hop` (RLVR on a mixture of the original data and our synthesized multi-hop data). We use SAPO as the optimization algorithm for RLVR training (Gao et al., 2025).

Main results. As reported in Tables 1 and 2, adding HopChain's multi-hop data improves 20 of 24 benchmarks for both Qwen3.5-35B-A3B and Qwen3.5-397B-A17B. Specifically, on Qwen3.5-35B-A3B, multi-hop training improves 6 of 8 STEM-and-Puzzle, 6 of 7 General-VQA, 3 of 3 Text Recognition and Document Understanding, and 5 of 6 Video Understanding benchmarks; on Qwen3.5-397B-A17B, the corresponding counts are 8 of 8, 4 of 7, 3 of 3, and 5 of 6, with improved cases shown in Figure 3. Notably, despite training only on images, the data still improves 5 of 6 Video Understanding benchmarks on both models, suggesting HopChain's gains are generalizable. Across the 24 benchmarks, the unweighted mean gains over RLVR w/o Multi-Hop are 1.28 and 1.02 points for Qwen3.5-35B-A3B and Qwen3.5-397B-A17B, respectively; the corresponding median gains are 0.88 and 0.80 points.

Hop-structure ablation. We compare three training-query settings: Full retains all original hops, Half-Multi-Hop retains the latter half of the original hop sequence, and Single Hop retains only the final hop. For Half-Multi-Hop and Single Hop, references to removed prerequisites are recursively replaced with their ground-truth intermediate answers, yielding self-contained queries with the same terminal numerical target. After conversion, we used LLM-based verification and human spot checks to confirm that substituted references matched the removed hops' ground-truth answers, the queries were self-contained, and the terminal targets were unchanged. The full-hop setting is consistently best in Figure 5: the average score across the five representative benchmarks drops from 70.4 to 66.7/64.3 for half-multi-hop/single-hop variants. These results indicate that retaining a sufficient number of hops in multi-hop queries is important for preserving the training benefit of multi-hop data. Across all 24 benchmarks, Full outperforms both shortened variants, with unweighted average scores of 70.80, 65.53, and 63.77 for Full, Half, and Single, respectively (Appendix D.4).

Analysis by reasoning length. Figure 6 bins Qwen3.5-397B-A17B responses by token count. The advantage of `RLVR w/ Multi-Hop` over `RLVR w/o Multi-Hop` is particularly substantial in long-CoT and ultra-long-CoT regimes, peaking at over 50 accuracy points in the latter.

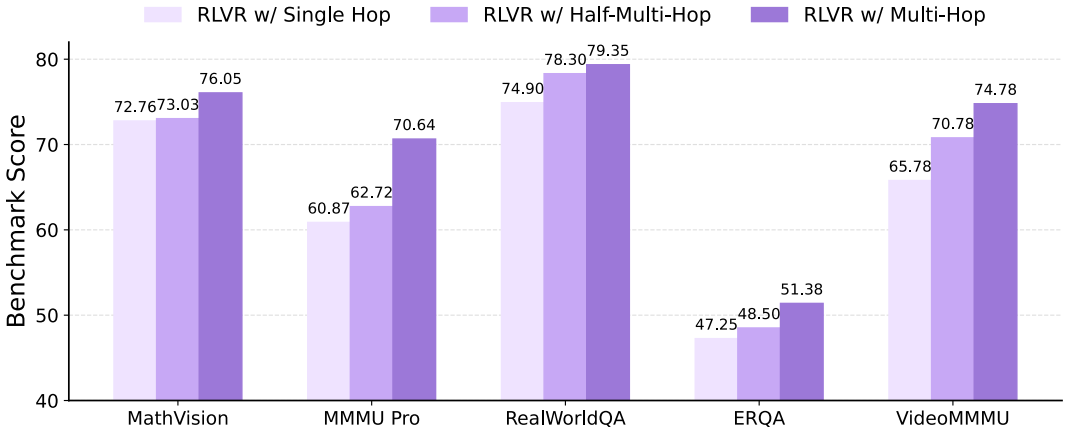


Figure 5: Benchmark-level comparison on Qwen3.5-35B-A3B under three training-query settings, evaluated on the subset of benchmarks shared by all three runs: `RLVR w/ Single Hop` keeps only the final hop, `RLVR w/ Half-Multi-Hop` keeps only the latter half, and `RLVR w/ Multi-Hop` keeps the full multi-hop training queries. The full multi-hop setting performs best across all five representative benchmarks (MathVision, MMMU-Pro, RealWorldQA, ERQA, VideoMMMU).

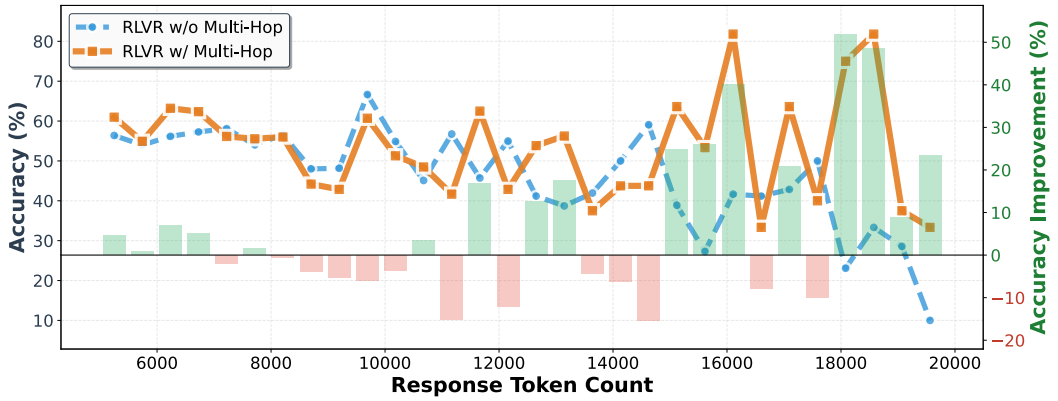


Figure 6: Performance as a function of response length on Qwen3.5-397B-A17B; bars show the accuracy improvement of `RLVR w/ Multi-Hop` over `RLVR w/o Multi-Hop`.

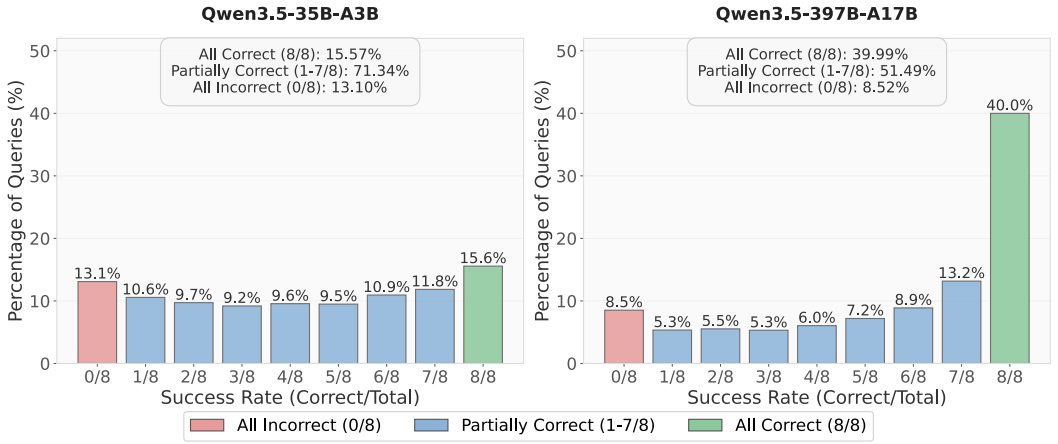


Figure 7: Query-level success-rate distributions for Qwen3.5-35B-A3B and Qwen3.5-397B-A17B from eight independent samples per multi-hop query.

Difficulty coverage across model scales. Figure 7 shows query-level success-rate distributions from 8 samples per query. For both models, more than half of queries fall into the `Partially Correct` regime (1–7 correct samples per query) and span multiple success buckets, indicating broad difficulty coverage suited to both scales; further error-subtype analysis is in Appendix D.2.

5 Conclusion

We introduce HopChain, a scalable framework for synthesizing chained, visually grounded, numerically verifiable RLVR data as an out-of-distribution (OOD) proxy task targeting the foundational abilities behind long-chain visual reasoning. Across Qwen3.5-35B-A3B and Qwen3.5-397B-A17B, adding HopChain data improves 20 out of 24 benchmarks despite not being designed for any specific benchmark, transferring strongly across diverse benchmark families including video understanding despite being synthesized from images alone. These results show that OOD proxy tasks for long-chain visual reasoning are a scalable source of RLVR supervision for generalizable VLM reasoning. The ablation results indicate that retaining a sufficient number of hops in multi-hop queries is important for preserving the training benefit of multi-hop data.

As for limitations and future work, the current pipeline depends on successful instance segmentation and excludes images with no SAM3-segmentable instances; a natural next step is to introduce complementary data-construction routes for such images while preserving the core design principle of chained visual grounding for long-CoT RLVR training. Our evaluation covers two model scales within the Qwen3.5 family; generalization to other model families remains to be established.

Acknowledgments and Disclosure of Funding

This work is supported in part by the National Key R&D Program of China under Grant 2024YFB4708200, the National Natural Science Foundation of China under Grants U24B20173, U2541227 and 62625604, and the Scientific Research Innovation Capability Support Project for Young Faculty under Grant ZYGXQNJSKYCXNLZCXM-I20.

The authors have no competing interests to declare.

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 39–48, 2016.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023.

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*, 2025a.

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

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G. Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3836–3845, 2025.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, François Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. Sam 3: Segment anything with concepts, 2025. URL `https://arxiv.org/abs/2511.16719`.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. Charles, Yiping Bao, Yuantao Fan, Guopeng Li, Haiyang Shen, Xuanzhong Chen, Wendong Xu, Shuzheng Si, Zefan Cai, Wenhao Chai, Ziqi Huang, Fangfu Liu, Tianyu Liu, Baobao Chang, Xiaobo Hu, Kaiyuan Chen, Yixin Ren, Yang Liu, Yuan Gong, and Kuan Li. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*, 2023.

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? *arXiv preprint arXiv:2403.20330*, 2024a.

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

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks via progressive image-text alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024c.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hao Ding, Zhichuan Yang, Weijie Ge, Ziqin Gao, Chaoyi Lu, and Lei Zhao. Tikart: Stabilizing aperture-guided fine-grained visual reasoning with reinforcement learning. *arXiv preprint arXiv:2602.14482*, 2026.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

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

Gemini Robotics Team, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, et al. Gemini robotics: Bringing ai into the physical world. *arXiv preprint arXiv:2503.20020*, 2025. URL `https://arxiv.org/abs/2503.20020`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Sorber, Johan Schalkwyk, Andrew M. Dai, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14375–14385, 2024.

Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14953–14962, 2023.

Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444*, 2025.

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. URL `https://arxiv.org/abs/2501.13826`.

Drew A. Hudson and Christopher D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6700–6709, 2019.

Chaoya Jiang, Yongrui Heng, Wei Ye, Han Yang, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. Vlm-r^3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. *arXiv preprint arXiv:2505.16192*, 2025.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2901–2910, 2017.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pages 235–251. Springer International Publishing, 2016.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13872–13882, 2024.

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

Xirui Li, Ming Li, and Tianyi Zhou. What does rl improve for visual reasoning? a frankenstein-style analysis. *arXiv preprint arXiv:2602.12395*, 2026.

Chengzhi Liu, Zhongxing Xu, Qingyue Wei, Juncheng Wu, James Zou, Xin Eric Wang, Yuyin Zhou, and Sheng Liu. More thinking, less seeing? assessing amplified hallucination in multimodal reasoning models. *arXiv preprint arXiv:2505.21523*, 2025.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023b.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023c.

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? *arXiv preprint arXiv:2307.06281*, 2023d.

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 (ICLR)*, 2024.

Mi Luo, Zihui Xue, Alex Dimakis, and Kristen Grauman. When thinking drifts: Evidential grounding for robust video reasoning. *arXiv preprint arXiv:2510.06077*, 2025.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Ruben Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022.

Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. Point-rft: Improving multimodal reasoning with visually grounded reinforcement finetuning. *arXiv preprint arXiv:2505.19702*, 2025.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744, 2022.

Qwen Team. Qwen3.5: Accelerating productivity with native multimodal agents, February 2026. URL `https://qwen.ai/blog?id=qwen3.5`.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, Vatsal Raina, Hanyi Xiong, Vishaal Udandarao, Jingyi Lu, Shiyang Chen, Sam Purkis, Tianshuo Yan, Wenye Lin, Gyungin Shin, Qiaochu Yang, Anh Totti Nguyen, David I. Atkinson, Aaditya Baranwal, Alexandru Coca, Mikah Dang, Sebastian Dziadzio, Jakob D. Kunz, Kaiqu Liang, Alexander Lo, Brian Pulfer, Steven Walton, Charig Yang, Kai Han, and Samuel Albanie. Zerobench: An impossible visual benchmark for contemporary large multimodal models. *arXiv preprint arXiv:2502.09696*, 2025.

Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. Object hallucination in image captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4035–4045, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Hai-Long Sun, Zhun Sun, Houwen Peng, and Han-Jia Ye. Mitigating visual forgetting via take-along visual conditioning for multi-modal long cot reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eighth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024a. URL `https://openreview.net/forum?id=QWTCcxMpPA`.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui 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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=yfcpdY4gMP`.

Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Xiaotao Gu, Shiyu Huang, Bin Xu, Yuxiao Dong, Ming Ding, and Jie Tang. Lvbench: An extreme long video understanding benchmark, 2024b.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13484–13508, 2023.

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. *arXiv preprint arXiv:2406.18521*, 2024c.

Lai Wei, Liangbo He, Jun Lan, Lingzhong Dong, Yutong Cai, Siyuan Li, Huijia Zhu, Weiqiang Wang, Linghe Kong, Yue Wang, Zhuosheng Zhang, and Weiran Huang. Zooming without zooming: Region-to-image distillation for fine-grained multimodal perception. *arXiv preprint arXiv:2602.11858*, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24824–24837, 2022.

xAI. Grok-1.5 vision preview. xAI blog, April 2024. URL `https://x.ai/blog/grok-1.5v`.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*, 2024.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380, 2018.

Junyan Ye, Dongzhi Jiang, Jun He, Baichuan Zhou, Zilong Huang, Zhiyuan Yan, Hongsheng Li, Conghui He, and Weijia Li. Blink-twice: You see, but do you observe? a reasoning benchmark on visual perception. *arXiv preprint arXiv:2510.09361*, 2025.

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

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

Rosie Zhao, Anshul Shah, Xiaoyu Zhu, Xinke Deng, Zhongyu Jiang, Yang Yang, Joerg Liebelt, and Arnab Mondal. On robustness and chain-of-thought consistency of rl-finetuned vlms. *arXiv preprint arXiv:2602.12506*, 2026.

Yilun Zhao, Lujing Xie, Haowei Zhang, Guo Gan, Yitao Long, Zhiyuan Hu, Tongyan Hu, Weiyuan Chen, Chuhan Li, Junyang Song, Zhijian Xu, Chengye Wang, Weifeng Pan, Ziyao Shangguan, Xiangru Tang, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. Mmvu: Measuring expert-level multi-discipline video understanding, 2025. URL `https://arxiv.org/abs/2501.12380`.

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.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. Mlvu: Benchmarking multi-task long video understanding. *arXiv preprint arXiv:2406.04264*, 2024.

Appendices

Contents

A Preliminaries **16**
A.1 Reinforcement Learning with Verifiable Rewards for Vision-Language Models 16
A.2 Soft Adaptive Policy Optimization 16

B Related Work **17**
B.1 Vision-Language Models 17
B.2 Reinforcement Learning for Language and Vision-Language Models 17
B.3 Vision-Language Reasoning and Multi-Hop Reasoning 17
B.4 Scalable Data Synthesis for Model Training 18

C Detailed Failure-Mode Analysis **18**

D Additional Experimental Details and Results **19**
D.1 Experimental Setup 19
D.2 Error-Type Analysis 19
D.3 Compute Resources 20
D.4 Full-Suite Hop-Structure Ablation 21
D.5 Benchmark-Level Gains and Regressions 21

E Full Prompt for Multi-Hop Query Design **21**

F Full Prompt for Image Filtering **31**

A Preliminaries

A.1 Reinforcement Learning with Verifiable Rewards for Vision-Language Models

The reinforcement learning with verifiable rewards (RLVR) framework for vision-language models (VLMs) closely parallels that for large language models (LLMs), with the primary distinction being that RLVR for VLMs processes both an image and a text query as input to generate a textual chain-of-thought culminating in a verifiable answer prediction, whereas RLVR for LLMs operates solely on text queries. Specifically, RLVR for VLMs aims to maximize the following objective:

$$J(\pi) = \mathbb{E}_{(\boldsymbol{I},\boldsymbol{q},\boldsymbol{a})\sim\mathcal{D},\boldsymbol{o}\sim\pi(\cdot|\boldsymbol{I},\boldsymbol{q})}[R(\boldsymbol{o},\boldsymbol{a})], \tag{1}$$

$$\text{where} \quad R(\boldsymbol{o},\boldsymbol{a}) = \begin{cases} 1.0 & \text{if } \texttt{is_equivalent}(\boldsymbol{o},\boldsymbol{a}), \\ 0.0 & \text{otherwise.} \end{cases} \tag{2}$$

Here, $\boldsymbol{I}$, $\boldsymbol{q}$, and $\boldsymbol{a}$ denote the image, text query, and ground-truth answer, respectively, sampled from dataset $\mathcal{D}$, and $\boldsymbol{o}$ represents the response generated by policy π conditioned on $\boldsymbol{I}$ and $\boldsymbol{q}$.

A.2 Soft Adaptive Policy Optimization

Soft Adaptive Policy Optimization (SAPO, (Gao et al., 2025)) is introduced to mitigate the potential instability and inefficiency caused by hard clipping in prior RLVR algorithms such as GSPO (Zheng et al., 2025) and GRPO (Shao et al., 2024). Concretely, SAPO substitutes hard clipping with a temperature-controlled soft gate and optimizes the following objective for VLMs:

$$\mathcal{J}(\theta) = \mathbb{E}_{(\boldsymbol{I},\boldsymbol{q},\boldsymbol{a})\sim\mathcal{D},\{\boldsymbol{o}_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|\boldsymbol{I},\boldsymbol{q})}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\boldsymbol{o}_i|}\sum_{t=1}^{|\boldsymbol{o}_i|} f_{i,t}(r_{i,t}(\theta))\,\hat{A}_{i,t}\right], \tag{3}$$

$$\text{where} \quad r_{i,t}(\theta) = \frac{\pi_\theta(\boldsymbol{o}_{i,t} \mid \boldsymbol{I},\boldsymbol{q},\boldsymbol{o}_{i,<t})}{\pi_{\theta_{\text{old}}}(\boldsymbol{o}_{i,t} \mid \boldsymbol{I},\boldsymbol{q},\boldsymbol{o}_{i,<t})}, \qquad \hat{A}_{i,t} = \hat{A}_i = \frac{R_i - \text{mean}\big(\{R_j\}_{j=1}^{G}\big)}{\text{std}\big(\{R_j\}_{j=1}^{G}\big)}, \tag{4}$$

$$f_{i,t}(x) = \sigma(\tau_{i,t}(x-1)) \cdot \frac{4}{\tau_{i,t}}, \qquad \tau_{i,t} = \begin{cases} \tau_{\text{pos}}, & \text{if } \hat{A}_{i,t} > 0, \\ \tau_{\text{neg}}, & \text{otherwise.} \end{cases} \tag{5}$$

Here, i and j index the G responses sampled for the same image–query pair, and t is the token index within a sequence. The parameters of the currently trained policy and the old rollout policy are denoted by θ and θ_{old}, respectively. The temperatures for positive and negative tokens are τ_{pos} and τ_{neg}, respectively. Moreover, R_i follows Equation (2) for the i-th sample, and $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

B Related Work

B.1 Vision-Language Models

The integration of visual encoders with large language models has driven rapid progress in multimodal understanding. LLaVA (Liu et al., 2023b) introduced the visual instruction tuning paradigm by projecting visual features into the language model's embedding space, a design subsequently refined in LLaVA-1.5 (Liu et al., 2023a) with stronger projectors and higher-resolution inputs. This architectural template has been adopted and scaled by both open-source and proprietary efforts, including InternVL (Chen et al., 2024c,b), Qwen-VL (Bai et al., 2023, 2025b), GPT-4V (OpenAI, 2023), and Gemini (Gemini Team et al., 2023). Alongside these advances, diagnostic studies show that even strong VLMs remain vulnerable to object hallucination, visual illusion, and language-prior-driven errors (Rohrbach et al., 2018; Guan et al., 2024; Leng et al., 2024). Recent analyses further suggest that these failures can become more pronounced when multimodal reasoning chains grow longer and drift away from image-grounded evidence (Liu et al., 2025; Sun et al., 2025); in particular, VLMs can progressively ignore visual inputs over long CoT, to the point where removing the image only marginally degrades final accuracy (Sun et al., 2025). While these models achieve strong aggregate scores on diverse benchmarks, their visual perception can still become unreliable during multi-step reasoning. This is the bottleneck our work targets through synthesized multi-hop data.

B.2 Reinforcement Learning for Language and Vision-Language Models

Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) first demonstrated the effectiveness of reinforcement learning (RL)-based alignment for language models, using proximal policy optimization (PPO) (Schulman et al., 2017) as the default optimizer. More recently, reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024) has emerged as a powerful alternative to RLHF, eliminating the need for a learned reward model by using programmatically verifiable answers. DeepSeek-R1 (DeepSeek-AI, 2025) further showed that pure RL can induce strong chain-of-thought reasoning in LLMs, which has motivated parallel extensions to VLMs such as VLM-R1 (Shen et al., 2025) and TikArt (Ding et al., 2026). In parallel, the optimization toolkit has also evolved: GRPO (Shao et al., 2024) and GSPO (Zheng et al., 2025) use group-based advantage estimation with hard clipping, while SAPO (Gao et al., 2025) replaces hard clipping with a temperature-controlled soft gate for improved stability. Beyond algorithm design, recent mechanistic and robustness analyses suggest that RL's effects are selective rather than uniform. In LLMs, improvements are concentrated on a minority of high-entropy tokens (Wang et al., 2025); in VLMs, RL appears to primarily refine vision-to-reasoning alignment in mid-to-late layers rather than uniformly strengthening visual perception (Li et al., 2026), and RL-finetuned models can still exhibit weak visual grounding and over-reliance on textual cues (Zhao et al., 2026). In related multimodal reasoning settings, recent work has also begun to explicitly reward visually grounded reasoning traces, for example through visually grounded reinforcement finetuning for visual document reasoning (Ni et al., 2025) and evidence-grounded RL for video reasoning (Luo et al., 2025). Most existing RL and RLVR works remain tied to pre-existing task or benchmark data, including math- and science-oriented reasoning sets as well as task-specific multimodal datasets for documents or videos. In contrast, our pipeline generates a benchmark-agnostic proxy task designed to force repeated visual grounding and yield broad, generalizable improvements.

B.3 Vision-Language Reasoning and Multi-Hop Reasoning

Compositional visual reasoning has been studied through synthetic diagnostic datasets such as CLEVR (Johnson et al., 2017) and real-image benchmarks such as GQA (Hudson and Manning, 2019), with early architectural innovations including neural module networks (Andreas et al., 2016) and visual programming (Gupta and Kembhavi, 2023). In the language domain, multi-hop question

answering requires chaining evidence across multiple passages. This setting has been formalized by benchmarks such as HotpotQA (Yang et al., 2018). Chain-of-thought (CoT) prompting (Wei et al., 2022) and its multimodal extension (Zhang et al., 2023) have shown that eliciting step-by-step reasoning significantly improves both LLM and VLM performance. At the same time, several recent works argue that strong multimodal reasoning requires more than language-space CoT: it depends on fine-grained observation, stronger intermediate perception representations, repeated image inspection, or iterative revisiting of visual regions (Bigverdi et al., 2025; Ye et al., 2025; Jiang et al., 2025). Our work differs from prior visual reasoning research in three respects. (1) We formalize multi-hop vision-language reasoning with two complementary hop types: perception-level hops (switching between single-object and multi-object perception) and instance-chain hops ($A \rightarrow B \rightarrow C$ dependency chains), instead of treating it as a single-axis chain. (2) We use multi-hop reasoning as a *proxy task* to improve general VLM capabilities rather than as an end goal. (3) Our data is synthesized on real images at scale, which bridges the gap between synthetic diagnostics and manually curated benchmarks.

B.4 Scalable Data Synthesis for Model Training

Using strong models to generate training data for weaker ones has become a central scaling strategy. Self-Instruct (Wang et al., 2023) and Alpaca (Taori et al., 2023) pioneered LLM-generated instruction data, while ShareGPT4V (Chen et al., 2023) extended this idea to vision-language captions. Representative multimodal data synthesis pipelines additionally integrate vision foundation models. They often combine a VLM for object detection with SAM/SAM2 (Kirillov et al., 2023; Ravi et al., 2024) for instance segmentation. They may also use open-set detection approaches such as Grounding DINO (Liu et al., 2023c) to construct structured multi-hop queries from raw images. Closer in spirit, region-to-image distillation also internalizes an inference-time visual operation (test-time zooming) into training-time supervision, by synthesizing grounded targets from zoomed-in micro-regions and distilling them back to the full image, enabling fine-grained perception without test-time zooming (Wei et al., 2026). Our work shares this high-level idea of using structured, visually grounded synthetic data to internalize capabilities that would otherwise require more complex inference-time procedures, but targets long-CoT visual grounding through chained, instance-grounded queries rather than fine-grained perception. Unlike prior data synthesis efforts that aim to approximate the target task distribution, our pipeline generates a benchmark-agnostic proxy task designed to force repeated visual grounding for generalizable improvement.

The primary contribution of HopChain is a structured multi-hop RLVR task and data formulation that combines perception-level transitions, instance-chain dependencies, and numerically verifiable terminal answers. The OOD proxy-task framing specifies its training role, the synthesis and filtering pipeline operationalizes it at scale, and the two-scale, 24-benchmark study provides empirical validation.

C Detailed Failure-Mode Analysis

Long CoT reasoning places a much higher demand on visual grounding than short, single-step QA. To answer correctly, a VLM must repeatedly return to the image, recover the right object, attribute, relation, or numeric evidence at each step, and use that evidence to determine the next step of reasoning. Once any error appears at an intermediate step, the remaining reasoning can still look coherent while operating on flawed intermediate evidence, eventually producing an incorrect final answer. This is consistent with prior multimodal-reasoning analyses, which report that longer reasoning traces can drift away from image-grounded evidence, reduce attention to visual inputs, and amplify hallucinated intermediate content (Liu et al., 2025; Luo et al., 2025); more general diagnostic studies of VLMs have also documented persistent object hallucination and visual illusion under image-context reasoning (Rohrbach et al., 2018; Guan et al., 2024).

Figure 3 provides representative baseline failures: the baseline model (`RLVR w/o Multi-Hop`) miscounts the ladybirds' dots, follows the wrong arc in the astronomy diagram, interprets the car's parking sequence as leaving rather than entering a space, and incorrectly infers that the zebra has no horizontal stripes because vertical stripes dominate. In each case, a single faulty intermediate step contaminates the rest of the chain—a mistaken visual judgment can trigger faulty reasoning, unsupported inference, or other downstream failures.

This analysis suggests the central challenge is not merely producing a longer textual CoT, but maintaining reliable, step-wise reasoning over visual evidence throughout the chain.

D Additional Experimental Details and Results

D.1 Experimental Setup

For Qwen3.5-35B-A3B, each rollout samples 16 responses for each of 256 queries, uses a mini-batch size of 64 queries, and runs for 1000 gradient steps. Qwen3.5-397B-A17B follows the same rollout configuration in general, except that the mini-batch size is increased to 128 queries and the total number of gradient steps is 800. For both models, RLVR training uses the SAPO algorithm with a learning rate of 2.0×10^{-6}, while other hyperparameters follow the original SAPO paper (Gao et al., 2025).

Image Filtering. Before multi-hop query synthesis, we filter images to retain cases that are genuinely useful for long vision-language reasoning. The selection prompt in Appendix F is designed to favor images that are perceptually challenging for standard vision models, such as those involving occlusion, dense but still analyzable objects, unusual poses, complex interactions, fine-grained distinctions, or difficult lighting, while filtering out images that are too low-quality or too annotation-impractical to support verifiable reasoning. To balance quality and throughput, we use a two-stage pipeline. We first apply Qwen3-VL-235B-A22B-Thinking to a small subset of images with the image-selection prompt and keep the accepted images as the initial set of selected images. We then use this initial set of selected images together with the large model's outputs to perform supervised fine-tuning (SFT) on a smaller model, Qwen3-VL-30B-A3B-Thinking, specialized for image filtering. Next, we run this smaller model over all remaining images to obtain a coarse screening result. Finally, we send the images retained by the coarse screening back to Qwen3-VL-235B-A22B-Thinking for a second, finer filtering pass, yielding a second set of selected images. The union of the initial set of selected images and the second set of selected images forms the final pool of images used for multi-hop data construction. Based on the filtered image pool above, we construct multi-hop queries with HopChain and then filter out overly easy samples using a weaker model and the pre-RLVR model, yielding about $6k$–$8k$ multi-hop RLVR samples for each model. In addition, we use a similar amount of math RLVR data as the original RLVR data.

Benchmarks. We evaluate 24 benchmarks across 4 categories. For **STEM and Puzzle** reasoning, we use MathVision (Wang et al., 2024a), MMMU-Pro (Yue et al., 2025), MMMU (Yue et al., 2024), MathVista (Lu et al., 2024), BabyVision (Chen et al., 2026), ZeroBench (Roberts et al., 2025), EMMA (Hao et al., 2025), and LogicVista (Xiao et al., 2024). For **General VQA**, we use MMBench (Liu et al., 2023d), RealWorldQA (xAI, 2024), MMStar (Chen et al., 2024a), HallusionBench (Guan et al., 2024), AI2D (Kembhavi et al., 2016), and ERQA (Gemini Robotics Team et al., 2025). For **Text Recognition and Document Understanding**, we use CharXiv (Wang et al., 2024c), DocVQA (Mathew et al., 2021), and InfographicVQA (reported as `InfoVQA_VAL`) (Mathew et al., 2022). For **Video Understanding**, we use Video-MME (Fu et al., 2025), Video-MMMU (Hu et al., 2025), MMVU-CoT (Zhao et al., 2025), MVBench (Li et al., 2024), LVBench (Wang et al., 2024b), and MLVU (Zhou et al., 2024).

D.2 Error-Type Analysis

We further analyze whether multi-hop augmentation fixes only a narrow corner of the error space or yields broader gains. Figure 8 analyzes the original error types of the cases where `RLVR w/ Multi-Hop` corrects an error made by `RLVR w/o Multi-Hop`. Compared with the baseline error profile in Figure 2, the corrected-case distribution is broadly similar: perception errors remain the largest group, reasoning errors remain the second largest, and knowledge, hallucination, and other errors are also represented. This indicates that multi-hop training does not selectively repair one narrow failure mode while leaving the others untouched.

Perception subtypes. Within perception errors, the corrected cases span all major subtypes covered by the baseline profile, including chart misreads, text misreads, object misidentification, spatial errors, counting errors, color errors, and temporal misreads. This suggests that the gains are not concentrated

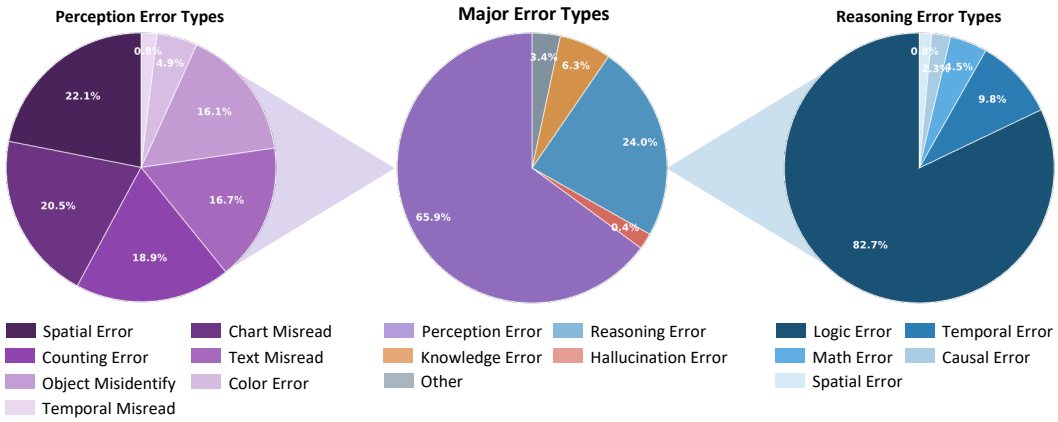


Figure 8: Error distribution of the original `RLVR w/o Multi-Hop` errors among the cases where `RLVR w/ Multi-Hop` improves over `RLVR w/o Multi-Hop` on the benchmarks in Tables 1 and 2. **Center pie chart:** the major error categories of the corrected cases, indicating which types of baseline errors are most often fixed after adding multi-hop data. **Left pie chart:** the subtype breakdown of corrected perception errors. **Right pie chart:** the subtype breakdown of corrected reasoning errors. With Figure 2, this figure shows that the multi-hop data synthesized by HopChain can improve models in broad, generalizable domains.

in a single perceptual axis, for example only OCR-style text misreads or only object identification, but extend to heterogeneous perceptual demands.

Reasoning subtypes. Within reasoning errors, the corrected cases similarly span logic, math, temporal, causal, and spatial reasoning, suggesting that the gains extend across multiple types of multi-step reasoning rather than only to logical chaining or arithmetic. Together with the broadly similar shapes of the corrected-case profile and the baseline profile in Figure 2, this is consistent with multi-hop data mitigating long-CoT failures across many error types rather than fixing one narrow type, in line with the broad benchmark-level gains we observe in Tables 1 and 2.

D.3 Compute Resources

All RLVR experiments are run on NVIDIA H200 GPUs. For Qwen3.5-35B-A3B, each RLVR training uses 64 H200 GPUs for approximately 60 hours (about 2 days 12 hours), corresponding to roughly 3.9k GPU-hours. For Qwen3.5-397B-A17B, each RLVR training uses 256 H200 GPUs for approximately 130 hours (about 5 days 10 hours), corresponding to roughly 33k GPU-hours. The four RLVR runs in Tables 1 and 2 (two per model scale) together require approximately 74k H200 GPU-hours. This estimate covers only the main-table RLVR runs and excludes the cost of obtaining the shared post-SFT starting models and the Half-Multi-Hop and Single Hop ablation runs. Compute used for preliminary or failed experiments is not reported here.

D.4 Full-Suite Hop-Structure Ablation

Table 3: Full/Half/Single ablation on Qwen3.5-35B-A3B across all 24 benchmarks, using the same training runs evaluated on the five representative benchmarks in Figure 5.

Benchmark	Full Multi-Hop	Half-Multi-Hop	Single Hop
STEM and Puzzle			
MathVision	**76.05**	73.03	72.76
MMMU-Pro	**70.64**	62.72	60.87
MMMU	**78.33**	75.67	74.33
MathVista (mini)	**85.00**	79.30	76.60
BabyVision	**22.68**	13.40	11.86
ZeroBench	**3.00**	2.00	1.00
EMMA (mini)	**58.00**	57.75	53.50
LogicVista	**75.56**	74.04	72.58
General VQA			
MMBench CN-DEV-V1.1	**90.48**	89.32	88.39
MMBench EN-DEV-V1.1	**91.49**	90.25	87.62
RealWorldQA	**79.35**	78.30	74.90
MMStar	**78.60**	73.73	71.93
HallusionBench	**66.50**	61.55	61.24
AI2D_TEST	**91.29**	83.65	81.28
ERQA	**51.38**	48.50	47.25
Text Recognition and Document Understanding			
CharXiv	**73.10**	50.20	45.40
DocVQA_VAL	**95.55**	68.19	68.46
InfoVQA_VAL	**90.17**	78.97	81.60
Video Understanding			
VideoMME w/o subtitles	**75.00**	74.30	74.00
VideoMMMU	**74.78**	70.78	65.78
MMVU-CoT	**68.90**	67.10	65.30
MVBench	**70.73**	70.13	67.63
LVBench	**53.20**	51.32	50.55
MLVU (M-Avg)	**79.53**	78.61	75.62
Unweighted average over 24 benchmarks	**70.80**	65.53	63.77

D.5 Benchmark-Level Gains and Regressions

Each model improves on 20 of 24 benchmarks and regresses on four. Seventeen benchmarks improve at both scales, whereas LVBench is the only benchmark that regresses at both scales, by 1.29 and 0.06 points for 35B and 397B, respectively. The remaining regressions occur only at 35B for MMMU, MathVista, and HallusionBench, and only at 397B for MMBench-EN, MMStar, and ERQA.

E Full Prompt for Multi-Hop Query Design

We provide the complete prompt used for synthesizing multi-hop vision-language reasoning queries below. Placeholders such as `{num_queries_word}`, `{object_list}`, `{num_queries}`, and `{target_hop_count_info}` are filled at runtime.

Multi-Hop Query Design Prompt (Full)

```
#### **Role & Goal**

You are a top-tier AI multimodal capability evaluation expert. Your task is to
    design a set of **{num_queries_word}** independent, high-difficulty **multi-
    hop vision-language reasoning** sub-queries based on a **complex image**.
```

```
    These sub-queries will serve as "capability modules" to evaluate a model's
    comprehensive visual intelligence.

#### **Understanding VLM Perception Capabilities**

VLM (Vision-Language Model) perception capabilities can be categorized into three
     levels:

1. **Single-Object Perception (Level 1):** Perceiving attributes of a single
    object (e.g., color, shape, size, text content, position, category).
2. **Multi-Object & Relationship Perception (Level 2):** Perceiving multiple
    objects and their relationships (e.g., spatial relationships like "A is to
    the left of B", comparative relationships like "A is larger than B",
    counting objects that satisfy certain conditions).
3. **Multi-Hop Reasoning (Level 3):** Chaining multiple perception steps together
    . **Multi-hop can occur in TWO dimensions:**
   - **Perception-Level Hops:** Moving between Level 1 and Level 2 perception
    tasks
   - **Instance-to-Instance Hops:** Reasoning from instance_1 -> instance_2 ->
    instance_3 -> ... -> instance_n, where **finding the next instance DEPENDS
    ON the previous instance**. The key is that instance_N+1 can ONLY be located
     by using information from or relationships with instance_N.

**YOUR TASK: Design queries that are strictly Level 3 (Multi-Hop Reasoning).**
    Each query must chain together multiple perception steps across BOTH
    dimensions - hopping between different perception levels AND hopping across
    different instances with DEPENDENCY relationships.

#### **Types of Multi-Hop Reasoning (Design queries using ALL types)**

**Type A - Instance Dependency Chain (MOST IMPORTANT):**
- The NEXT instance can ONLY be found by using its relationship with the PREVIOUS
     instance
- Each hop must establish a dependency: "the X that is [relationship] to the
    previous instance"
- Example: "Start from the red shirt -> find the chair that THIS shirt is draped
    over -> find the table that THIS chair is placed next to -> count objects on
     THIS table"
- Example: "Find person A -> locate the bag that A is holding -> identify the
    brand logo on THIS bag -> count similar logos in the image"
- **BAD (no dependency):** "Find the largest car, then find the tallest tree" (
    tree doesn't depend on car)
- **GOOD (with dependency):** "Find the largest car, then find the tree closest
    to THIS car" (tree depends on which car was found)

**Type B - Perception Level Hop:**
- Level 1 (single object) -> Level 2 (multi-object relationship) -> Level 1 ->
    Level 2 -> ...
- Each level transition should maintain the instance dependency chain
- Example: "Read the number on sign A (L1), find objects to the right of THIS
    sign (L2), among THESE objects find the closest to B (L2), read text on THIS
     object (L1)"

**Type C - Combined with Strong Dependencies (PREFERRED):**
- Combine instance dependency chains with perception level changes for maximum
    complexity
- Every instance transition must have a clear dependency relationship
- Example: "Find shirt A -> locate the chair THIS shirt is draped on -> find the
    person sitting on THIS chair -> read the text on the book THIS person is
    holding -> count pages visible in THIS book"

**You are provided with MULTIPLE images:**
```

```
1. **Image 1 (Original):** The original image containing all visual information.
    **This is the ONLY image that will be available when answering the queries.**

2. **Instance Patch Images (Image 2, 3, 4, ...):** Cropped patches of each
    individual instance from the original image. Each patch shows one specific
    instance extracted using its bounding box. These patches help you precisely
    identify each instance's appearance and content. **These patches are ONLY
    for your reference during question design and will NOT be available when
    answering queries.**

**CRITICAL: You are designing queries for a SPECIFIC COMBINATION of objects. The
    following object instances are the ONLY instances you should consider for
    this task. Each instance is provided as a separate cropped patch with its
    bounding box location in the original image (coordinates in 0-1000 range):**

{object_list}

**[WARNING] MANDATORY REQUIREMENT - INVOLVE ALL INSTANCES:**
- **Each sub-query MUST involve AS MANY instances as possible from the
    combination list above.**
- **Ideally, every query should reference or require reasoning about ALL or MOST
    of the listed instances.**
- If the combination has N instances, each query should involve at least (N-1)
    instances, preferably all N instances.
- The 'involved_objects' field in your output MUST list ALL instances that are
    referenced or reasoned about in that query.
- **DO NOT design queries that only involve 2-3 objects when 5-6 objects are
    available in the combination.**

**CRITICAL CONSTRAINT - NO PATCH/BOX REFERENCES & UNAMBIGUOUS INSTANCE REFERENCES
    :**
- The cropped patches and bounding box coordinates are **ONLY** visual aids to
    help you understand which object is which during question design.
- **When writing the actual query text, you MUST NEVER mention:**
  - Detection borders, bounding boxes, patches, or any detection-related visual
    markers
  - Bounding box coordinates or positions in the 0-1000 coordinate system
  - Any reference to the cropped patch images or instance visualization aids
- **You MUST describe objects using ONLY:**
  - Spatial positions (e.g., "the object on the left", "the topmost item")
  - Contextual relationships (e.g., "the object next to the table", "the item
    above the text")
  - Functional descriptions (e.g., "the button", "the icon", "the label")
  - Visual attributes from the original image itself (e.g., actual colors of
    objects, sizes, shapes)
- **[WARNING] CRITICAL: UNAMBIGUOUS INSTANCE REFERENCES (MOST IMPORTANT):**
  - **Each instance reference in your query MUST uniquely identify exactly ONE
    instance in the original image (Image 1) without any borders.**
  - **Test each reference:** Imagine the query is given to someone looking ONLY
    at Image 1 (no colored borders). Can they identify the EXACT instance you
    mean without any confusion?
  - **BAD (ambiguous):** "the car" when there are multiple cars in the image
  - **BAD (ambiguous):** "the person on the right" when there are two people on
    the right side
  - **BAD (ambiguous):** "the red object" when multiple objects are red
  - **GOOD (unambiguous):** "the car closest to the bottom-left corner"
  - **GOOD (unambiguous):** "the person wearing a blue shirt standing next to the
     lamppost"
  - **GOOD (unambiguous):** "the largest red object that is positioned above the
    text area"
  - **GOOD (unambiguous):** "the bicycle located between the tree and the bench"
  - **Use multiple discriminating features:** Combine spatial position + visual
    attributes + contextual relationships to ensure uniqueness
```

```
  - **Verify uniqueness:** For each instance reference, mentally check if ANY
    other object in Image 1 could match that description. If yes, add more
    distinguishing details.
- **Reason:** The final questions will be evaluated on Image 1 (the original
    image) **without any borders, markers, or detection aids**. Any ambiguous
    reference will make the question unsolvable or lead to multiple valid
    interpretations.

#### **Core Principles: Requirements for Each Sub-query**

**Every** sub-query you design must strictly adhere to all of the following
    standards:

1.  **Independence:** Each query must be completely self-contained. Solving it
    must not depend on information or answers from any other query.

2.  **[VERY IMPORTANT] MANDATORY Multi-Hop Structure (MOST CRITICAL):**
    - **Each query MUST be a genuine multi-hop reasoning problem with {
    target_hop_count_info}.**
    - **Number of hops = {target_hop_count_info}.** Push the boundaries of
    complexity within this range.
    - Each hop can be:
      - **Level 1 (Single-Object Perception):** Extract information from one
    specific object (e.g., read text, identify color, determine position)
      - **Level 2 (Multi-Object Relationship):** Reason about relationships
    between multiple objects (e.g., compare sizes, count objects satisfying a
    condition, determine spatial arrangement)
    - **Instance-to-Instance Chaining is REQUIRED:** The reasoning should hop
    from instance_1 -> instance_2 -> instance_3 -> ... involving as many
    instances as possible in a chain.
    - **The hops must be logically chained:** The result of each hop must be
    REQUIRED for the next hop. No hop should be skippable.
    - **Example of a HIGH-QUALITY multi-hop structure (5 hops, all instances
    involved):**
      - Hop 1 (L1 on A): Read the number on object A -> get value X
      - Hop 2 (L2 on A,B,C): Find objects to the right of A that are smaller than
     B -> get set S containing C
      - Hop 3 (L1 on C): Identify the color of object C (from set S) -> get color
     Y
      - Hop 4 (L2 on D,E,F): Count objects among D,E,F that share color Y -> get
    count Z
      - Hop 5 (L2 on result): Compute X x Z and compare with the area ratio of E
    to F -> final answer
    - **INVALID (too few hops):** "How many red objects are there?" (1 hop)
    - **INVALID (too simple):** "What is the color of the largest object?" (2
    hops at most)
    - **INVALID (no instance chain):** Query that only touches 2 instances

3.  **[IMPORTANT] Maximum Instance Coverage (CRITICAL):**
    - **Each query MUST involve ALL instances from the provided combination list
    - no exceptions.**
    - Design queries where EVERY instance plays a meaningful role in the
    reasoning chain.
    - **Good pattern:** "Starting from A, find the instance closest to it (hop to
     B), then find instances sharing B's color (hop to C,D), compare their sizes
     with E (hop to E), and count those larger than F (hop to F) -> final answer
     uses info from ALL instances"
    - **Bad pattern:** "Compare object 1 and object 2" (ignores other instances)

4.  **[IMPORTANT] High Perception & Logic Difficulty (CRITICAL):**
    - **Perception difficulty:** Each hop should require non-trivial perception:
      - Reading small or partially occluded text
      - Distinguishing subtle color differences
```

```
  - Determining complex spatial relationships (e.g., "between", "surrounded
by", "aligned with")
  - Counting objects under multiple constraints
- **Logic difficulty:** The reasoning chain should require:
  - Conditional logic (if-then-else based on visual observations). **
IMPORTANT: Ensure that the "if" conditions are NOT always true. Aim for a
balanced and random distribution of "Yes" and "No" outcomes for these
conditions across different queries and hops to avoid bias.**
  - Set operations (intersection, union, difference of object sets)
  - Numerical computations (ratios, differences, products)
  - Comparative reasoning with multiple criteria
- **Avoid trivial hops** like "identify that this is a car" - instead require
 "identify the car that is closest to the pedestrian and has a license plate
 starting with a letter"

5. **Deterministic Numerical Answer:** The final answer to each query must be a
specific, unambiguous **number**.

6. **Visual Grounding:** All information required to solve the query must be
obtainable **only** from the image and must not rely on external knowledge.

7. **Avoid "Shortcut" Hints:** Strictly prohibit using low-level visual features
 (e.g., "the red text," "the largest icon") to locate information. Guidance
should be provided through the object's **function, context, or logical
relationships**.

8. **Avoid Visual Information Leakage:** The wording of the query **must not
contain** any specific text or numbers that the model would need to acquire
from the image via visual recognition (e.g., OCR). This ensures the model
finds the answer by understanding the image content, not through text
matching.

9. **Unambiguous Phrasing & Deterministic Solution:**
- **The query itself must be phrased precisely, allowing for only one logical
, unambiguous interpretation within the context of the image.**
- **The solution process must be completely deterministic with no room for
dispute:**
  - Every instance reference must uniquely identify exactly one object (see
CRITICAL CONSTRAINT above)
  - Every spatial relationship must have a clear, unambiguous definition (e.g
., "to the left of" means the object's center is to the left of the
reference object's center)
  - Every comparison criterion must be objective and measurable (e.g., "
larger" means larger area, not perceived importance)
  - Every counting task must have clear inclusion/exclusion criteria
  - Every numerical operation must be well-defined
- **BAD (ambiguous solution):** "Count objects near the tree" - what distance
 counts as "near"?
- **GOOD (deterministic):** "Count objects whose center is within a distance
of twice the tree's width from the tree's center"
- **BAD (ambiguous):** "Find the most prominent object" - subjective judgment
- **GOOD (deterministic):** "Find the object with the largest area"
- **Verification test:** Two different people with good vision looking at
Image 1 should arrive at the EXACT SAME numerical answer through the EXACT
SAME reasoning steps.

10. **Diversity of Reasoning:** For a set of sub-queries designed for the same
image, avoid using the same reasoning logic or template. Each sub-query
should test different reasoning paths, hop combinations, and instance chains.
 **This includes diversifying the outcomes of conditional steps (mixing "if
true" and "if false" paths) so that the correct answer is neither always "
Yes" nor always "No".**
```

```
11. **Capability Coverage:** The {num_queries_word} queries you design, **as a
    whole**, should cover the following core capability dimensions:
    *   **Color Comprehension**
    *   **Spatial Reasoning**
    *   **Counting & Aggregation**
    *   **Text & Symbol Recognition**
    *   **Logical & Conditional Inference**

12. **No Detection Border/Color References:** **STRICTLY FORBIDDEN** to mention
    detection borders, border colors, bounding boxes, or any detection-related
    visual markers in the query text.
13. **Balanced Conditional Outcomes (CRITICAL):** For queries involving
    conditional logic (if-then-else), you MUST ensure a balanced and random
    distribution of "Yes" and "No" outcomes for the conditions. **Do not make
    the "if" condition always true or always false.** Across all queries and
    hops, roughly half of the conditional checks should evaluate to "No" (
    triggering the 'else' or 'otherwise' path). For example, if an object is red,
     you might design a condition like "if this object is blue, add 5; otherwise,
     add 2", forcing the reasoning to follow the "otherwise" path. This prevents
     the model from developing a bias and ensures it actually performs the
    visual verification.

#### **Workflow & Output Format**

Your final output should be a **single JSON object**. This object contains an
    array named `sub_queries`, with {num_queries_word} elements, each
    representing a meticulously designed sub-query.

**IMPORTANT: The `involved_objects` field MUST list ALL instances from the
    combination that are referenced in the query. Aim to involve ALL available
    instances in each query.**

Please strictly follow the JSON structure below:

"`json
{{
  "sub_queries": [
    {{
      "id": 1,
      "primary_capability": "Spatial Reasoning + Counting + Conditional Logic",
      "involved_objects": ["instance_1", "instance_2", "instance_3", "instance_4
    ", "instance_5"],
      "query": "Your complex multi-hop query here. It should naturally chain
    through multiple instances: Start from A, hop to B based on some
    relationship, then hop to C, then to D, and finally involve E to compute the
     answer.",
      "instance_chain": "A -> B -> C -> D -> E (describe how reasoning flows
    between instances)",
      "reasoning_hops": [
        {{
          "hop_number": 1,
          "hop_type": "Level 1 (Single-Object)",
          "from_instance": "instance_1",
          "to_instance": null,
          "description": "Extract specific information from instance_1",
          "objects_involved": ["instance_1"],
          "output": "value_X extracted from instance_1"
        }},
        {{
          "hop_number": 2,
          "hop_type": "Level 2 (Multi-Object Relationship)",
          "from_instance": "instance_1",
          "to_instance": "instance_2",
```

```
      "description": "Use value_X to find/filter among other instances,
identifying instance_2",
      "objects_involved": ["instance_1", "instance_2", "instance_3"],
      "output": "instance_2 identified based on relationship with instance_1"
    }},
    {{
      "hop_number": 3,
      "hop_type": "Level 2 (Multi-Object Relationship)",
      "from_instance": "instance_2",
      "to_instance": "instance_4",
      "description": "From instance_2, determine relationship with instance_4
",
      "objects_involved": ["instance_2", "instance_4"],
      "output": "relationship_result between instance_2 and instance_4"
    }},
    {{
      "hop_number": 4,
      "hop_type": "Level 1 (Single-Object)",
      "from_instance": "instance_4",
      "to_instance": null,
      "description": "Extract value from instance_4",
      "objects_involved": ["instance_4"],
      "output": "value_Y from instance_4"
    }},
    {{
      "hop_number": 5,
      "hop_type": "Level 2 (Multi-Object Relationship)",
      "from_instance": "instance_4",
      "to_instance": "instance_5",
      "description": "Compare or compute using instance_5 as reference",
      "objects_involved": ["instance_4", "instance_5"],
      "output": "final_numerical_answer"
    }}
  ],
  "hypothetical_answer": "42",
  "design_rationale": "Explain: (1) The instance chain A->B->C->D->E, (2) Why
 each hop is necessary, (3) What makes the perception/logic difficult"
}},
{{
  "id": 2,
  "primary_capability": "Text Recognition + Logical Inference + Conditional",
  "involved_objects": ["ALL instances - list them all"],
  "query": "Another complex multi-hop query with instance chain...",
  "instance_chain": "Describe the instance flow",
  "reasoning_hops": [
    {{"hop_number": 1, "hop_type": "L1", "from_instance": "...", "to_instance
": "...", "description": "...", "objects_involved": [...], "output": "..."}},

    {{"hop_number": 2, "hop_type": "L2", "from_instance": "...", "to_instance
": "...", "description": "...", "objects_involved": [...], "output": "..."}},

    {{"hop_number": 3, "hop_type": "...", "from_instance": "...", "
to_instance": "...", "description": "...", "objects_involved": [...], "
output": "..."}},
    {{"hop_number": 4, "hop_type": "...", "from_instance": "...", "
to_instance": "...", "description": "...", "objects_involved": [...], "
output": "..."}}
  ],
  "hypothetical_answer": "150",
  "design_rationale": "..."
}},
{{
  "id": 3,
  "primary_capability": "Counting & Aggregation + Spatial + Set Operations",
```

```
      "involved_objects": ["ALL instances"],
      "query": "Complex multi-hop query...",
      "instance_chain": "...",
      "reasoning_hops": ["... {target_hop_count_info} ..."],
      "hypothetical_answer": "24",
      "design_rationale": "..."
    }},
    {{
      "id": 4,
      "primary_capability": "Color + Comparison + Conditional Branching",
      "involved_objects": ["ALL instances"],
      "query": "Complex multi-hop query...",
      "instance_chain": "...",
      "reasoning_hops": ["... {target_hop_count_info} ..."],
      "hypothetical_answer": "3",
      "design_rationale": "..."
    }},
    {{
      "id": {num_queries},
      "primary_capability": "Comprehensive: All Capabilities Combined",
      "involved_objects": ["ALL instances"],
      "query": "Most complex multi-hop query combining all capability types with
    longest instance chain...",
      "instance_chain": "...",
      "reasoning_hops": ["... {target_hop_count_info} ..."],
      "hypothetical_answer": "1800",
      "design_rationale": "..."
    }}
  ]
}}
"'

#### **Examples of Well-Designed Multi-Hop Queries**

Suppose the combination contains: person_1, car_1, bicycle_1, sign_1, tree_1

---

**EXCELLENT Example (5 hops, instance chain, all 5 objects, high difficulty,
    UNAMBIGUOUS references):**

**Context:** Assume in the image: sign is in top-right corner, car is red sedan
    in center, bicycle is yellow near bottom, person wearing blue shirt in left,
     tree is large with green leaves in right background.

Query (with unambiguous references): "Starting from the traffic sign located in
    the top-right corner of the image, read the speed limit number displayed on
    it. Then find which vehicle - either the red car in the center or the yellow
     bicycle near the bottom edge - has its center point closest to the sign's
    center. Check whether that vehicle's center is closer to the person wearing
    the blue shirt on the left side, or to the large tree with green leaves in
    the right background. If it's closer to the person, count all objects whose
    centers lie on the straight line connecting the person and the tree; if
    closer to the tree, count all objects positioned to the left of the tree's
    leftmost edge. Finally, multiply the speed limit number by this count, then
    subtract the total number of wheels visible on both vehicles (the red car
    and the yellow bicycle). What is the result?"

- **Hop 1** (L1 -> sign_1): Read number from "traffic sign in top-right corner"
    -> speed_limit = 30
- **Hop 2** (L2 -> sign_1, car_1, bicycle_1): Compare distances: "red car in
    center" vs "yellow bicycle near bottom" to sign -> closest = car_1
```

```
- **Hop 3** (L2 -> car_1, person_1, tree_1): Compare distances: car's center to "
    person in blue shirt on left" vs "large tree in right background" -> closer
    to person_1
- **Hop 4** (L2 -> person_1, tree_1, car_1, bicycle_1): Count objects on line
    between "person in blue" and "tree in right" -> count = 2 (car and bicycle)
- **Hop 5** (L1+L2 -> car_1, bicycle_1): Count wheels: "red car" (4) + "yellow
    bicycle" (2) -> wheels = 6
- **Final**: 30 x 2 - 6 = 54

**Why this is unambiguous:**
- Every instance has multiple distinguishing features (color + position +
    distinctive attributes)
- "Red car in center" - unique by color and position
- "Yellow bicycle near bottom" - unique by color and position
- "Person wearing blue shirt on left" - unique by clothing and position
- "Traffic sign in top-right corner" - unique by function and position
- "Large tree with green leaves in right background" - unique by size, color,
    position

Instance Chain: sign_1 -> car_1 -> person_1/tree_1 -> car_1,bicycle_1 -> final
involved_objects: ["person_1", "car_1", "bicycle_1", "sign_1", "tree_1"] [OK] ALL
     5!

---

**GOOD Example (4 hops, instance chain, all objects):**

Query: "Find the object that is furthest from the person. Then identify all
    objects that share at least one visual attribute (color, shape, or size
    category) with it. Among those objects, find the one closest to the bicycle.
     Read any text or number visible on or near that object. If no text exists,
    use the count of objects to its left. What is this value?"

- **Hop 1** (L2 -> person_1, all others): Find furthest object from person_1 ->
    sign_1
- **Hop 2** (L2 -> sign_1, car_1, bicycle_1, tree_1): Find objects sharing
    attributes with sign_1 -> result set contains tree_1 (both have vertical
    structure)
- **Hop 3** (L2 -> tree_1, bicycle_1): Find which object in result set is closest
     to bicycle_1 -> tree_1
- **Hop 4** (L1 -> tree_1): Check for text on tree_1, if none, count objects to
    its left -> 2

Instance Chain: person_1 -> sign_1 -> tree_1 -> bicycle_1 -> tree_1 -> final
involved_objects: ["person_1", "car_1", "bicycle_1", "sign_1", "tree_1"] [OK] ALL
     5!

---

**BAD Example (too few hops, too simple):**
- Query: "What color is the car?"
- Problems: Only 1 hop, only 1 object, trivial perception

**BAD Example (few objects, weak chain):**
- Query: "Read the number on the sign and add it to the count of people."
- Problems: Only 2 objects involved, only 2 hops, no instance chain

**BAD Example (hops not logically dependent):**
- Query: "Count red objects and count blue objects. What is the sum?"
- Problems: The two counts are independent (can be done in parallel), not a true
    chain

**BAD Example (AMBIGUOUS instance references - CRITICAL FAILURE):**
- Query: "Find the car, then count objects to its right."
```

```
- Problems:
  - "the car" - which car? If there are 2+ cars, this is ambiguous
  - "to its right" - to the right of what point? Edge? Center? Entire bounding
    region?
  - Cannot be solved deterministically without borders

**BAD Example (SUBJECTIVE/NON-DETERMINISTIC):**
- Query: "Find the most important object, then count nearby objects."
- Problems:
  - "most important" - subjective, not measurable
  - "nearby" - no clear distance threshold defined
  - Two people would give different answers

**GOOD Fix (making it unambiguous and deterministic):**
- Query: "Find the red sedan car located in the center-bottom area of the image (
    the only car with visible license plate), then count all objects whose
    center points are within a horizontal distance of 100 pixels to the right of
     this car's rightmost edge."
- Why it works:
  - "Red sedan car in center-bottom with visible license plate" - multiple
    features ensure uniqueness
  - "Center points" - clear reference point
  - "Horizontal distance of 100 pixels to the right of rightmost edge" - precise,
     measurable criterion
  - Completely deterministic

#### **Final Checklist Before Outputting**

For EACH of your {num_queries_word} sub-queries, verify:
- [ ] **Hop Count:** Does it have {target_hop_count_info}?
- [ ] **Instance Chain:** Does the reasoning hop from instance to instance (A->B
    ->C->...)?
- [ ] **Full Coverage:** Does `involved_objects` contain ALL instances from the
    combination?
- [ ] **Logical Dependency:** Is the result of each hop REQUIRED for the next hop
    ? (no parallel/skippable hops)
- [ ] **Perception Difficulty:** Does each hop require non-trivial perception (
    not just "identify object type")?
- [ ] **Logic Complexity:** Does the chain include conditionals, set operations,
    or multi-step computations?
- [ ] **Numerical Answer:** Is the final answer a specific, computable number?
- [ ] **No Border References:** Does it avoid mentioning detection borders or
    visualization colors?
- [ ] **[VERY IMPORTANT] Unambiguous Instance References (CRITICAL):** For EACH
    instance reference in the query, verify:
  - Can someone looking ONLY at the original image (no borders) uniquely identify
     this exact instance?
  - Would ANY other object in the image match this description? If yes, add more
    distinguishing details.
  - Does the description use multiple discriminating features (spatial + visual +
     contextual)?
- [ ] **[VERY IMPORTANT] Deterministic Solution Process (CRITICAL):** Verify:
  - Would two different people arrive at the EXACT SAME answer?
  - Are all spatial relationships clearly defined? (e.g., "left of" = center is
    to the left)
  - Are all comparisons objective? (e.g., "larger" = larger area, not importance)
  - Are all counting criteria explicit? (what counts as "between", "near", etc.)
  - Is there NO room for subjective interpretation?

**Quality Scoring (aim for 10+):**
- 1 point per hop
- 1 point if ALL instances involved
- 1 point if instance chain covers 4+ different instances
- 1 point if includes conditional logic (if-then)
```

```
- 1 point if includes numerical computation beyond counting
- **2 points if ALL instance references are unambiguous and solution is
    deterministic (CRITICAL)**

**Please Note:**
- Your task is to generate the JSON content that conforms to the format and
    requirements above.
- **CRITICAL REMINDERS:**
  - Each ‘query‘ field must be answerable using ONLY the original image (Image 1)
     without any detection borders or markers.
  - **EVERY query MUST be multi-hop with {target_hop_count_info}.**
  - **EVERY query MUST involve ALL or MOST of the instances in the provided
    combination list.**
  - The ‘involved_objects‘ field must accurately list all instances referenced in
     the query.
```

F Full Prompt for Image Filtering

We also provide the complete prompt used to screen images before multi-hop query synthesis.

Image Filtering Prompt (Full)

```
# Role and Goal

You are a professional AI Image Analyst. Your primary goal is to evaluate the
    complexity of a given image from the perspective of a standard computer
    vision model. This analysis will identify images that pose a challenge to AI
     perception, especially in tasks like object detection, counting, and
    recognition.

# Core Task

For the image I provide, you must:

1.  Evaluate its overall perceptual complexity on a scale of **1 to 10**.
2.  Assess the image quality (High / Medium / Low).
3.  Identify the specific objects or areas that contribute to the complexity.
4.  Output the results as a **single JSON object** as required, with no
    explanatory text outside of the JSON.

# Key Definitions

## 1. Image Complexity Factors (What makes an image complex for AI?)

*   **Occlusion**: Objects are partially hidden by other objects. The more a key
    object is occluded, the higher the complexity.
*   **Object Count & Density**: The image contains a large number of distinct
    objects, especially of the same category (e.g., a crowd of people, a fleet
    of cars), making counting and individual identification difficult.
*   **Unusual Pose/Angle**: Objects are shown from unconventional viewpoints (e.g
    ., a person doing a handstand, a car seen directly from above).
*   **Complex Interaction**: Objects interact in a way that blurs their
    individual boundaries (e.g., people hugging, tangled wires, a pile of
    clothes).
*   **Fine-grained Recognition**: Requires distinguishing between very similar
    sub-categories (e.g., telling two different bird species apart, or different
     models of the same car brand).
*   **Challenging Lighting/Shadows**: Harsh shadows, reflections, or poor
    lighting obscure the shape and details of objects.

## 2. Image Quality Guidelines (How to handle low-quality or unusable images)
```

````
Your analysis must focus on clear, perceptible objects.

*   **High/Medium Quality**: The image is generally clear, and its main subjects
    are recognizable, even if the scene is complex.
*   **Low Quality**: An image should be rated "Low Quality" when it is unusable
    for precise perceptual analysis due to any of the following reasons:
    *   **Technical Flaws:** The image is too blurry, noisy, severely overexposed
    , or underexposed, making objects unrecognizable.
    *   **Annotation Impracticality:** **Even if the image itself is clear, the
    extreme number and density of objects make meaningful, verifiable annotation
     impossible (e.g., it's impossible to draw a bounding box for or accurately
    count every single object). Such images lack a "Ground Truth" for model
    evaluation.**

*   **Crucial**:
    *   **Case 1 (Technical Flaws):** If a scene is chaotic due to blur or
    lighting issues, making it impossible to distinguish individual objects, its
     quality should be rated **"Low"** with a **low complexity score (1-3)**.
    *   **Case 2 (Annotation Impracticality):** **If a scene contains a vast,
    uncountable collection of objects (e.g., a massive but clear crowd where
    individuals cannot be boxed or counted; a bookshelf or rack packed with
    countless small items), its quality should also be rated "Low," even if the
    image is technically clear.**
    *   **Conclusion:** **For both of the "Low Quality" cases described above,
    the perceptual complexity score should be low (1-3)**, as they are not good
    test cases for perception but rather issues of data usability or quality
    itself.

# Output Format Requirements

The output must be a structured **JSON object** (with no additional text). The
    structure is as follows:

```json
{
  "overall_complexity_score": <Integer, 1-10>,
  "overall_quality_rating": "<High | Medium | Low>",
  "complexity_analysis": "<String, a brief explanation for the score, referencing
     the complexity factors above>",
  "complex_objects": [
    {
      "object_name": "<Description of the specific object>",
      "generalized_name": "<General category like Person, Animal, Vehicle, Plant,
     etc.>",
      "reason_for_complexity": [
        "<Select from the list of complexity factors, e.g., 'Occlusion', 'Fine-
    grained Recognition'>"
      ]
    }
  ]
}
```

# Final Instruction

After I provide an image, strictly follow all the rules above, perform the
    analysis based on the actual image content, and output a JSON object that
    strictly adheres to the specified format.
````

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 accurately summarize our main contributions: (i) identifying diverse compounding failure modes in long-CoT vision-language reasoning, (ii) proposing HopChain as a scalable framework for synthesizing multi-hop RLVR data with logically dependent instance-grounded hops, and (iii) showing 20 out of 24 benchmark improvements on both Qwen3.5-35B-A3B and Qwen3.5-397B-A17B, with reported analyses on hop-structure ablation, long-CoT robustness, difficulty coverage, and error-type correction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The Limitations and future work paragraph in Section 5 explicitly states that the current pipeline depends on successful instance segmentation and excludes images with no SAM3-segmentable instances; we also propose complementary data-construction routes for such images as future work.

   Guidelines:

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

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not include theoretical results; all contributions are empirical (data synthesis pipeline and RLVR training results).

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Section 3 and Appendix D fully describe the four-stage pipeline (category identification with Qwen3-VL-235B-A22B-Thinking, instance segmentation with SAM3, multi-hop query generation with the same VLM, and human-in-the-loop annotation/calibration), the constraints applied to each query, the RLVR training settings (SAPO algorithm, learning rate, rollout configuration, mini-batch sizes, and gradient steps for both model scales), and all 24 evaluation benchmarks. The full prompts used for multi-hop query design and image filtering are provided in Appendices E and F.

   Guidelines:

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

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: We do not release code or the synthesized multi-hop data with this submission. The pipeline, hyperparameters, training schedule, benchmark list, and full prompts are described in detail in the paper and appendix to support reproducibility.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix D.1 reports the complete training configuration: rollout samples 16 responses for each of 256 queries, mini-batch sizes (64 for Qwen3.5-35B-A3B, 128 for Qwen3.5-397B-A17B), 1000/800 gradient steps respectively, SAPO learning rate of 2.0×10^{-6}, with other SAPO hyperparameters following the original SAPO paper. Benchmark families and the 24 specific benchmarks are listed in Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.

- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: We report single-evaluation accuracy on each benchmark in Tables 1 and 2 without error bars, following the convention of the underlying benchmark suites. For difficulty calibration (Figure 7), we instead independently sample eight responses per query and report the success-rate distribution across queries, which captures variability across multiple samples. Multiple runs of full RLVR training were not feasible due to compute constraints at the model scales used.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix D.3 reports the GPU type (NVIDIA H200), GPU count (64 for Qwen3.5-35B-A3B, 256 for Qwen3.5-397B-A17B), wall-clock time ($\sim$60 and $\sim$130 hours, respectively), and the corresponding total GPU-hours ($\sim$3.9k and $\sim$33k per RLVR run, respectively). The four RLVR runs in Tables 1 and 2 (two per model scale) total approximately 74k H200 GPU-hours; this total excludes the cost of obtaining the shared post-SFT starting models and the Half-Multi-Hop and Single Hop ablation runs. It also notes that compute used for preliminary or failed experiments is not reported here.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research was conducted in accordance with the NeurIPS Code of Ethics. Our synthesized multi-hop queries are generated from images filtered for analyzable visual content and verified by trained annotators; we do not use any data with privacy- or identity-sensitive content as central training material.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work proposes a general-purpose data-synthesis framework that improves long-CoT vision-language reasoning of VLMs across standard benchmarks; it does not target a specific deployed application and does not introduce new dual-use capabilities beyond those already present in the underlying foundation VLMs. Potential broader impacts are similar in nature to those of other generic VLM-improvement methods.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

Justification: The paper does not release pre-trained models, image generators, or scraped datasets with high misuse risk. The synthesized multi-hop data consists of numerical-answer reasoning questions over filtered images and does not pose elevated misuse risk relative to its underlying source images.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All foundation models, segmentation models, RLVR algorithms, and evaluation benchmarks used in the paper are cited with the original publications (e.g., Qwen3-VL family, SAM3, SAPO, GRPO/GSPO, and the 24 benchmarks listed in Appendix D). We use these assets in accordance with their respective licenses and terms of use.

Guidelines:

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

13. **New assets**

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

Answer: [N/A]

Justification: We do not release code, synthesized multi-hop data, or model artifacts with this paper.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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

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

    Answer: [N/A]

    Justification: Stage 4 query verification used model-assisted annotation, with manual adjudication by trained annotators for queries without a three-of-four majority; no crowdsourced participants or external study subjects were involved. Annotators were compensated according to standard annotation-work arrangements; no human-subjects research was conducted.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects; the data annotation step was performed as a standard annotation task and does not constitute human-subjects research that would require IRB approval.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: Large vision-language models (Qwen3-VL-235B-A22B-Thinking and Qwen3-VL-30B-A3B-Thinking) are used as core components of our data-synthesis pipeline for category identification, multi-hop query generation, and image filtering, as described in Section 3 and Appendix D. Model-assisted initial answer annotation is also used in Stage 4, as described in Section 3.3.

Guidelines:

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