CONTEXT-AWARE RL FOR AGENTIC AND MULTIMODAL LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) often fail when answering requires identifying a small but decisive piece of evidence within a long or complex context, such as a single line in a tool trace or a subtle detail in an image. We propose CONTEXTRL, a context-aware reinforcement learning (RL) method that improves long-horizon reasoning and multimodal performance through an *indirect* auxiliary objective. Instead of supervising only the final answer, CONTEXTRL presents the model with a query, an answer, and two highly similar contexts, and rewards it for selecting the context that supports the query–answer pair, thereby encouraging fine-grained grounding. We construct contrastive context data in two domains: for coding agents, trajectories serve as contexts, yielding 1k pairs built via condition filtering; for multimodal reasoning, images serve as contexts, yielding 7k pairs built via generative editing and similarity search. CONTEXTRL achieves average gains of +2.0% over standard GRPO on 5 long-horizon benchmarks, and +1.8% across 12 diverse visual question answering benchmarks. To disentangle the effect of the proposed objective from that of additional data, we compare against data-augmentation baselines that repurpose the same contrastive contexts as standard query–context–answer examples. These baselines provide little to no improvement, showing that the gains arise from the proposed context-selection objective rather than from the contrastive data alone. Data and code will be publicly released.

1 INTRODUCTION

Large language models (LLMs) have progressed rapidly (Team et al., 2023; Qwen Team, 2026; OpenAI, 2025; Team, 2024; DeepSeek-AI, 2026), evolving from text completion systems into general-purpose reasoning engines (Wei et al., 2023; Shao et al., 2024) capable of operating over rich, long-form contexts. This progress has enabled a wide range of intelligent systems, including agentic systems (Yao et al., 2023b; Yang et al., 2024; Anthropic, 2026; OpenClaw, 2026; Yao et al., 2023a; Shinn et al., 2023; Wang et al., 2023; Schick et al., 2023) that interleave reasoning with tool use over extended horizons, and multimodal models (Liu et al., 2023a; Alayrac et al., 2022; Li et al., 2023; Team et al., 2024; Bai et al., 2025a; OpenAI, 2025) that combine fine-grained perception with textual reasoning over high-dimensional inputs.

As these systems scale to increasingly complex tasks, their performance depends not only on reasoning ability, but also on grounding decisions in sparse yet decisive contextual evidence (Liu et al., 2023b; Mei et al., 2025): an early observation in a long agent trajectory, a single line in a tool trace, or a subtle visual detail in a dense image. When such evidence is overlooked, models may make locally plausible but context-inconsistent decisions, such as modifying a pre-defined variable in code or missing a small visual cue needed to answer a question. A growing body of work (Turpin et al., 2023; Wang et al., 2026b; Chen et al., 2026; Kamradt, 2023; Bai et al., 2024) suggests that modern models often exhibit this failure mode. We refer to it as ***context unawareness***: the relevant information is available in the context, yet the model's prediction is not grounded in it. Qualitative examples are provided in Figure 5.

To quantify this failure mode, we introduce a controlled *contrastive context probe*. We construct 200 contrastive context pairs from agentic trajectories and 200 from visual question answering (VQA) images, with examples shown in Figure 3. Each example presents the model with

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

a question, a candidate answer, and two closely matched contexts that support different answers; the model must select the context that justifies the candidate answer. As shown in Figure 1, this simple test reveals a roughly 40-point gap between proprietary models (OpenAI, 2025; Anthropic, 2026) and widely used open-source models (Qwen Team, 2026; Yang et al., 2025a). Notably, strong open-source models such as Qwen3 (VL) 8B and Qwen3.5 9B (Qwen Team, 2026; Yang et al., 2025a) perform close to random choice, despite their competitive performance on standard benchmarks. These results suggest that strong benchmark performance can obscure failures in context grounding, where models struggle to identify the evidence that supports a given answer.

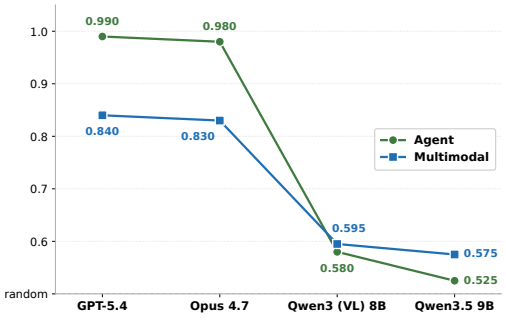

Figure 1: **Contrastive context selection given a query–answer pair.** GPT-5.4 and Claude Opus 4.7 perform reliably, while Qwen3 (VL) and Qwen3.5 remain close to random choice despite strong performance on standard benchmarks.

Motivated by this observation, we propose **Context-Aware Reinforcement Learning** (CONTEXTRL), a post-training method that augments reinforcement learning (RL) with explicit context-selection supervision. Whereas standard RL optimizes final answers through outcome rewards, CONTEXTRL adds an auxiliary objective that rewards the model for selecting the context that supports a given answer. As illustrated in Figure 2, we construct contrastive context pairs through condition search, generative editing, and similarity-based retrieval, and incorporate them into standard GRPO post-training (Shao et al., 2024; Yu et al., 2025) via a logit-level contrastive loss that favors the context aligned with the ground-truth answer.

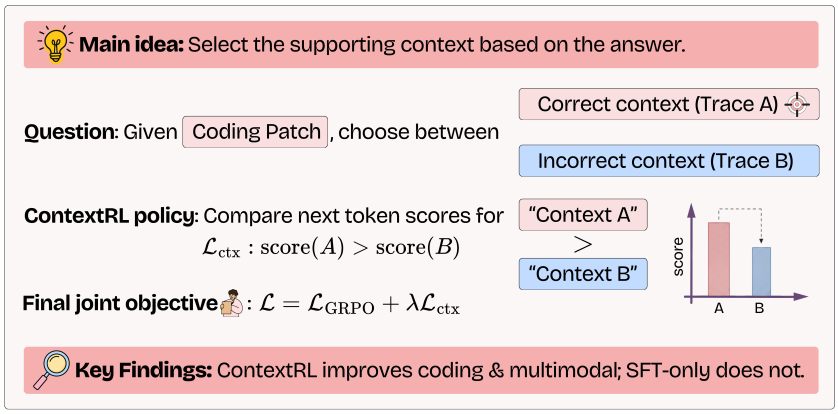

Figure 2: **Overview of CONTEXTRL.** We augment GRPO with a context-awareness loss built from contrastive context pairs. For each query-answer pair (Q, A), an anchor context $Context\ A$ supports the answer, while a confounder $Context\ B$ is superficially similar but inconsistent. Given the answer, the auxiliary objective trains the policy to select the supporting context over the confounder, improving context-aware reasoning in both coding and multimodal settings.

Despite its simplicity, this auxiliary signal yields consistent improvements across 5 long-horizon and 12 multimodal benchmarks. For long-horizon experiments, we construct 1k contrastive trajectory pairs from SWE-smith trajectories (Yang et al., 2025b) using condition filtering. CONTEXTRL improves over the GRPO baseline by +2.9% and +1.0% on average across five agentic and long-context benchmarks, using Klear-AgentForge-8B (Wang et al., 2025a) and Qwen3-8B (Yang et al., 2025a) as base models, respectively. For multimodal experiments, we construct 7k contrastive image pairs from diverse sources through generative editing and similarity-based retrieval. Across 12 benchmarks, CONTEXTRL improves over standard GRPO (Shao et al., 2024) by +2.0% on Qwen2.5-VL-7B and +1.6% on Qwen3-VL-8B, on average. To isolate the source of these gains, we compare against data-augmentation baselines that consume the same contrastive data through SFT or outcome-based RL.

Strikingly, these baselines *collapse or fail entirely*: supervised augmentation drives the long-horizon agent's resolve rate to as low as 0%, while outcome-based augmentation adds essentially nothing. Only the proposed context-selection objective converts the same data into consistent gains, indicating that the improvement stems from the training objective rather than the contrastive data alone. These results show that a single context-selection objective improves both long-horizon reasoning and multimodal understanding, without architectural changes or large-scale human annotation.

Taken together, our findings suggest that failures in long-horizon reasoning and visual understanding often reflect a common limitation: models do not reliably ground their predictions in the evidence available to them. Our diagnostic probe exposes this limitation, and CONTEXTRL shows that it can be addressed with a lightweight context-selection objective. Comparisons with data-augmentation baselines further show that the key factor is not the contrastive data alone, but the way this signal is integrated into training. By decoupling *what* answer to produce from *which* context supports it, CONTEXTRL offers a simple, lightweight, and modality-agnostic mechanism for improving context grounding in complex, information-rich settings.

2 CONTEXTRL

We introduce **Context-Aware Reinforcement Learning** (CONTEXTRL), a training framework that explicitly encourages models to ground their predictions in the provided context. Our key idea is to augment standard RL post-training with a *context-selection* signal: instead of only rewarding correct outputs, we additionally train the model to identify *which context supports a given answer*. As illustrated in Figure 2, we first construct *contrastive context pairs* (C^+, C^-) across both agentic and multimodal settings. Each pair is associated with a query Q and an anchor answer A, where C^+ is the *supporting* context for A and C^- is a minimally perturbed but *confounding* alternative that instead supports a different answer. The context takes the form of an agent trajectory τ for agentic coding, or an image I for multimodal tasks. We then jointly optimize the standard GRPO objective on the original task data and a context-awareness loss $\mathcal{L}_{\text{CA}}$ on these contrastive pairs, which conditions on (Q, A) and rewards the model for selecting C^+ over C^-.

In the following, we describe the construction of contrastive data for agentic (§2.1) and multimodal (§2.2) settings, followed by the joint training objective (§2.3).

2.1 AGENTIC CONTRASTIVE CONTEXT PAIRS CONSTRUCTION

Figure 3 (left) illustrates our pipeline for constructing contrastive trajectory pairs in the agentic setting. Each instance consists of a query Q (a GitHub issue), an answer A (the reference patch), and the trajectory τ capturing the agent's reasoning trace, tool interactions, sandbox observations, and submitted patch. Correctly producing A requires identifying which parts of the trajectory provide supporting evidence for the edit.

Mining contrastive trajectories. We construct contrastive trajectory pairs from the 66k trajectories (Wang et al., 2025a) built from SWE-smith (Yang et al., 2025b). Our pipeline applies a cascade of increasingly restrictive filters: candidate trajectories must (i) originate from the same repository and commit, (ii) modify the same file, (iii) target the same function or class, and (iv) correspond to distinct but semantically related issues. To prevent trivial leakage, patch contents inside edit commands are masked with `<PATCH_MASKED>`. These constraints yield pairs that are nearly identical at the token level: same repository, commit, file, and function, differing only in a small, decisive code region. Thus, selecting the correct trajectory requires understanding the context rather than exploiting surface statistics. We then apply automatic verification with GPT-5.4 OpenAI (2025) that explicitly screens for and rejects pairs that expose superficial shortcut cues or have ambiguous labels; cases the verifier marks `UNCERTAIN` are escalated to manual inspection. This cascade is deliberately aggressive: only **1k** high-quality contrastive trajectory pairs survive, *i.e.* 1.5% of the 66k source trajectories. Details of the step-by-step procedure are provided in Appendix B.1 and the verification prompt is provided in Appendix D.3.

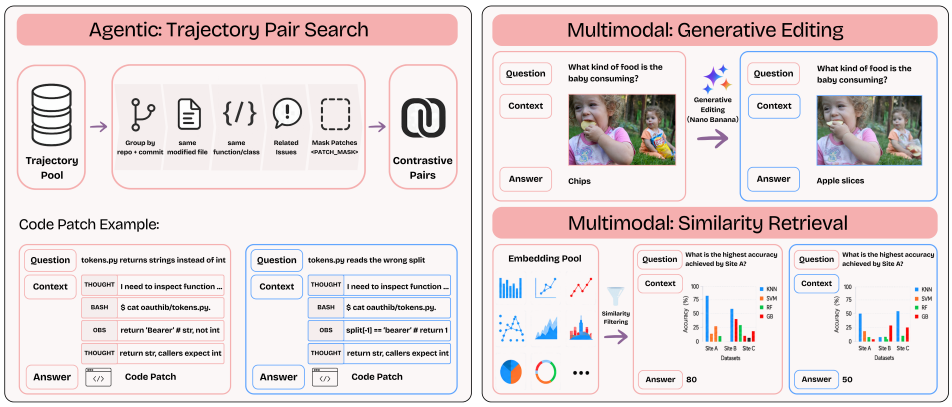

Figure 3: **Contrastive context pairs construction pipeline. Left:** Step-by-step filtering to mine contrastive trajectory pairs for the *agentic setting*. **Right:** Generative editing and similarity-based retrieval to mine contrastive image pairs for the *multimodal setting*. A concrete (C^+, C^-) example is shown alongside each method. Details can be found in Section 2.1 and 2.2.

2.2 Multimodal Contrastive Context Pairs Construction

Figure 3 (right) shows our construction of contrastive image pairs through *generative editing* and *similarity-based retrieval*. We collect image, question, and answer triplets from five domains: charts, geometry, non-geometric math, science diagrams, and natural images, covering diverse patterns of visual grounding. Dataset sources are detailed in Appendix B.2.

Generative editing for natural images. For natural images, we construct contrast pairs via controlled editing. Given (I, Q, A), we generate a modified image I' that preserves global scene structure while changing the answer to A'. This process consists of four stages: (1) *Instruction generation:* GPT-5.4 proposes an edit prompt that would change the answer; (2) *Image synthesis:* the edit is applied using Nano Banana 2 (Team et al., 2023); (3) *Verification:* GPT-5.4 explicitly rejects edited images showing visible editing artifacts (blur, warping, broken object boundaries, implausible lighting, texture mismatch, or floating objects) or global restyling that could act as a whole-image "tell", and requires the edit to stay localized to the answer-relevant region while leaving question-irrelevant content unchanged (verification prompt in Appendix D.4). (4) Cases marked as `UNCERTAIN` by the verifier are additionally reviewed by the authors. This yields approximately 700 high-quality pairs from 2k candidates, a $\sim$65% rejection rate.

Similarity-based retrieval for structured images. For structured or text-rich images, direct editing often violates underlying constraints (e.g., geometric consistency or chart semantics). Instead, we construct pairs by retrieval. We embed images using Qwen3-VL-Embedding 8B (Li et al., 2026), and for each (I, Q, A) retrieve (I', Q, A') such that $\cos(f_I(I), f_I(I')) \geq \alpha_I$ and $A \neq A'$, where we set α_I to 0.85 so that every retrieved negative is, by construction, highly visually similar to its positive yet supports a different answer. We then filter candidates using GPT-5.4 to remove semantically unrelated pairs or annotation artifacts (verification prompt in Appendix D.4). This yields 6,300 high-quality pairs from over 200k candidates.

The final multimodal training set contains **7k** contrast image pairs, combining both strategies.

2.3 Context-Aware Reinforcement Learning

We use GRPO as the base RL objective. In the agentic setting, rewards are binary based on whether the generated patch passes all test cases. In the multimodal setting, the reward is based on exact answer match. However, these rewards do not distinguish whether correct outputs are grounded in the provided context. We therefore add a *context-awareness loss*.

Context-awareness loss. Given a contrastive instance $z = (Q, A, C^+, C^-)$, the model should assign higher confidence to the positive context C^+ over the negative context C^-. We operationalize this as a two-way multiple choice: a single prompt presents Q, A, and the two contexts as labeled options ("A"/"B"), with the order of C^+ and C^- randomized per instance to remove position bias

(prompt templates in Appendix D.2, D.1). Let t^+ and t^- be the option-letter tokens assigned to C^+ and C^-, and let $\ell_\theta^+(z)$ and $\ell_\theta^-(z)$ be the model's next-token logits for t^+ and t^- at the answer position, computed by teacher forcing rather than from a sampled rollout. We define the margin $\Delta_\theta(z) = \ell_\theta^+(z) - \ell_\theta^-(z)$ and optimize:

$$\mathcal{L}_{\mathrm{CA}}(z;\theta) = -\log \sigma\left(\mathrm{clip}\left(\Delta_\theta(z), -c, c\right)\right), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function and $c > 0$ controls margin clipping. This objective encourages correct context selection while preventing large margins from dominating training. Importantly, the loss is modality-agnostic and applies uniformly to agent trajectories and images.

Joint objective. Let $\mathcal{D}_{\mathrm{RL}}$ denote the task data and $\mathcal{D}_{\mathrm{CA}}$ the contrastive dataset. We optimize:

$$\mathcal{L}(\theta) = \mathbb{E}_{x\sim\mathcal{D}_{\mathrm{RL}}}\left[\mathcal{L}_{\mathrm{GRPO}}(x;\theta)\right] + \lambda \mathbb{E}_{z\sim\mathcal{D}_{\mathrm{CA}}}\left[\mathcal{L}_{\mathrm{CA}}(z;\theta)\right], \tag{2}$$

where $\lambda > 0$ balances task performance and context awareness. The two objectives are complementary. GRPO optimizes for correct outputs, while $\mathcal{L}_{\mathrm{CA}}$ enforces alignment between outputs and their supporting context. We apply the same formulation across both agentic and multimodal settings.

3 Long Horizon Experiments

We evaluate ContextRL on five agentic and long-context benchmarks. We first describe the experimental setup (§3.1). Then, we present a comprehensive evaluation across five agentic and long-context benchmarks (§3.2). We demonstrate that our method consistently outperforms the RL baseline on **every** benchmark and exhibits strong **generalization** beyond the training distribution.

3.1 Experimental Setup

Base models. We experiment with two base models with different levels of agentic coding capability: Qwen3-8B (Yang et al., 2025a), a general-purpose model, and Klear-AgentForge-8B (Wang et al., 2025a), a model specifically fine-tuned for complex agentic coding.

Baselines. For each model, we compare Base (no RL), standard GRPO, and ContextRL. Our 8k training instances comprise 7k SWE-Gym (Pan et al., 2025)/SWE-Smith (Yang et al., 2025b) tasks for $\mathcal{L}_{\mathrm{GRPO}}$ and 1k contrastive pairs for $\mathcal{L}_{\mathrm{CA}}$; the RL baseline uses 8k source-matched tasks. Appendix E.1 provides details. We additionally report *off-the-shelf* reference models at larger scales, including Qwen3-14B, Qwen3-32B (Yang et al., 2025a), and Qwen3-Coder-30B (Yang et al., 2025a).

Evaluation benchmarks. In-distribution (ID) evaluation uses SWE-Bench Verified and Lite (Yang et al., 2024) (resolve rate). Out-of-distribution (OOD) evaluation covers LiveCodeBench v6 (Jain et al., 2024) (solve rate), LongBench v2 (Bai et al., 2024) (accuracy), and NIAH (Kamradt, 2023) (mean recall across depths and context lengths).

3.2 Main results

Table 1 reports the performance (%) of all baselines and our trained models across five benchmarks.

Consistent gains across base models. ContextRL improves over outcome-only RL on every benchmark for both base models. This highlights the robustness of Eq. 2 beyond outcome-only RL.

Competitive with much larger reference models. From Klear-AgentForge-8B, ContextRL outperforms the 4× larger Qwen3-32B and code-specialized Qwen3-Coder-30B on SWE-Bench. This shows that targeted context supervision can compensate for substantial scale differences.

OOD generalization. ContextRL beats both Base and RL baseline on **every** OOD benchmark. On NIAH, standard GRPO *regresses* while ContextRL surpasses the base. On LongBench v2, the gain is especially pronounced on the long-input subset where input length stretches context grounding the furthest. This pattern indicates that the learned signal transfers beyond the agentic training domain and captures a more general notion of context grounding.

Ablations. We investigate the effect of the context-awareness loss coefficient λ and find that performance peaks at $\lambda = 0.005$ (Appendix I.1). Additional subtask-level analyses are provided in Appendix G.

Model	In-distribution		Out-of-distribution			
	SWE-Bench Verified	SWE-Bench Lite	LiveCodeBench v6	LongBench v2		NIAH
				Overall	Long	
Off-the-shelf reference models						
Qwen3-14B	8.40	6.00	57.1	34.2	24.1	99.5
Qwen3-32B	8.40	6.00	61.1	36.8	31.5	99.3
Qwen3-Coder-30B	28.8	22.0	37.7	42.5	41.7	85.7
Trained from Qwen3-8B						
Base model	5.00	2.67	44.6	31.6	27.8	98.8
RL baseline	6.20	2.67	46.3	31.8	26.9	98.5
Ours	**7.00**	**4.00**	**47.4**	**33.2**	**29.6**	**99.0**
Δ	+0.8	+1.3	+1.1	+1.4	+2.7	+0.5
Trained from Klear-AgentForge-8B						
Base model	26.6	21.0	21.7	27.4	21.3	68.3
RL baseline	28.0	21.7	22.3	27.0	24.1	65.5
Ours	**30.2**	**24.0**	**24.0**	**29.6**	**28.7**	**71.3**
Δ	+2.2	+2.3	+1.7	+2.6	+4.6	+5.8

Table 1: **Main results on 5 long-horizon benchmarks.** CONTEXTRL consistently outperforms RL baseline across all the tasks for both base models. Notably, CONTEXTRL demonstrates robust generalization to long-context OOD tasks where standard outcome-based RL struggles or regresses.

4 MULTIMODAL EXPERIMENTS

We further evaluate CONTEXTRL on 12 multimodal benchmarks requiring coupled visual perception and downstream reasoning (setup: §4.1; results: §4.2). Consistent gains across two base models and all categories indicate stronger visual grounding that propagates to higher-level reasoning.

4.1 EXPERIMENTAL SETUP

Base models. We experiment with two vision-language models of comparable scale: Qwen2.5-VL-7B-Instruct (Bai et al., 2025b) and Qwen3-VL-8B-Instruct (Bai et al., 2025a).

Baselines. We compare Base, standard GRPO, and CONTEXTRL. The 45k-example training set contains 38k single-image tasks for $\mathcal{L}_{\text{GRPO}}$ and 7k contrastive pairs for $\mathcal{L}_{\text{CA}}$; the RL baseline uses 45k source-matched QA examples. For Qwen2.5-VL, we also report PAPO (Wang et al., 2026b), a strong but non-directly comparable perception-aware RL reference trained on its own curated dataset. Appendix E.2 gives full protocols.

Evaluation benchmarks. We evaluate on 12 diverse multimodal benchmarks spanning mathematical reasoning (MathVista (Lu et al., 2024), MathVerse (Zhang et al., 2024b), MathVision (Wang et al., 2024b)), general understanding (MMMU-Pro (Yue et al., 2024b), MMMU (Yue et al., 2024a)), perception (V* (Wu & Xie, 2024), MMStar (Chen et al., 2024), BLINK (Fu et al., 2024)), science (ScienceQA (Lu et al., 2022), PhyX (Shen et al., 2025), OlympiadBench Physics (He et al., 2024)), and real-world scenes (MME-RealWorld Lite (Zhang et al., 2025)). For consistency and reproducibility, all evaluations are conducted with two widely-adopted toolkit, LMMs-Eval (Zhang et al., 2024a) and VLMEvalKit (Duan et al., 2024).

4.2 MAIN RESULTS

Table 2 presents results across both base models and all 12 benchmarks.

Consistent gains. CONTEXTRL outperforms RL on **every** benchmark for both base models. The improvements span perception-heavy, reasoning-heavy, and mixed tasks, without gains in one category coming at the expense of another. The breadth and consistency of these gains suggest improved context grounding rather than category- or model-specific tuning.

Ablations. The λ ablation in the multimodal setting shows a similar trend, with small values (0.001 or 0.005, depending on the model) yielding the best performance (Appendix I.2).

Benchmark	Qwen2.5-VL-7B Base	RL Base	PAPO	Ours	Δ	Qwen3-VL-8B Base	RL Base	Ours	Δ
Mathematical Reasoning									
MathVista	68.2	72.5	72.7	**73.6**	+1.1	75.8	78.7	**79.8**	+1.1
MathVerse	43.9	45.3	**49.7**	49.1	+3.8	56.1	65.0	**66.4**	+1.4
MathVision	22.8	25.5	**27.3**	26.8	+1.3	46.2	49.2	**52.0**	+2.8
General Multimodal Understanding									
MMMU-Pro	36.6	41.3	42.6	**42.8**	+1.5	41.3	55.9	**57.5**	+1.6
MMMU	50.7	53.3	53.2	**54.6**	+1.3	66.4	69.1	**70.1**	+1.0
Fine-grained Visual Perception									
V*	70.1	70.7	71.7	**73.3**	+2.6	82.2	84.8	**85.9**	+1.1
MMStar	62.6	64.1	63.4	**65.1**	+1.0	70.5	73.5	**74.8**	+1.3
BLINK	55.3	56.5	58.5	**58.9**	+2.4	64.4	65.1	**66.6**	+1.5
Scientific Reasoning									
ScienceQA	88.2	91.0	92.7	**95.4**	+4.4	94.4	95.6	**96.6**	+1.0
PhyX	25.4	48.7	46.8	**50.0**	+1.3	45.5	72.1	**73.4**	+1.3
OlympiadBench Phy	1.5	3.1	2.2	**4.6**	+1.5	7.9	8.1	**9.9**	+1.8
Real-world Scene Understanding									
MME-RealWorld Lite	38.4	45.1	45.1	**46.7**	+1.6	48.7	51.9	**54.8**	+2.9
Overall Avg.	47.0	51.4	52.2	**53.4**	+2.0	58.3	64.1	**65.7**	+1.6

Table 2: **Main results on 12 diverse multimodal benchmarks.** CONTEXTRL achieves the best performance across most sub-categories, surpassing both the RL baseline and strong reference methods like PAPO. The broad improvements indicate that CONTEXTRL successfully translates enhanced visual grounding into stronger end-to-end reasoning capabilities.

Additional analyses. Appendix G provides subtask-level analyses, while Appendix H compares CONTEXTRL against methods based on different RL paradigms and at different model scales.

5 COMPARISON BETWEEN DATA AUGMENTATION AND CONTEXTRL

We investigate whether the gains from CONTEXTRL can be reproduced by standard data augmentation using the same contrastive data. This comparison isolates whether the improvement arises from the *data itself* or from the *training objective used to consume it*.

First, we introduce two natural data-augmentation baselines, DA-SFT and DA-RL, which utilize the same contrastive data (§5.1). We then compare their end-task performance in both agentic and multimodal settings, demonstrating that standard augmentation either provides no meaningful signal or leads to catastrophic policy collapse (§5.2). Finally, we provide a mechanism study to analyze the effectiveness of CONTEXTRL(§5.3).

5.1 DATA AUGMENTATION BASELINES

DA-SFT: We first perform SFT on the contrastive data using a cross-entropy loss to predict the correct context. The resulting model is then further optimized with standard GRPO on the task data. **DA-RL**: We mix contrastive data with standard task data. Each contrastive instance is treated as a binary decision problem, with reward 1 if the model selects the correct context and 0 otherwise. Under this formulation, contrastive and task examples share the same outcome-based reward signal. Both baselines use exactly the same contrastive dataset as CONTEXTRL and are trained for the same number of steps, ensuring a controlled comparison.

Configuration	Klear-AgentForge-8B Verified	Klear-AgentForge-8B Lite	Qwen3-8B Verified	Qwen3-8B Lite
RL Baseline	28.0	21.7	6.20	2.67
DA-SFT	6.40	1.33	0.00	0.00
DA-RL	27.6	21.7	5.60	3.00
Ours	**30.2**	**24.0**	**7.00**	**4.00**

Table 3: **Data augmentation results on agentic benchmarks.**

5.2 END-TASK PERFORMANCE COMPARISON

Agentic results. Table 3 shows two patterns. First, DA-SFT leads to **policy collapse**: Klear-AgentForge-8B drops from 28.0/21.7 to 6.4/1.3, and Qwen3-8B to 0.00/0.00. Although the model

Benchmark	Qwen2.5-VL-7B				Qwen3-VL-8B			
	RL Base	DA-SFT	DA-RL	**Ours**	RL Base	DA-SFT	DA-RL	**Ours**
MathVista	72.5	70.8	73.4	**73.6**	78.7	79.0	78.3	**79.8**
MathVerse	45.3	47.8	45.8	**49.1**	65.0	65.1	65.2	**66.4**
MathVision	25.5	25.1	26.5	**26.8**	49.2	47.3	49.5	**52.0**
MMMU-Pro	41.3	42.5	41.0	**42.8**	55.9	55.3	56.2	**57.5**
MMMU	53.3	53.7	54.1	**54.6**	69.1	68.2	69.4	**70.1**
V*	70.7	69.6	71.2	**73.3**	84.8	85.3	84.8	**85.9**
MMStar	64.1	64.0	64.0	**65.1**	73.5	73.1	73.6	**74.8**
BLINK	56.5	55.1	57.5	**58.9**	65.1	65.3	65.6	**66.6**
ScienceQA	91.0	92.4	92.1	**95.4**	95.6	95.4	95.8	**96.6**
PhyX	48.7	49.0	48.9	**50.0**	72.1	71.9	72.1	**73.4**
OlympiadBench Phy	3.1	3.1	3.1	**4.6**	8.1	7.9	8.6	**9.9**
MME-RealWorld Lite	45.1	45.0	44.0	**46.7**	51.9	52.8	52.8	**54.8**
Overall Avg.	51.4	51.5	51.8	**53.4**	64.1	63.9	64.3	**65.7**

Table 4: **Comparison with standard data augmentation in the multimodal setting.** DA-SFT and DA-RL yield negligible improvements over the RL baseline. In contrast, CONTEXTRL effectively extracts the learning signal to boost performance across all 12 benchmarks.

learns the context-selection task, supervised training on short responses severely disrupts the long-horizon interaction ability required for agentic coding. Second, DA-RL is **nearly indistinguishable** from the RL baseline, indicating that simply mixing contrastive examples into RL contributes little additional learning signal. In contrast, CONTEXTRL improves both models and benchmarks.

Multimodal results. Table 4 shows that DA-SFT provides negligible changes over the RL baseline, while DA-RL yields marginal gains (+0.4% / +0.2%). In contrast, CONTEXTRL achieves substantially larger gains (+2.0% / +1.6%) and improves every benchmark, suggesting that contrastive data alone is insufficient and that the key lies in how the contrastive signal is incorporated into training.

5.3 MECHANISM ANALYSIS: WHY DATA AUGMENTATION FAILS

Mechanism check: selection accuracy vs. end-task performance. To test whether each method learns context selection, we evaluate all methods on the held-out test set introduced in Fig.1 and plot selection accuracy against end-task performance in Fig.4. Three patterns emerge.

(i) Outcome-only RL fails to learn context selection. The RL baseline stays near the base-model cluster across all panels, showing that final-task rewards alone provide little signal for distinguishing context-grounded reasoning from shortcuts. DA-RL inherits this limitation, since framing context selection as binary reward prediction merely repackages the same sparse outcome signal.

(ii) DA-SFT learns context selection but harms the policy. DA-SFT achieves the highest selection accuracy in nearly every setting, confirming a strong supervision signal in the contrastive data. However, this rarely translates to downstream performance gains. In agentic setting, it nearly collapses end-task performance. We attribute this to SFT on short-answer examples shifting the model distribution away from the long-form, multi-turn patterns the agent scaffold requires. While this mismatch is mild for short multimodal responses, it is catastrophic in long-horizon settings.

(iii) Context selection alone is insufficient. Both DA-SFT and CONTEXTRL push selection accuracy to 85-92%, but only CONTEXTRL consistently improves downstream task performance. This suggests that context selection is necessary but not sufficient: the model must acquire this capability without disrupting the policy behaviors required for the original task distribution.

Why CONTEXTRL avoids both failure modes. CONTEXTRL adds context-awareness loss as a bounded auxiliary objective inside the on-policy GRPO loop. Two properties are critical.

(a) Updates remain constrained. The GRPO importance-ratio clipping term $\mathrm{clip}(\rho_t^{(i)}, 1-\epsilon, 1+\epsilon)$ and KL regularization keep the policy close to the reference model π_{ref}, while the clipped margin objective $\mathrm{clip}(\Delta_\theta(z), -c, c)$ in Eq. 1 suppresses auxiliary gradients once C^+ and C^- are well separated. Together, these mechanisms preserve the original policy distribution and prevent the catastrophic forgetting observed in DA-SFT.

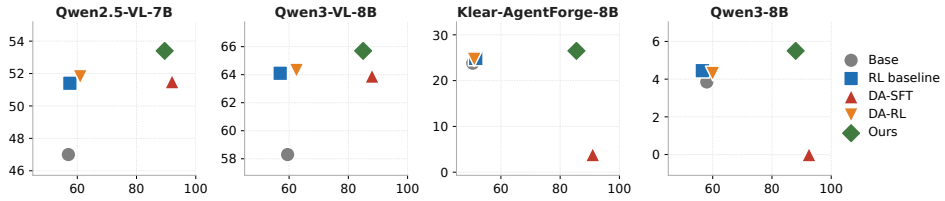

Figure 4: **Context-selection accuracy versus end-task performance.** The x-axis denotes selection accuracy. The y-axis denotes end-task performance. Top-right is optimal. DA-SFT achieves high selection accuracy but collapses on the y-axis. DA-RL fails to learn discrimination at all. CONTEXTRL is the *only* method that couples high context awareness with end-task improvements.

(b) The auxiliary signal is dense. Unlike DA-RL's sparse $0, 1$ rewards, $\mathcal{L}_{\text{CA}}$ directly supervises the relative preference between C^+ and C^- on every example. Meaningful gradients are therefore produced even when the current policy would rarely sample the correct context, precisely the regime where outcome-only RL struggles. Figure 4 reflects this clearly: CONTEXTRL is the only method that both achieves high context selection accuracy and strong end-task performance across all settings.

Overall, the contrastive data alone is insufficient. The key is the objective used to consume it.

6 RELATED WORK

RL Post-training in Agentic and Multimodal Setting. Reinforcement learning has become a standard recipe for post-training LLMs. Building on RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2024), recent verifiable-reward methods such as GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025) have driven rapid progress on mathematical and coding reasoning (team, 2025; Wei et al., 2025). The same recipe has been adapted to the agentic setting, where SWE-RL (Wei et al., 2025) and DeepSWE (Luo et al., 2025) explore RL as a new training paradigm for software-engineering agents. It also extends naturally to multimodal LLMs (Bai et al., 2025b;a), where Vision-R1 (Huang et al., 2026) and PAPO (Wang et al., 2026b) refine the RL pipeline for visual reasoning. These methods, however, optimize outcome correctness alone and provide no signal about whether the model actually grounds its answer in the supplied context. Our method CONTEXTRL targets this gap: it is a modality-agnostic training method, compatible in principle with standard policy-gradient algorithms, and supplies a process-level signal that complements outcome rewards. In this work, we instantiate it on top of GRPO.

Context Utilization and Contrastive Supervision. A growing body of work (Liu et al., 2023b; Hsieh et al., 2024; Mei et al., 2025) shows that LLMs often fail to use their supplied context faithfully. Two distinct lines of work attempt to address this. The first targets long-context settings directly: FILM (An et al., 2024) introduces information-intensive supervision for long-context retrieval, LongRLVR (Chen et al., 2026) adds context rewards in long-context RL, and MemOCR (Shi et al., 2026) preserves sparse but decisive evidence through memory- and layout-aware compression rather than contrastive selection. The second draws on contrastive supervision, widely adopted in supervised fine-tuning and DPO-style objectives: VC-STaR (Pan et al., 2026) improves VLM reasoning by contrasting visually similar VQA pairs, mDPO (Wang et al., 2024a) augments DPO (Rafailov et al., 2024) with an image-side preference term, MMEmb-R1 (Wang et al., 2026a) performs pair-aware contrastive selection for multimodal embedding, and CARE (Wang et al., 2025b) contrasts a correct rollout against hard-negative failures for verifiable reasoning. CONTEXTRL shares this contrastive spirit but operates along a different axis: rather than preferring one response over another under a fixed context, we prefer one context over another under a fixed *(Query, Answer)* pair.

7 CONCLUSION

We introduce CONTEXTRL, a context-aware RL framework that adds a lightweight context-selection objective to outcome-based post-training. Across 17 benchmarks, CONTEXTRL improves over GRPO in both long-horizon and multimodal settings. Comparisons with data-augmentation baselines show that the gains come from the training objective rather than the contrastive data alone. These results position context selection as a simple, general signal for improving context grounding.

AI USE STATEMENT

In this work, we used generative AI tools to construct and quality-control portions of the training data. GPT-5.4 was used to generate editing instructions for natural-image examples and to automatically verify candidate contrastive trajectory and image pairs. Nano Banana 2 was used to produce localized image edits for constructing synthetic contrastive examples. The authors designed the data-construction pipeline, prompts, acceptance criteria, training method, experiments, and evaluation. All ambiguous AI-generated data were manually reviewed by the authors. We also used LLMs solely to polish author-written manuscript text by correcting grammatical and spelling errors; all substantive content was written by the authors. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide the complete construction procedures for contrastive data in Appendix B, source-level dataset composition in Appendix C, all prompts in Appendix D, training protocols and hyperparameters in Appendix E, and compute requirements in Appendix F. We will release the training data and code upon publication.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. URL https://arxiv.org/abs/2204.14198.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, and Jian-Guang Lou. Make your llm fully utilize the context, 2024. URL https://arxiv.org/abs/2404.16811.

Anthropic. Claude opus 4.7, April 2026. URL https://www.anthropic.com/claude/opus.

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, 2025a. URL https://arxiv.org/abs/2511.21631.

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

Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*, 2024.

Shiyi Cao, Sumanth Hegde, Dacheng Li, Tyler Griggs, Shu Liu, Eric Tang, Jiayi Pan, Xingyao Wang, Akshay Malik, Graham Neubig, Kourosh Hakhamaneshi, Richard Liaw, Philipp Moritz, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Skyrl-v0: Train real-world long-horizon agents via reinforcement learning, 2025a.

Shiyi Cao, Dacheng Li, Fangzhou Zhao, Shuo Yuan, Sumanth R Hegde, Connor Chen, Charlie Ruan, Tyler Griggs, Shu Liu, Eric Tang, et al. Skyrl-agent: Efficient rl training for multi-turn llm agent. *arXiv preprint arXiv:2511.16108*, 2025b.

Guanzheng Chen, Michael Qizhe Shieh, and Lidong Bing. Longrlvr: Long-context reinforcement learning requires verifiable context rewards, 2026. URL https://arxiv.org/abs/2603.02146.

Lin Chen, Jinsong Li, Xiaoyi Dong, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.

Xingyu Fu, Yushi Hu, Bangzheng Li, et al. BLINK: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

Chaoqun He et al. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024. URL https://arxiv.org/abs/2404.06654.

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Xu Tang, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models, 2026. URL https://arxiv.org/abs/2503.06749.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code, 2024. URL https://arxiv.org/abs/2403.07974.

Greg Kamradt. Needle in a haystack – pressure testing LLMs. https://github.com/gkamradt/LLMTest_NeedleInAHaystack, 2023.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. URL https://arxiv.org/abs/2301.12597.

Mingxin Li, Yanzhao Zhang, Dingkun Long, Keqin Chen, Sibo Song, Shuai Bai, Zhibo Yang, Pengjun Xie, An Yang, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Qwen3-vl-embedding and qwen3-vl-reranker: A unified framework for state-of-the-art multimodal retrieval and ranking. *arXiv preprint arXiv:2601.04720*, 2026.

Zongxia Li, Wenhao Yu, Chengsong Huang, Zhenwen Liang, Rui Liu, Fuxiao Liu, Jingxi Che, Dian Yu, Jordan Boyd-Graber, Haitao Mi, et al. Self-rewarding vision-language model via reasoning decomposition. *arXiv preprint arXiv:2508.19652*, 2025.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023a. URL https://arxiv.org/abs/2304.08485.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023b. URL https://arxiv.org/abs/2307.03172.

Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Shieh. Noisyrollout: Reinforcing visual reasoning with data augmentation. *Advances in Neural Information Processing Systems*, 38:2923–2957, 2026.

Pan Lu, Swaroop Mishra, Tony Xia, et al. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35, 2022.

Pan Lu, Hritik Bansal, Tony Xia, et al. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2024.

Michael Luo, Naman Jain, Jaskirat Singh, Sijun Tan, Ameen Patel, Qingyang Wu, Alpay Ariyak, Colin Cai, Tarun Venkat, Shang Zhu, Ben Athiwaratkun, Manan Roongta, Ce Zhang, Li Erran Li, Raluca Ada Popa, Koushik Sen, and Ion Stoica. Deepswe: Training a state-of-the-art coding agent from scratch by scaling rl. https://pretty-radio-b75.notion.site/DeepSWE-Training-a-Fully-Open-sourced-State-of-the-Art-Coding-Agent-by-Scaling-RL-22281902c1468193aabbe9a8c59bbe33, 2025. Notion Blog.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. A survey of context engineering for large language models, 2025. URL https://arxiv.org/abs/2507.13334.

OpenAI. Openai gpt-5 system card, 2025. URL https://arxiv.org/abs/2601.03267.

OpenClaw. Open-source personal ai assistant, March 2026. URL https://github.com/openclaw/openclaw.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with swe-gym, 2025. URL https://arxiv.org/abs/2412.21139.

Zhiyu Pan, Yizheng Wu, Jiashen Hua, Junyi Feng, Shaotian Yan, Bing Deng, Zhiguo Cao, and Jieping Ye. Through the lens of contrast: Self-improving visual reasoning in vlms, 2026. URL https://arxiv.org/abs/2603.02556.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Hui Shen, Taiqiang Wu, Qi Han, Yunta Hsieh, Jizhou Wang, Yuyue Zhang, Yuxin Cheng, Zijian Hao, Yuansheng Ni, Xin Wang, Zhongwei Wan, Kai Zhang, Wendong Xu, Jing Xiong, Ping Luo, Wenhu Chen, Chaofan Tao, Zhuoqing Mao, and Ngai Wong. Phyx: Does your model have the "wits" for physical reasoning?, 2025. URL https://arxiv.org/abs/2505.15929.

Yaorui Shi, Shugui Liu, Yu Yang, Wenyu Mao, Yuxin Chen, Qi Gu, Hui Su, Xunliang Cai, Xiang Wang, and An Zhang. MemOCR: Layout-aware visual memory for efficient long-horizon reasoning, 2026. URL https://arxiv.org/abs/2601.21468.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023. URL https://arxiv.org/abs/2303.11366.

Deepseek team. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

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

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

LLaMa3 Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

OpenThoughts-Agent Team. OpenThoughts-Agent. https://www.open-thoughts.ai/blog/agent, December 2025.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Fei Wang, Wenxuan Zhou, James Y. Huang, Nan Xu, Sheng Zhang, Hoifung Poon, and Muhao Chen. mdpo: Conditional preference optimization for multimodal large language models, 2024a. URL https://arxiv.org/abs/2406.11839.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. URL https://arxiv.org/abs/2305.16291.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset, 2024b. URL https://arxiv.org/abs/2402.14804.

Qi Wang, Hongzhi Zhang, Jia Fu, Kai Fu, Yahui Liu, Tinghai Zhang, Chenxi Sun, Gangwei Jiang, Jingyi Tang, Xingguang Ji, Yang Yue, Jingyuan Zhang, Fuzheng Zhang, Kun Gai, and Guorui Zhou. Klear-agentforge: Forging agentic intelligence through posttraining scaling, 2025a. URL https://arxiv.org/abs/2511.05951.

Yongxin Wang, Zhicheng Yang, Meng Cao, Mingfei Han, Haokun Lin, Yingying Zhu, Xiaojun Chang, and Xiaodan Liang. CARE what fails: Contrastive anchored-REflection for verifiable multimodal reasoning, 2025b. URL https://arxiv.org/abs/2512.19554.

Yuchi Wang, Haiyang Yu, Weikang Bian, Jiefeng Long, Xiao Liang, Chao Feng, and Hongsheng Li. MMEmb-R1: Reasoning-enhanced multimodal embedding with pair-aware selection and adaptive control, 2026a. URL https://arxiv.org/abs/2604.06156.

Zhenhailong Wang, Xuehang Guo, Sofia Stoica, Haiyang Xu, Hongru Wang, Hyeonjeong Ha, Xiusi Chen, Yangyi Chen, Ming Yan, Fei Huang, and Heng Ji. Perception-aware policy optimization for multimodal reasoning, 2026b. URL https://arxiv.org/abs/2507.06448.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I. Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution, 2025. URL https://arxiv.org/abs/2502.18449.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. *arXiv preprint arXiv:2312.14135*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,

Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

John Yang, Carlos E Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik R Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.15793.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. Swe-smith: Scaling data for software engineering agents, 2025b. URL https://arxiv.org/abs/2504.21798.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models, 2023a. URL https://arxiv.org/abs/2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023b. URL https://arxiv.org/abs/2210.03629.

Shenzhi Wang Yaowei Zheng, Junting Lu et al. Easyr1: An efficient, scalable, multi-modality rl training framework. https://github.com/hiyouga/EasyR1, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Xiang Yue, Yuansheng Ni, Kai Zhang, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. *arXiv preprint arXiv:2311.16502*, 2024a.

Xiang Yue et al. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024b.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. Lmms-eval: Reality check on the evaluation of large multimodal models, 2024a. URL https://arxiv.org/abs/2407.12772.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, et al. MathVerse: Does your multi-modal LLM truly see the diagrams in visual math problems? *arXiv preprint arXiv:2403.14624*, 2024b.

Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans?, 2025. URL https://arxiv.org/abs/2408.13257.

A Qualitative Examples of Context Unawareness

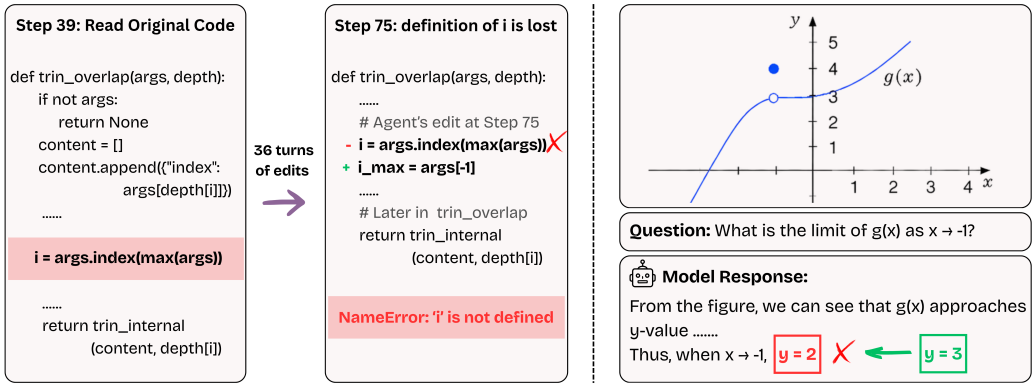

Figure 5: **Context unawareness manifests across both agentic and multimodal systems. Left:** In an agentic code-editing setting, the model has access to the relevant source file but fails to maintain consistency with the surrounding context across edits. As a result, it removes the definition of variable i that is subsequently referenced, causing a runtime error. **Right:** In a multimodal reasoning setting, the model fails to correctly ground its answer in the visual evidence. Although the relevant information is present in the figure, it misreads the y value of $g(x)$ as 2 rather than 3 as $x \to -1$, leading to an incorrect prediction.

B Additional Method Details

B.1 Detailed Procedure for Mining Agentic Contrast Trajectories

We mine contrastive trajectory pairs from the 66k SWE-smith trajectories (Wang et al., 2025a; Yang et al., 2025b) with the following step-by-step procedure.

1. *Group by repository and commit.* We group trajectories by repository and commit. This ensures that paired trajectories share the same underlying code base.
2. *Filter by modified file.* Within each group, we keep only pairs whose reference patches modify the same file. This makes the two contexts comparable at the file level.
3. *Filter by target function or class.* We parse the patch hunks and identify the function or class touched by each patch. We keep pairs that modify the same function or class. This creates pairs that differ in a small but important region of the code.
4. *Filter by issue relation.* We keep pairs whose issue descriptions are distinct but related. For example, two issues may refer to the same API, the same corner case family, or the same function behavior. This step avoids redundant pairs while preserving semantic relatedness.
5. *Mask direct patch leakage.* We replace patch content inside edit commands with `<PATCH_MASKED>`. This prevents the model from solving the context selection task by directly reading the edit command. We do not mask file views, test outputs, error messages, or reasoning traces, because these are valid parts of the context available to the agent.

Each of these filters is applied conservatively, so the number of qualifying pairs drops sharply at every stage. We further verify all resulting pairs with GPT-5.4 followed by manual inspection, removing pairs that remain ambiguous or contain residual leakage. After this pipeline, only 1k pairs out of the original 66k trajectories survive, forming our final agentic contrastive training set.

B.2 Multimodal Data Sources

We utilize source datasets spanning five visual domains to construct multimodal contrast context pairs. We use generative editing for natural images and similarity-based retrieval for structured visual inputs. The source datasets used for each domain are as follows:

- **Chart:** ChartQA, DVQA
- **Geometry:** Geo170K, Geometry3K, GeomVerse, GeoQA3, MAVIS-Geometry, R-CoT
- **Non-geometric math:** ICON-QA, MAVIS-Function, MMK12
- **Science:** M3CoT, ScienceQA
- **Natural images:** Visual CoT, ViRL39K

B.3 Dataset Difficulty and Quality Control: Screening for Artifacts and Shortcut Cues

A natural concern for any newly constructed contrastive dataset is whether the positive context can be selected by exploiting construction artifacts or superficial shortcut cues rather than by genuine context grounding. We address this along three axes: (i) the verifiers used to build the data are themselves explicit artifact/shortcut filters; (ii) the pairs are hard by construction; and (iii) the downstream evidence (OOD transfer and the data-augmentation controls) is inconsistent with artifact exploitation.

Quality control is an artifact/shortcut filter. Both verifiers are designed to *reject*, rather than tolerate, the cues a shortcut solver would exploit. For trajectories, criterion (iii) of the verifier rejects pairs with large length or formatting disparities, patch-specific tokens that leak into only one trajectory, only one trajectory inspecting the modified file or function, or masking applied inconsistently between the two trajectories. For images, criterion (i) (Appendix D.4) rejects edited images with visible editing artifacts (blur, warping, broken boundaries, implausible lighting, texture mismatch, floating objects), and criterion (iii) rejects global restyling or any change to question-irrelevant regions that could serve as a whole-image "tell". In other words, the superficial-cue analysis one might run as a post-hoc audit is built directly into our acceptance test: a pair solvable by such cues is, by design, rejected. Any case the verifier cannot confidently clear is marked `UNCERTAIN` and escalated to manual review by the authors, who apply the same criteria.

Pairs are hard by construction. The negatives are not random; the construction constrains them to lie near the similarity ceiling. *(i) Agentic.* Paired trajectories share the same repository, commit, modified file, and target function or class, and differ only in a small decisive code region, with edit commands masked by `<PATCH_MASKED>`; the two contexts are therefore nearly identical at the token level. *(ii) Multimodal, generative editing.* The negative is a localized edit of the positive that changes only the answer-relevant region while preserving the rest of the scene, so the two images are identical almost everywhere. *(iii) Multimodal, retrieval.* Negatives are retrieved under a high cosine-similarity threshold $\cos(f_I(I), f_I(I')) \geq \alpha_I$, so they are visually close to the positive by construction. Selection therefore requires resolving a small, decisive difference rather than a salient one.

Aggressive rejection. Each construction pipeline retains only a small, high-precision fraction of candidates after automatic verification and manual review, as summarized in Table 5.

Setting	Construction	Candidates	Retained
Agentic	cascade mining + verify	66,000 traj.	1,000 (1.5%)
Multimodal	generative editing	2,000	∼700 (∼35%)
Multimodal	similarity retrieval	>200,000	6,300 (∼3.1%)

Table 5: **Filtering funnel.** Each construction pipeline retains only a small, high-precision fraction of candidate pairs after automatic verification and manual review.

Why this is unlikely to be artifact exploitation. Two pieces of evidence in the main text bear on whether the learned signal is artifact detection. First, CONTEXTRL improves on benchmarks that contain *none* of our constructed inputs: the agentic OOD suite (LiveCodeBench, LongBench v2, NIAH) contains no mined pairs or masked patches, and all 12 multimodal benchmarks use natural, unedited single images in the standard (non-selection) format. An artifact-detection skill would be inapplicable on these benchmarks by construction, yet the gains transfer broadly—behavior expected of content-level grounding, not of artifact detection. Second, in the data-augmentation study (§5.3), DA-SFT attains the *highest* context-selection accuracy (85–92%) yet fails to improve, and often collapses, downstream performance; if high selection accuracy were driven by exploitable artifacts, the configuration that maximizes it should transfer best, whereas the opposite holds. Finally, in the generative-editing pipeline the editor and verifier are distinct models (Nano Banana 2 and GPT-5.4, respectively), so the verifier audits another model's output rather than its own, reducing the chance that generator-specific artifacts pass undetected. Together these indicate that what CONTEXTRL learns from the contrastive pairs is not reducible to construction-specific artifacts.

C DATASET COMPOSITION

This section gives the per-source breakdown of the multimodal training set used in §4. The complete training set contains **45,000** examples drawn from a single `jsonl` file: **38,000** standard single-image task examples used for the GRPO objective $\mathcal{L}_{\text{GRPO}}$, and **7,000** two-image contrastive instances used for the context-selection objective $\mathcal{L}_{\text{CA}}$. The 7,000 contrastive instances correspond to the **7k contrastive image pairs** reported in §2.2, and the 85%/15% task vs. contrastive split matches the proportion described in §4.

Subgroup	Source dataset	Count	% of total
Geometry	MAVIS-Geometry	18,776	41.72
	Geo170K	1,224	2.72
	R-CoT	344	0.76
	Geometry3K	140	0.31
	GeoQA3	128	0.28
	GeomVerse	117	0.26
	Subtotal	**20,729**	**46.06**
Non-geometric Math	MAVIS-Function	6,914	15.36
	ICON-QA	763	1.70
	MMK12	56	0.12
	Subtotal	**7,733**	**17.18**
Chart	DVQA	5,485	12.19
	ChartQA	40	0.09
	Subtotal	**5,525**	**12.28**
Science	M3CoT	3,449	7.66
	ScienceQA	1,634	3.63
	Subtotal	**5,083**	**11.30**
Natural Image	Visual-CoT	5,600	12.44
	ViRL39K	330	0.73
	Subtotal	**5,930**	**13.18**
Total		**45,000**	**100.00**

Table 6: Source-level composition of the 45,000-example multimodal training set.

Figure 6 visualizes the subgroup-level composition of the full 45,000-example training set across the five visual subgroups (Geometry, Non-geometric Math, Chart, Science, Natural Image). Table 6 reports the source-level breakdown within each subgroup.

Two implementation notes apply to the source-level counts. First, examples in the *Natural Image* subgroup whose answer-relevant content was modified by Nano Banana 2 are saved with new image

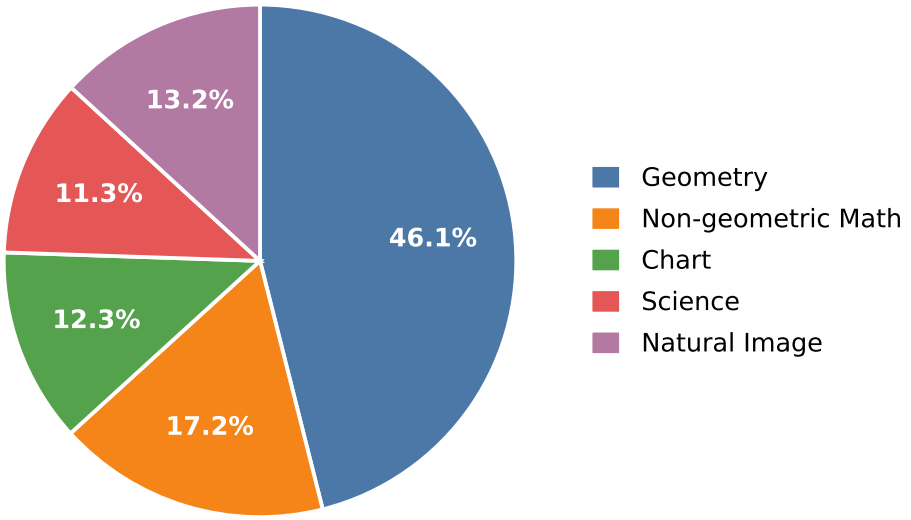

Figure 6: **Subgroup composition of the multimodal training set (45,000 examples).** Per-source counts within each subgroup are listed in Table 6.

identifiers; we count both the original and the generatively edited variant under their underlying source dataset (Visual-CoT). Second, in the contrastive subset, both images of a pair share the same source dataset by construction (similarity-based retrieval is performed within a source pool), so every pair contributes a single tally to its source.

D Prompt Templates

This section lists the prompt templates used in our experiments. Curly braces (e.g. `{question}`, `{trajectory_a}`) denote slots filled with per-instance content; all other text is fixed across instances.

D.1 Trajectory Selection Prompt

This template is used to construct the agentic context-selection instances $z = (Q, A, \tau, \tau')$. The model is shown a code patch and two candidate trajectories whose edit commands have been masked, and must identify which trajectory produced the patch.

Trajectory Selection Prompt Template

```
You are given a code patch that fixes a bug in a Python project, along
↪  with two
candidate agent trajectories labeled Option A and Option B. Each
↪  trajectory shows
an agent exploring a codebase and diagnosing a bug; the actual edit
↪  commands in
the trajectories have been replaced with the placeholder
↪  <PATCH_MASKED>.

Exactly one of the two trajectories is the one that produced the code
↪  patch
```

```
shown below. Your task is to identify which trajectory (A or B)
↪  produced the
patch.

To decide, reason about:
  - What bug each trajectory is diagnosing (from the agent's reasoning
  ↪  and the
    buggy code it inspects).
  - Whether the given code patch fixes the bug described in trajectory
  ↪  A or in
    trajectory B.

===== CODE PATCH =====
{code_patch}

===== OPTION A =====
{trajectory_a}

===== OPTION B =====
{trajectory_b}

Output only a single letter, either "A" or "B", as your final answer.
```

D.2 Image Selection Prompt

This template is used to construct the multimodal context-selection instances $z = (Q, A, I, I')$. Two images are placed before the prompt body; the model must select which image is consistent with the reference QA pair.

Image Selection Prompt Template

```
<image>
<image>
Task: You will be presented with two images, a specific question, and
↪  its
ground-truth answer. Your goal is to determine which image (if any)
↪  accurately
reflects the information provided in the QA pair.

Input Data:
The following QA pair is the reference for your verification:
```text
[REFERENCE DATA]
Question: {question}
Ground-truth Answer: {ground_truth_answer}
```

Select from the following choices.
Options:
A. The first image
B. The second image

Please select the correct answer from the options above.
```

D.3 Contrast Trajectory Pairs Verification Prompt

This is the prompt used by GPT-5.4 to verify the quality of contrast trajectory pairs. The verifier jointly assesses trajectory coherence, answer validity, and the absence of superficial shortcut cues. Examples marked `UNCERTAIN` are escalated to manual review.

Trajectory selection Verification Prompt Template

```
You are a quality verifier for an agentic trajectory dataset.
You will be shown a code patch together with two candidate agent
trajectories whose edit commands have been masked, and your job is to
decide whether this selection instance is valid for inclusion in
the dataset.

You will receive:
- Code Patch: the diff that fixes a bug in a Python project
- Trajectory A: a candidate agent trajectory with edit commands
  replaced by <PATCH_MASKED>
- Trajectory B: a second candidate trajectory with edit commands
  replaced by <PATCH_MASKED>
- Ground-Truth Label: which trajectory (A or B) actually produced
  the code patch

A valid selection instance must satisfy ALL THREE of the following
criteria:

(i) Trajectory coherence.
    Each trajectory must be internally consistent: the agent's
    reasoning, file exploration, and observed code should plausibly
    belong to a single coherent debugging rollout. There should be no
    truncation artifacts, broken tool calls, contradictory
    ↪  observations,
    or content that leaks the masked edit (e.g., explicit mention of
    the final patch text, or post-edit verification output that
    ↪  reveals
    the fix).

(ii) Validity of the ground-truth label.
     Given the code patch, exactly one of the two trajectories must be
     the unambiguous source. The bug diagnosed in the labeled
     trajectory must clearly match what the patch fixes, and the other
     trajectory must diagnose a genuinely different bug or different
     root cause that the given patch would NOT fix. If both
     trajectories' diagnoses are consistent with the patch, or if
     neither clearly is, this criterion fails.

(iii) Absence of superficial shortcut cues.
      The two trajectories should be comparable along
      content-independent dimensions, so that selection requires
      understanding the bug rather than exploiting surface statistics.
      Examples of failures: large length disparity, different
      formatting or tool-call styles, only one trajectory inspecting
      the file/function actually modified by the patch, distinctive
      tokens or identifiers from the patch appearing in only one
      trajectory, or masking applied inconsistently between the two.

Analyze the instance carefully against each criterion, then provide
your judgment in the following format:

<overall>VALID or INVALID or UNCERTAIN</overall>
<confidence>A number from 1 to 5</confidence>


Where:
- VALID: All three criteria clearly pass.
- INVALID: At least one criterion clearly fails.
- UNCERTAIN: You cannot confidently decide on at least one criterion.
- Confidence: 1 = very uncertain, 5 = very confident in your overall
↪  judgment.

Be thorough and precise in your comparison.
```

D.4 CONTRAST IMAGE PAIRS VERIFICATION PROMPT

This is the prompt used by GPT-5.4 to verify the quality of contrast image pairs. The verifier jointly assesses visual coherence, answer validity, and preservation of question-irrelevant content, and emits an overall `VALID`/`INVALID`/`UNCERTAIN` decision together with a confidence score. Examples marked `UNCERTAIN` are escalated to manual review.

Contrast Image Pair Verification Prompt Template

```
You are a quality verifier for a VQA dataset. You
will be shown a contrast pair created by editing an original image,
↪ and
your job is to decide whether the edited example is valid.

You will receive:
- Image 1 (Original Image, I): the source image before editing
- Image 2 (Edited Image, I'): the image after a localized edit
- Question (Q): a question about the image
- Original Answer (A): the correct answer to Q on the original image I
- New Answer (A'): the intended correct answer to Q on the edited
↪ image I'

A valid edited example must satisfy ALL THREE of the following
↪ criteria:

(i) Visual coherence of the edited image.
    The edited image I' must look natural and internally consistent.
    There should be no obvious editing artifacts, such as blurred or
    warped regions, broken object boundaries, implausible lighting or
    shadows, mismatched textures, floating objects, or physically or
    semantically incoherent content.

(ii) Validity of the new answer A' given I' and Q.
     When the question Q is asked about the edited image I', the
     ↪ answer
     A' must be clearly and unambiguously correct based solely on the
     visual content of I'. If A' is not supported, only partially
     supported, or if multiple answers (including the original A)
     ↪ remain
     plausible for I', this criterion fails.

(iii) Preservation of question-irrelevant content.
      All regions of the image that are not relevant to answering Q
      should remain essentially unchanged between I and I'. The edit
      should be localized to the answer-relevant region; unrelated
      objects, background, layout, style, and overall scene
      ↪ composition
      must be preserved. Global restyling, scene-wide changes, or
      unintended modifications to irrelevant regions cause this
      criterion to fail.

Analyze the pair carefully against each criterion, then provide your
judgment in the following format:

<overall>VALID or INVALID or UNCERTAIN</overall>
<confidence>A number from 1 to 5</confidence>

Where:
- VALID: All three criteria clearly pass.
- INVALID: At least one criterion clearly fails.
- UNCERTAIN: You cannot confidently decide on at least one criterion
- Confidence: 1 = very uncertain, 5 = very confident in your overall
  judgment.
```

```
Be thorough and precise in your comparison.
```

D.5 AGENTIC CODING PROMPT

This is the prompt used for the agentic coding setting. It consists of a system message that fixes the response format and a per-instance template that wraps the PR description and task instructions. `{task}` is filled with the natural-language PR description for the SWE-Bench instance.

Agentic Coding System Prompt

```
You are a helpful assistant that can interact multiple times with a
↪  computer
shell to solve programming tasks.
Your response must contain exactly ONE bash code block with ONE
↪  command (or
commands connected with && or ||).

Include a THOUGHT section before your command where you explain your
↪  reasoning
process.
Format your response as shown in <format_example>.

<format_example>
THOUGHT: Your reasoning and analysis here

```bash
your_command_here
```
</format_example>

Failure to follow these rules will cause your response to be rejected.
```

Agentic Coding Instance Prompt Template

```
<pr_description>
Consider the following PR description:
{task}
</pr_description>

<instructions>
# Task Instructions

## Overview
You're a software engineer interacting continuously with a computer by
↪  submitting
commands.
You'll be helping implement necessary changes to meet requirements in
↪  the PR
description.
Your task is specifically to make changes to non-test files in the
↪  current
directory in order to fix the issue described in the PR description in
↪  a way
that is general and consistent with the codebase.

IMPORTANT: This is an interactive process where you will think and
↪  issue ONE
command, see its result, then think and issue your next command.

For each response:
1. Include a THOUGHT section explaining your reasoning and what you're
↪  trying
```

```
   to accomplish
2. Provide exactly ONE bash command to execute

## Important Boundaries
- MODIFY: Regular source code files in /testbed (this is the working
↪  directory
  for all your subsequent commands)
- DO NOT MODIFY: Tests, configuration files (pyproject.toml,
↪  setup.cfg, etc.)

## Recommended Workflow
1. Analyze the codebase by finding and reading relevant files
2. Create a script to reproduce the issue
3. Edit the source code to resolve the issue
4. Verify your fix works by running your script again
5. Test edge cases to ensure your fix is robust

## Command Execution Rules
You are operating in an environment where
1. You write a single command
2. The system executes that command in a subshell
3. You see the result
4. You write your next command

Each response should include:
1. A **THOUGHT** section where you explain your reasoning and plan
2. A single bash code block with your command

Format your responses like this:

<format_example>
THOUGHT: Here I explain my reasoning process, analysis of the current
↪  situation,
and what I'm trying to accomplish with the command below.

```bash
your_command_here
```
</format_example>

Commands must be specified in a single bash code block:

```bash
your_command_here
```

CRITICAL REQUIREMENTS:
- Your response SHOULD include a THOUGHT section explaining your
↪ reasoning
- Your response MUST include EXACTLY ONE bash code block
- This bash block MUST contain EXACTLY ONE command (or a set of
↪ commands
  connected with && or ||)
- If you include zero or multiple bash blocks, or no command at all,
↪ YOUR
  RESPONSE WILL FAIL
- Do NOT try to run multiple independent commands in separate blocks
↪ in one
  response
- Directory or environment variable changes are not persistent. Every
↪ action is
  executed in a new subshell.
- However, you can prefix any action with
  `MY_ENV_VAR=MY_VALUE cd /path/to/working/dir && ...` or write/load
```

```
  environment variables from files

## Environment Details
- You have a full Linux shell environment
- Always use non-interactive flags (-y, -f) for commands
- Avoid interactive tools like vi, nano, or any that require user
↪  input
- If a command isn't available, you can install it

## Useful Command Examples

### Create a new file:
```bash
cat <<'EOF' > newfile.py
import numpy as np
hello = "world"
print(hello)
EOF
```

Edit files with sed:
```bash
# Replace all occurrences
sed -i 's/old_string/new_string/g' filename.py
```

View file content:
```bash
# View specific lines with numbers
nl -ba filename.py | sed -n '10,20p'
```

Submission
When you've completed your work (reading, editing, testing), and
↪ cannot make
further progress issue exactly the following command:

```bash
echo MINI_SWE_AGENT_FINAL_OUTPUT && git add -A && git diff --cached
```

This command will submit your work.
You cannot continue working (reading, editing, testing) in any way on
↪ this task
after submitting.
</instructions>
```

D.6 Multimodal Reasoning Prompt

This is the format prompt used during GRPO training and evaluation in the multimodal reasoning setting. The model is required to produce its chain of thought and final answer in dedicated tags.

Multimodal Reasoning Prompt Template

```
{Question}
{Options}
To answer this question, you need to think step by step. Provide your
↪  thinking
process in <think></think>. Provide your final answer in
↪  <answer></answer>.
```

E Training Details and Hyperparameters

E.1 Agentic Setting

We train with GRPO implemented in the **SkyRL** framework (Cao et al., 2025a), which is designed for long-horizon tool-use rollouts in sandboxed coding environments. We adopt mini-SWE-agent (Yang et al., 2024) as the agent scaffold for both training rollouts and evaluation. The training set comprises 8k instances in total: 7k standard agentic coding tasks drawn from SWE-Gym (Pan et al., 2025) and SWE-Smith (Yang et al., 2025b) for $\mathcal{L}_{\text{GRPO}}$, and 1k contrastive trajectory pairs constructed by the pipeline in §2.1 for $\mathcal{L}_{\text{CA}}$. To match the total amount of training data, the RL baseline is trained on 8k SWE-Gym/SWE-Smith tasks (the same 7k plus 1k additional samples drawn from SWE-Gym/SWE-Smith in lieu of contrastive pairs) for the same number of steps. SWE-Gym/SWE-Smith instances are disjoint from SWE-Bench Verified and SWE-Bench Lite by construction, and we manually verified that no training instance overlaps with either evaluation set. The context-selection data uses the prompt in §D.1; for Klear-AgentForge-8B, we prefill the response with `"The correct option is"` so that the next-token logit aligned to the answer letter is well-defined, while for Qwen3-8B no prefill is needed. Table 7 lists the key hyperparameters.

Hyperparameter	**Value**
KL loss / coefficient	enabled, $\beta = 1.0 \times 10^{-3}$, k_3 estimator
Optimizer / learning rate	AdamW, 2.0×10^{-6}
Train batch size	16
Policy mini-batch	8
Rollouts per prompt G	8
Max prompt length per turn	4,096
Max response length per turn	4,096
Max input length (full ctx)	28,672
Max agent turns	35
CA batch size	8
CA max prompt length	32,768
CA margin clip c	5.0
CA coefficient λ	0.005 (Klear-AgentForge-8B) / 0.001 (Qwen3-8B)
Total epochs	3

Table 7: Key training hyperparameters for the agentic coding setting.

E.2 Multimodal Setting

We train with the GRPO algorithm implemented in the Easy-R1 framework (Yaowei Zheng et al., 2025). The training set consists of 45k examples: 38k standard single-image QA examples for $\mathcal{L}_{\text{GRPO}}$ and 7k contrastive image pairs for $\mathcal{L}_{\text{CA}}$. To match the total amount of training data, the RL baseline is trained on 45k standard QA examples (the same 38k plus 7k additional samples drawn from the same source pool in lieu of contrastive pairs) for the same number of steps. For source datasets that also appear in our evaluation suite (notably ScienceQA), training draws exclusively from the official train split while evaluation uses the official test split; we manually verified that no training example overlaps with any evaluation benchmark. Both Qwen2.5-VL-7B and Qwen3-VL-8B share the same backbone configuration; the only base-model-specific difference is the $\mathcal{L}_{\text{CA}}$ coefficient λ, selected from the ablation in §I.2. For Qwen3-VL-8B we additionally override the chat template with the official thinking template so that rollouts follow the `<think>...</think>` format, and use a slightly relaxed validation sampler (temperature 0.6, top-p 0.95). The context-selection data uses the prompt in §D.2. For both Qwen2.5-VL-7B and Qwen3-VL-8B, no response prefill is needed. Table 8 lists the key hyperparameters.

F Compute Resources

Hardware. All RL training runs in this paper are conducted on a single node equipped with $4\times$ NVIDIA H200 (141GB HBM3e) GPUs, intra-node NVLink interconnect, and $\geq$ 500 GB host RAM. The same hardware configuration is used for both the multimodal and agentic settings.

Hyperparameter	Value
KL coefficient β	1.0×10^{-2}
Optimizer	AdamW, weight decay 1.0×10^{-2}
Learning rate	1.0×10^{-6}
Max gradient norm	1.0
Rollout batch size	256
Actor mini-batch size	64
Rollouts per prompt G	8
Rollout temperature	1.0
Max prompt length	16,384
Max response length	4,096
CA batch size	32
CA margin clip c	5.0
CA coefficient λ	0.005 (Qwen2.5-VL-7B) / 0.001 (Qwen3-VL-8B)
Total epochs	3

Table 8: Key training hyperparameters for the multimodal setting.

Per-experiment GPU time. Table 9 reports the approximate wall-clock and GPU-hour cost per training run on the hardware above. "Per-experiment" refers to a single end-to-end RL training run for one base model under one configuration (*e.g.*, one entry in Table 2 or Table 1); ablation studies (§I, §I.1) and the data-augmentation comparisons (§5) each consume an additional run of comparable cost.

Setting	Base model	Wall-clock (h)	GPU-hours
Multimodal	Qwen2.5-VL-7B	~ 60	~ 240
	Qwen3-VL-8B	~ 72	~ 288
Agentic	Klear-AgentForge-8B	~ 72	~ 288
	Qwen3-8B	~ 72	~ 288

Table 9: Approximate per-run compute cost on $4\times$ H200. GPU-hours are wall-clock $\times 4$.

G Subtask-Level Analysis

We conduct a subtask-level analysis on SWE-Bench Verified, LongBench v2, MathVerse, and MMStar to identify where ContextRL is most effective.

G.1 Agentic and Long-Context Settings

Model	File localization	Repair | localization	Resolve rate
Base model	72.2	36.8	26.6
RL baseline	72.6	38.6	28.0
ContextRL	**76.2**	**39.6**	**30.2**
Δ(ContextRL$-$RL)	+3.6	+1.0	+2.2

Table 10: Subtask-level results (%) on SWE-Bench Verified. "Repair | localization" is the repair rate conditioned on correct file-level localization.

In the agentic setting, ContextRL improves file-level localization accuracy on SWE-Bench Verified from 72.6% to 76.2% (+3.6), while the repair rate conditioned on correct localization increases from 38.6% to 39.6% (+1.0). This suggests that much of the gain comes from better identification of relevant code evidence. On LongBench v2, ContextRL delivers its largest gains on hard examples and long inputs, improving accuracy by $+5.5$ and $+4.6$ points, respectively.

Model	Short	Medium	Long	Easy	Hard	Overall
Base model	38.9	20.9	21.3	27.1	27.6	27.4
RL baseline	36.7	20.5	24.1	**29.2**	25.7	27.0
CONTEXTRL	**39.4**	**21.9**	**28.7**	27.1	**31.2**	**29.6**
Δ(CONTEXTRL−RL)	+2.7	+1.4	+4.6	−2.1	+5.5	+2.6

Table 11: Subtask-level accuracy (%) on LongBench v2, grouped by context length and difficulty.

G.2 MULTIMODAL SETTINGS

Model	Text Lite	Text Dominant	Vision Intensive	Vision Dominant	Vision Only	Overall
Qwen2.5 Instruct	46.7	52.8	41.9	40.5	37.8	43.9
Qwen2.5 RL baseline	48.0	52.2	43.4	41.8	41.1	45.3
PAPO	52.3	57.9	47.5	45.7	45.1	49.7
Qwen2.5 CONTEXTRL	50.0	55.3	**48.6**	**45.8**	**45.9**	49.1
Δ(CONTEXTRL−RL)	+2.0	+3.1	+5.2	+4.0	+4.8	+3.8
Qwen3 Instruct	62.4	68.4	55.7	57.6	36.5	56.1
Qwen3 RL baseline	67.1	**75.3**	60.9	60.4	61.2	65.0
Qwen3 CONTEXTRL	**67.4**	74.5	**62.4**	**63.2**	**64.6**	**66.4**
Δ(CONTEXTRL−RL)	+0.3	−0.8	+1.5	+2.8	+3.4	+1.4

Table 12: Subtask-level accuracy (%) on MathVerse.

Model	Coarse Perc.	Fine-grained Perc.	Instance Reas.	Logical Reas.	Sci. & Tech.	Math	Overall
Qwen2.5 Instruct	70.0	52.4	68.4	65.2	50.4	69.2	62.6
Qwen2.5 RL baseline	71.6	54.4	70.8	67.2	50.0	**70.8**	64.1
PAPO	72.4	54.8	70.4	64.4	48.4	70.0	63.4
Qwen2.5 CONTEXTRL	**74.0**	**57.2**	**72.0**	67.6	50.0	69.6	**65.1**
Δ(CONTEXTRL−RL)	+2.4	+2.8	+1.2	+0.4	0.0	−1.2	+1.0
Qwen3 Instruct	72.8	58.8	77.2	75.2	85.2	54.0	70.5
Qwen3 RL baseline	72.0	59.6	**79.2**	81.2	87.6	61.2	73.5
Qwen3 CONTEXTRL	**76.0**	**62.4**	78.0	**82.4**	**88.0**	**62.0**	**74.8**
Δ(CONTEXTRL−RL)	+4.0	+2.8	−1.2	+1.2	+0.4	+0.8	+1.3

Table 13: Subtask-level accuracy (%) on MMStar.

In the multimodal setting, the gains are consistently larger on vision-dependent subsets. On MathVerse, Qwen2.5-VL improves by $+5.2$, $+4.0$, and $+4.8$ points on Vision Intensive, Vision Dominant, and Vision Only subsets, compared with $+2.0$ and $+3.1$ on the more text-heavy subsets. Qwen3-VL shows the same trend. On MMStar, the largest gains also appear in coarse- and fine-grained perception, reaching up to $+4.0$ and $+2.8$ points.

Overall, CONTEXTRL is most beneficial when success depends on locating relevant code evidence, integrating difficult or long-context information, or extracting fine-grained visual details. This pattern is consistent with the intended context-grounding mechanism.

H EXPANDED EVALUATION

We expand our evaluation along two axes: (i) different training paradigms and RL algorithms, including SkyRL-Agent (Cao et al., 2025b) and OpenThinker-Agent (Team, 2025) in the agentic setting, and NoisyRollout (Liu et al., 2026) and SelfRewarded-R1 (Li et al., 2025) in the multimodal setting; and (ii) comparisons spanning 7B, 8B, and 32B parameter scales.

H.1 AGENTIC RESULTS

CONTEXTRL substantially outperforms the SkyRL-Agent and OpenThinker-Agent on both SWE-Bench Verified and SWE-Bench Lite.

Model	SWE-Bench Verified	SWE-Bench Lite
SkyRL-Agent-8B-v0	5.00	3.33
OpenThinker-Agent-8B-RL	13.4	10.3
CONTEXTRL	**30.2**	**24.0**

Table 14: Comparison with publicly available 8B agentic RL models. Results are resolve rates (%).

Model	MathVista	MathVerse	MathVision	MMMU-Pro	MMMU	V*
NoisyRollout 7B	67.7	48.2	**27.6**	40.8	53.2	71.2
SelfRewarded-R1-7B	66.8	47.2	25.9	42.7	54.2	**73.3**
Qwen2.5 CONTEXTRL	**73.6**	**49.1**	26.8	**42.8**	**54.6**	**73.3**
NoisyRollout 32B	71.6	54.4	40.6	50.8	60.4	67.0
Qwen3 CONTEXTRL	**79.8**	**66.4**	**52.0**	**57.5**	**70.1**	**85.9**

Table 15: Expanded multimodal evaluation (part 1). All values are accuracy (%).

H.2 MULTIMODAL RESULTS

The multimodal results show that CONTEXTRL outperforms other RL paradigms at comparable model scales. Moreover, Qwen3-VL-8B with CONTEXTRL surpasses NoisyRollout-32B on all 12 benchmarks despite using a substantially smaller backbone.

I ABLATIONS

In this section we provide detailed ablation studies for both agentic and multimodal settings.

I.1 AGENTIC SETTING

Effect of context-awareness loss coefficient λ. We study the sensitivity of the agentic setting to the weight λ of the context-awareness loss in Eq. 2, sweeping $\lambda \in \{0.001, 0.005, 0.01\}$ on Klear-AgentForge-8B. Table 17 reveals a clear trend: performance peaks at $\lambda = 0.005$. At $\lambda = 0.001$, the resolve rate is indistinguishable from the RL baseline reported in Table 1, indicating that the context-awareness signal is too weak to meaningfully influence training. At $\lambda = 0.01$, the score drops below even the RL baseline, suggesting that an overly strong CA term competes with the GRPO objective and degrades final task performance. We adopt $\lambda = 0.005$ as the default for agentic training.

λ	Verified	Lite
0.001	28.2	21.3
0.005	**30.2**	**24.0**
0.01	28.2	20.0

Table 17: Effect of λ on agentic coding. Resolve rate (%) on SWE-Bench.

I.2 MULTIMODAL SETTING

We conduct ablation studies to understand the sensitivity of our method to three key hyperparameters: the proportion of contrastive data, the maximum response length during RL rollouts, and the contrastive loss coefficient λ.

Effect of contrastive data proportion. We vary the proportion of contrastive image pairs in the training set across five levels: 5%, 10%, 15%, 20%, and 50%. Using Qwen2.5-VL-7B as the base model, we maintain the same total training steps across all five settings. Figure 7 visualizes performance across all 12 benchmarks.

The results reveal a clear inverted-U pattern. Increasing the contrastive proportion from 5% to 15% generally improves performance on most benchmarks. However, further increasing to 20% and especially 50% degrades performance broadly: at 50%, nearly all benchmarks are at or below the 5% level. This trend indicates that excessive contrastive data displaces the standard reasoning supervision. We select 15% as our default setting, as it strikes the best balance between context sensitivity and reasoning capability.

Model	MMStar	BLINK	ScienceQA	PhyX	Olympiad Phy	MME-RW Lite	Avg.
NoisyRollout 7B	65.0	55.3	90.7	**50.5**	4.3	46.6	51.7
SelfRewarded-R1-7B	64.8	55.8	90.9	49.3	3.5	**47.3**	51.8
Qwen2.5 ContextRL	**65.1**	**58.9**	**95.4**	50.0	**4.6**	46.7	**53.4**
NoisyRollout 32B	69.1	62.0	91.9	68.5	7.2	45.5	57.4
Qwen3 ContextRL	**74.8**	**66.6**	**96.6**	**73.4**	**9.9**	**54.8**	**65.7**

Table 16: Expanded multimodal evaluation (part 2). MME-RW denotes MME-RealWorld.

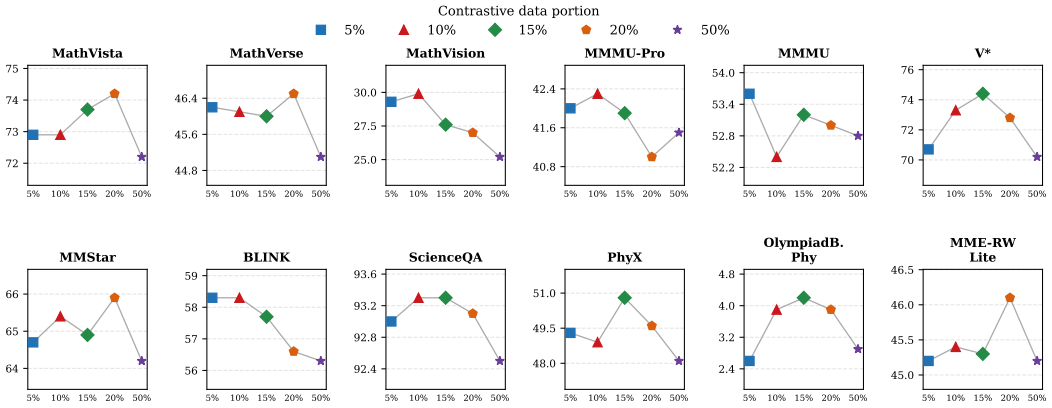

Figure 7: Effect of contrastive data proportion on Qwen2.5-VL-7B performance. The x-axis is data proportion and the y-axis is accuracy (%). The 15% setting achieves the best overall balance. Runs in this ablation use a separate, shorter training schedule.

Effect of maximum response length. During RL rollouts, the maximum response length controls the reasoning budget available to the model. We compare three settings, 2048, 4096, and 8192 tokens, on Qwen3-VL-8B. Figure 8 shows the results across all 12 benchmarks.

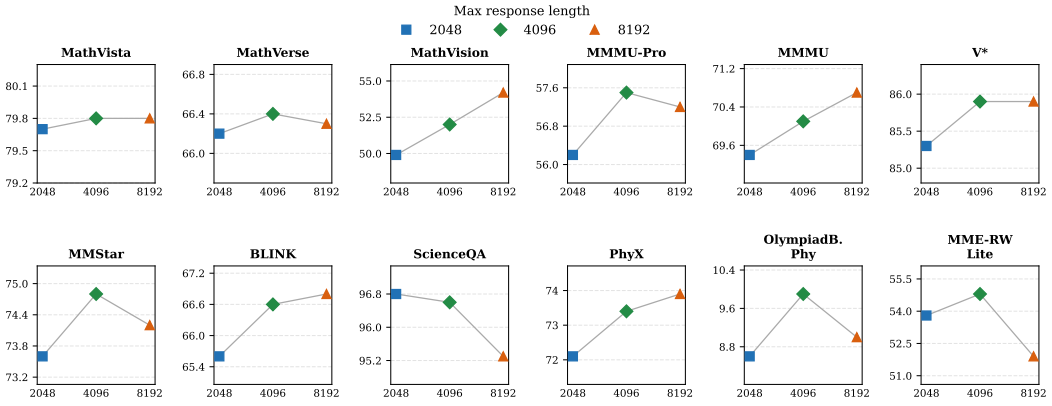

Figure 8: Effect of maximum response length on Qwen3-VL-8B performance. Max length 4096 offers the best average performance across all benchmarks.

From figure 8 we can find that increasing the response length from 2048 to 4096 yields consistent improvements across most benchmarks, with notable gains on tasks requiring extended reasoning chains like MathVision and MMMU-Pro. Further increasing to 8192 provides additional gains on MathVision and MMMU, but slightly hurts performance on ScienceQA, OlympiadBench Physics, and MME-RealWorld, likely because longer generation windows introduce noise on tasks that do not benefit from extended reasoning. We select 4096 as our default, as it offers the best average performance across all benchmarks.

Effect of contrastive loss coefficient λ. The coefficient λ in Eq. 2 controls the relative weight of the context-selection loss. We compare $\lambda \in \{0.001, 0.005, 0.01\}$ on both Qwen2.5-VL-7B and Qwen3-VL-8B. Figure 9 shows per-benchmark results.

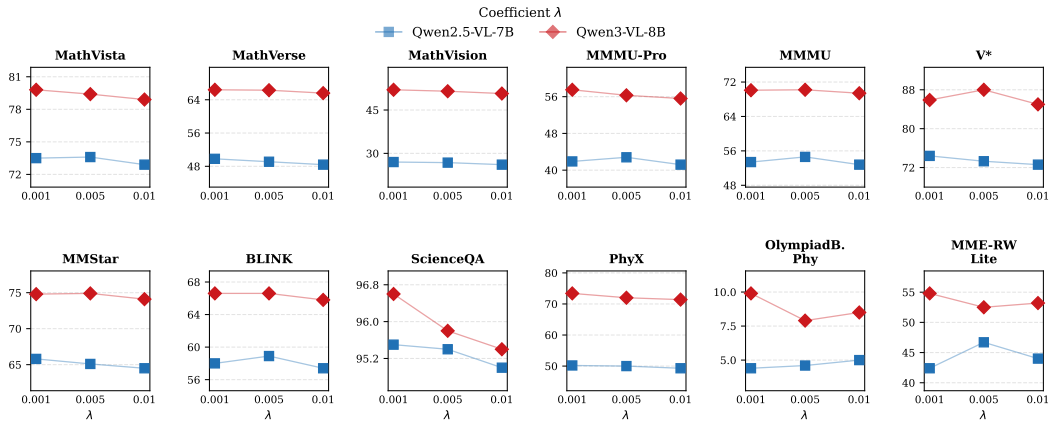

Figure 9: Effect of the contrastive loss coefficient λ on benchmark performance. Both models are stable across small λ values but degrade on the majority of benchmarks once λ is increased to 0.01.

Within the small-λ regime, both models are relatively robust to the choice between 0.001 and 0.005. On Qwen2.5-VL-7B, $\lambda = 0.005$ achieves slightly better overall performance, while on Qwen3-VL-8B, $\lambda = 0.001$ performs slightly better on average. Further increasing λ to 0.01, however, degrades performance on the majority of benchmarks for both models. This matches our expectation: the context-selection loss is intended as an auxiliary objective that augments rather than competes with the RL objective. When λ becomes too large, the CA gradient overwhelms the GRPO update signal, and the auxiliary signal ends up doing more harm than good. We therefore select $\lambda = 0.005$ for Qwen2.5-VL-7B and $\lambda = 0.001$ for Qwen3-VL-8B.

J Limitations

Due to computational constraints, all of our experiments are conducted on base models with fewer than 10B parameters; we have not validated ContextRL at substantially larger scales (e.g., 30B+ or 70B+). Moreover, the majority of base models we evaluate are drawn from the Qwen family. Validating ContextRL across a broader set of model families would strengthen the generality of our claims.

K Broader Impacts

We propose a new RL post-training paradigm that explicitly rewards context grounding rather than only the final answer, with the aim of making both multimodal and agentic LLMs more faithful to the evidence in their inputs. We expect the positive impact to come from improved reliability in tasks where shortcut answers are dangerous: software-engineering agents that must read code carefully before editing it, and visual reasoning systems that must read fine-grained perceptual evidence before answering. Like any technique that strengthens general LLM capabilities, ContextRL could in principle be used to amplify harmful applications, but it does not require any additional sensitive data. We do not anticipate negative societal impacts.