AdaC-GRPO: Towards Mastering Long-Horizon Visual Sudoku Puzzles via Adaptive Curriculum Learning

Sida Li[1] **Yuanxin Liu**[1] **Linli Yao**[1] **Shicheng Li**[1] **Xu Sun**[1]
Haoli Bai[2] **Xiao-Hui Li**[2]

[1]National Key Laboratory for Multimedia Information Processing, Peking University
[2]Huawei Technologies Co., Ltd.

{lisida,liuyuanxin,linliyao}@stu.pku.edu.cn {lisc99,xusun}@pku.edu.cn
{baihaoli,lixiaohui33}@huawei.com

Abstract

While reinforcement learning with verifiable reward (RLVR) enhances reasoning in large language models, its efficacy in long-horizon multimodal tasks remains underexplored. Visual Sudoku puzzles pose a representative challenge, requiring multi-step deductive reasoning over both local constraints and global dependency chains. To this end, we introduce FG-Sudoku, a fine-grained benchmark categorized by grid size and required deductive steps. Evaluations reveal that frontier MLLMs fail on complex long-range dependencies (e.g., 9×9 puzzles). During post-training, this near-zero success rate causes severe reward sparsity, halting standard RLVR. We propose Adaptive Curriculum Group Relative Policy Optimization (AdaC-GRPO) to overcome this. Unlike rigid or prior-based schedules, AdaC-GRPO dynamically constructs training batches by tracking the model's "Zone of Proximal Development" via online rollout accuracy. Experiments demonstrate that AdaC-GRPO enables Qwen2.5-VL-7B to approach the performance of Gemini-2.5-Flash while bridging 62.2% of the overall accuracy gap to Gemini-2.5-Pro, with its generalizability further validated across InternVL-3.5 and Qwen3-VL. Project page: https://github.com/DuHeng0128/AdaC-GRPO.

1 Introduction

Reinforcement learning with verifiable reward (RLVR) has achieved remarkable success in enabling large language models (LLMs) to tackle a wide range of reasoning tasks, including mathematical problem solving (Cobbe et al., 2021; Liu et al., 2024b; Hendrycks et al., 2021; Lu et al., 2024), general visual puzzles (Chia et al., 2024; Zhang et al., 2025), and video reasoning (Hu et al., 2025; Cheng et al., 2025; Liu et al., 2025; Qi et al., 2025). However, while current paradigms excel at tasks requiring localized reasoning or pattern recognition, it remains unclear whether they can effectively handle **long-horizon multimodal reasoning** tasks, such as solving *Visual Sudoku Puzzles*.

Sudoku is a logic-intensive puzzle that requires filling a partially completed $N \times N$ grid with numbers under strict constraints, involving multi-step deductive reasoning over both short-range local constraints and long-range global dependencies. The generalized Sudoku decision problem is NP-complete (Yato and Seta, 2003), posing a formidable challenge for computational reasoning and serving as a clean abstraction of real-world constraint satisfaction problems.

In this work, we investigate the capability of multimodal large language models (MLLMs) to solve visual Sudoku puzzles (where the Sudoku grid is presented visually). Specifically, we seek to answer two key research questions: (1) To what extent can current MLLMs autonomously solve visual Sudoku puzzles without external symbolic solvers? (2) How can we improve their performance on such long-horizon reasoning tasks?

To address the first question, we introduce **FG-Sudoku**, a fine-grained diagnostic benchmark spanning multiple difficulty levels, chosen because Sudoku's difficulty is fully controllable by solver-derived deductive cost and its correctness is fully verifiable without any LLM-as-judge. The difficulty of each instance is quantified by the grid size and the number of deductive steps required by standard solving algorithms. Using this benchmark, we evaluate several frontier MLLMs and observe a clear limitation: while they perform competently on short-range constraints, their accuracy drops to near-zero on challenging 9×9 instances due to failure in capturing long-range dependency chains. In RLVR post-training, this directly induces a **reward-variance collapse**: with a binary outcome reward and a near-zero pass rate, every sample in a GRPO group receives the same reward, the group-relative advantage degenerates to zero, and policy gradients vanish. This is exactly the regime where curricu-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 48255–48274
October 24-29, 2026 ©2026 Association for Computational Linguistics

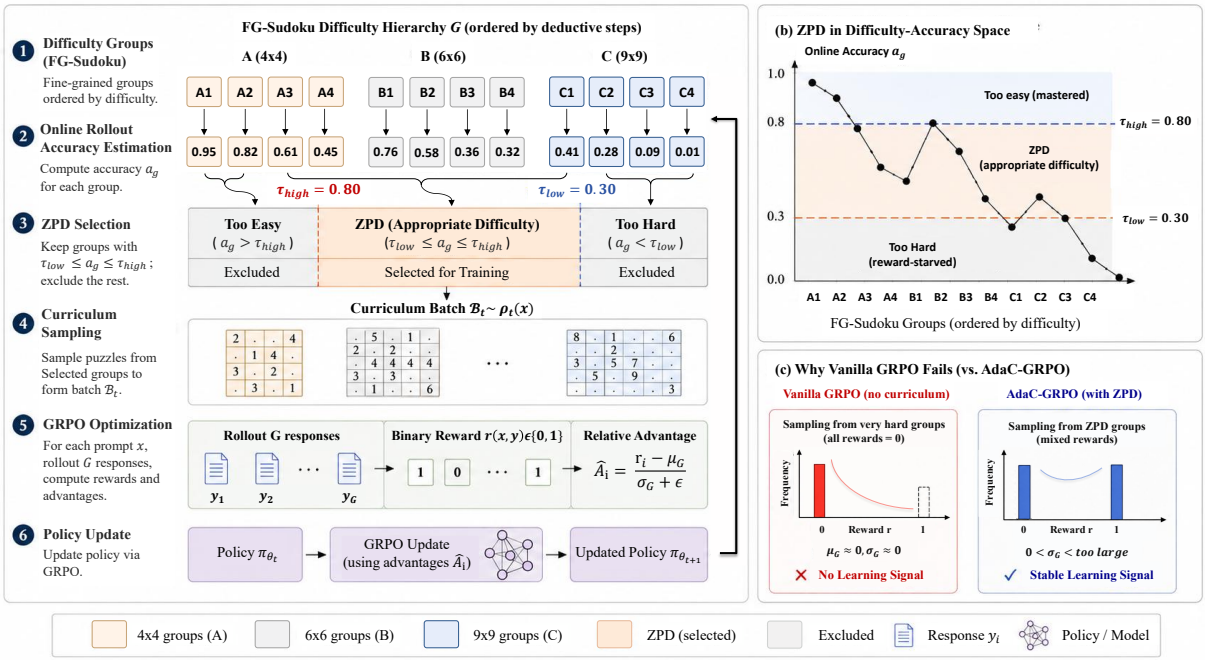

Figure 1: Overview of AdaC-GRPO. The policy dynamically selects FG-Sudoku groups within the Zone of Proximal Development (ZPD) based on online rollout accuracy, excluding trivially solved and reward-starved tasks.

lum learning should help.

To answer the second question, we propose **Adaptive Curriculum Group Relative Policy Optimization (AdaC-GRPO)**. Existing curricula struggle with strict-constraint RL: static schedules (e.g., CurrReFT (Deng et al., 2025)) advance regardless of the model's competence, while adaptive samplers (e.g., AdaRFT (Shi et al., 2025)) prioritize the highest-error instances. In long-horizon reasoning, however, the hardest instances yield a zero pass rate, so error-driven mining exacerbates reward starvation and halts optimization. AdaC-GRPO instead shifts from error-driven mining to *competence-driven progression*: it uses online rollout accuracy not to find the hardest tasks but to pinpoint the model's "Zone of Proximal Development" (ZPD), bounding the active sampling distribution from both below and above, ensuring that each batch lies in the sweet spot where the model is sufficiently challenged yet still receives dense positive reinforcement.

Experimental results show that while standard supervised fine-tuning (SFT) followed by GRPO improves Qwen2.5-VL-7B on easy and medium Sudoku cases, the model continues to struggle on hard puzzles. In contrast, AdaC-GRPO lifts Qwen2.5-VL-7B's C-group accuracy from a near-zero baseline to 0.219, approaching Gemini-2.5-Flash on the hardest 9×9 tier and closing 62.2% of the overall gap to Gemini-2.5-Pro, while outperforming curriculum baselines such as AdaRFT and CurrReFT by over 5× on the hardest tier. The framework also generalizes to InternVL-3.5 and Qwen3-VL.

Our main contributions are as follows:

- We introduce **FG-Sudoku**, a diagnostic benchmark for evaluating MLLMs on visual Sudoku reasoning.
- We propose **AdaC-GRPO**, which addresses reward-variance collapse in binary-reward RLVR by constraining sampling to a Zone of Proximal Development.
- We empirically show that AdaC-GRPO substantially improves Qwen2.5-VL-7B on visual Sudoku, consistently outperforming SFT, GRPO, CurrReFT, and AdaRFT on the same backbone. Our model approaches Gemini-2.5-Flash on the hardest 9×9 tier, and closes 62.2% of the overall gap to Gemini-2.5-Pro.

2 Related Work

2.1 Symbolic and constraint reasoning

Classical constraint satisfaction and symbolic reasoning frameworks form the foundation for combinatorial puzzles such as Sudoku (Yato and Seta, 2003). Neuro-symbolic methods attempt

to integrate differentiable learning with logic constraints (Bhuyan et al., 2024), while recent hybrid systems enable models to propagate symbolic consistency through latent representations. However, these systems typically rely on explicit rule supervision or modular solver, limiting scalability and flexibility. Our work takes a complementary approach: rather than embedding symbolic solvers into networks, we treat Sudoku as a self-contained, end-to-end environment where the model must infer constraint structure implicitly through reward feedback.

2.2 Multimodal reasoning and benchmarks

Recent multimodal foundation models—including Gemini 2.5 (Gemini Team, Google, 2025), Qwen2.5-VL (Bai et al., 2025b), GPT-4o (OpenAI, 2024), DeepSeek-V3 (DeepSeek-AI, 2024), and InternVL-3 (Zhu et al., 2025)—exhibit strong perception-language alignment yet continue to struggle with multi-step relational reasoning and constraint satisfaction. A growing body of benchmarks has sought to expose these weaknesses (Luo et al., 2025). ReasonBench (Potamitis et al., 2025) and MATH-Bench (Liu et al., 2024b) emphasize symbolic and numerical reasoning; CLEVR (Johnson et al., 2017), and GQA (Hudson and Manning, 2019) target visual compositionality and logical consistency; while ScienceQA (Lu et al., 2022) and VCR (Zellers et al., 2019) explore multimodal question answering requiring factual grounding and inference. More recently, datasets such as ReasonVQA (Tran et al., 2025) extend this paradigm to visual abstraction and analogical reasoning, highlighting persistent failures of current models to maintain symbolic consistency across modalities. Meanwhile, reasoning-intensive puzzle benchmarks—such as SudokuBench (Seely et al., 2025), Zebra-CoT (Li et al., 2025), and PuzzleBench (Zhang et al., 2025) further highlight the brittleness of current models under explicit symbolic constraints. Yet these datasets focus primarily on final-answer accuracy, leaving open the challenge of long-range constraint propagation.

Within this context, FG-Sudoku introduces a complementary perspective: a formally defined constraint-satisfaction reasoning task that preserves combinatorial complexity. It offers a continuous difficulty spectrum derived from solver statistics and enables quantitative analysis of end-to-end reasoning improvement. By focusing on a minimal but expressive domain, FG-Sudoku serves as a diagnostic counterpart to large-scale open-ended reasoning suites, providing a controlled environment for probing how multimodal models propagate local constraints into globally consistent solutions.

2.3 Reinforcement fine-tuning and curriculum learning

Reinforcement fine-tuning (RFT) has emerged as a central tool for improving reasoning behavior beyond supervised imitation (Ouyang et al., 2022; Rafailov et al., 2023; Christiano et al., 2017). This paradigm has been prominently validated by recent frontier models achieving state-of-the-art reasoning capabilities, such as OpenAI o3 and the DeepSeek series (e.g., DeepSeek-V3 and R1) (DeepSeek-AI, 2024, 2025). Recent extensions, including GRPO (Shao et al., 2024), RRHF (Yuan et al., 2023), and DPO-based methods (Rafailov et al., 2023), reformulate the reward-weighted objective to improve stability. Multimodal variants such as RL-VLM-F (Wang et al., 2024) and Video-R1 (Feng et al., 2025) demonstrate that policy-gradient training can enhance factuality and consistency, but they remain highly vulnerable to sparse or discontinuous reward signals.

Classical work (Bengio et al., 2009) and recent adaptive curricula (Matiisen et al., 2020; Graves et al., 2017) dynamically adjust task difficulty to match a model's evolving competence. In the reasoning domain, approaches such as AdaRFT (Shi et al., 2025), CurrReFT (Deng et al., 2025), and GRPO (Shao et al., 2024) have improved optimization efficiency on sparse-reward tasks. Unlike static schedules or prior-based scoring that collapse on hard constraints, our proposed **AdaC-GRPO** dynamically tracks the model's *Zone of Proximal Development (ZPD)* via online rollout accuracy. This structure-aware window ensures the dense positive feedback necessary for end-to-end optimization.

DAPO (Yu et al., 2025) also addresses zero-variance rollout groups, but it does so after sampling: prompts whose rewards are all zero or all one are discarded and replaced. AdaC-GRPO makes its routing decision before the training rollouts. It estimates competence at the difficulty-group level and assigns the next batch to a group inside the ZPD. The distinction matters in hard regions, where post-hoc filtering can spend most of the rollout budget before finding that a prompt provides no reward contrast for estimating group-relative advantages.

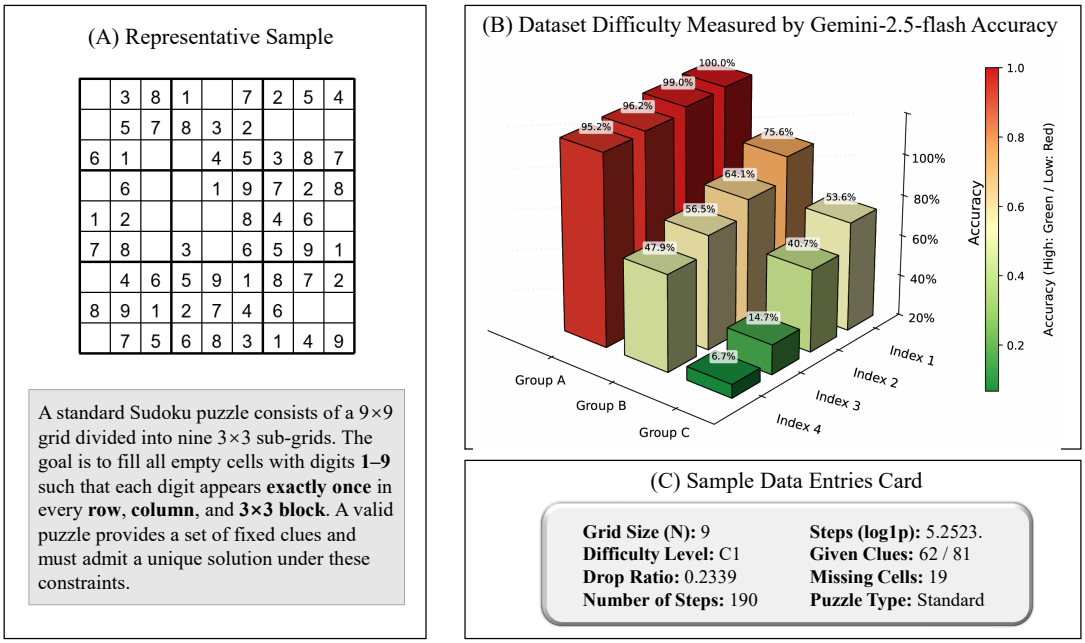

Figure 2: Overview of the dataset, showing a representative Sudoku sample, model-measured difficulty steps across subsets, and the metadata structure of individual entries. Panel (B) reports Gemini-2.5-Flash's pass@1 accuracy on each subgroup as a post-hoc empirical validation of our solver-derived difficulty hierarchy; correctness is determined by exact-grid matching. Higher bars generally correspond to easier subgroups.

3 FG-Sudoku: Dataset Construction

To systematically study long-horizon reasoning, we introduce FG-Sudoku, which contains puzzles at three canonical grid sizes: 4×4, 6×6, and 9×9. The complete dataset contains 3,072 training instances and 768 held-out test instances, uniformly distributed across grid sizes and carefully stratified into four difficulty tiers. We further use Gemini 2.5 Pro (Gemini Team, Google, 2025) to synthesize high-quality **chain-of-thought (CoT) traces**. These traces can be used to bootstrap the model during the initial supervised fine-tuning (SFT) phase.

3.1 Generation and Validation Pipeline.

For each size, we generate valid Sudoku boards and dynamically sample a drop ratio (fraction of given cells removed) to ensure layout diversity. Furthermore, we apply a deterministic constraint solver alongside a backtracking search (with an MRV heuristic) to eliminate instances with multiple solutions, guaranteeing strict unique solvability. The solver also records a structural trace of operations. By weighting and aggregating these operations, we derive a solver-based estimate of the deductive cost, S, for each instance. This highly optimized pipeline synthesizes 3K valid instances in roughly 60 seconds on a standard CPU, enabling flexible on-the-fly curriculum generation. See Appendix A for detailed drop-ratio distributions and solver heuristics.

3.2 Difficulty Stratification.

Driven by the $O(n^{n^2})$ search space, S follows a heavily right-tailed distribution. Naive equal-width binning triggers *curriculum starvation*: most instances concentrate in the easy bins, leaving the hard tail under-sampled and unable to produce a reliable training signal. We instead use **quantile-based partitioning** (K=4, balanced by count), which guarantees uniform sample mass across difficulty groups regardless of the distribution's skew. Formally, the k-th tier for grid size n is defined by the empirical quantiles $q_k^{(n)}$ of S:

$$\mathcal{G}_{n,k} = \left\{ i \mid q_{k-1}^{(n)} \leq s_i < q_k^{(n)} \right\}$$

This provides balanced data buffers across the continuous spectrum of deductive complexity, neutralizing the underlying heavy-tailed distribution.

3.3 Advantages and Solver Traces.

Existing benchmarks such as SudokuBench and Zebra-CoT mainly provide curated puzzles for static evaluation, sometimes accompanied by natural-language reasoning. In FG-Sudoku, the recorded traces allow us to select instances by

reasoning-path properties and compare model behavior under controlled settings. These properties include the structure of dependency chains and the number of steps over which global exclusivity constraints propagate. Figure 3 shows a solution trace visualization of a 4×4 Sudoku puzzle, demonstrating that even seemingly simple instances require long, tightly coupled sequences of reasoning.

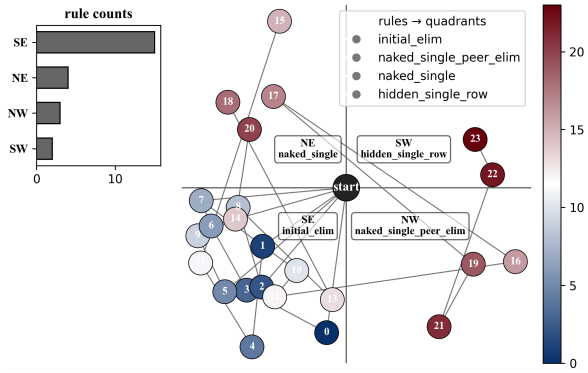

Figure 3: Visualization of the solution trace for a 4×4 Sudoku puzzle. Each quadrant represents a distinct class of operations. Operations applied to the same cell are connected by straight-line links, and the colors change continuously from blue to red to indicate the temporal sequence of the reasoning steps.

4 AdaC-GRPO

4.1 Competence-Driven Curriculum via ZPD Scheduling

Let $x \in \mathcal{X}$ be a Sudoku instance drawn from a time-varying task distribution $\rho_t(x)$. Due to the combinatorial state space $\mathcal{O}(n^{n^2})$, uniform sampling across the full difficulty hierarchy $\mathcal{G}$ produces highly sparse reward signals and uninformative policy gradients on complex instances.

AdaC-GRPO reframes the curriculum as a dynamic distribution-matching problem bounded by the model's *Zone of Proximal Development* (ZPD). At curriculum epoch t with current policy π_{θ_t}, we define the empirical rollout accuracy for a difficulty group $g \in \mathcal{G}$ as

$$a_g^{(t)} = \hat{\mathbb{E}}_{\substack{x \sim \mathcal{D}_g \\ \mathbf{y} \sim \pi_{\theta_t}(\cdot|x)}} \Big[\mathbf{1}\big(\text{Grid}(\mathbf{y}) = \mathbf{y}^\star_{\text{grid}}(x)\big) \Big], \tag{1}$$

where $\mathcal{D}_g$ is the uniform distribution over instances in group g, $\mathbf{y}^\star_{\text{grid}}(x)$ is the ground-truth solution grid, and $\text{Grid}(\mathbf{y})$ extracts the answer matrix from the model output. The indicator checks **exact cell-by-cell match**. The ZPD is the feasibility region $[\tau_{\text{low}}, \tau_{\text{high}}]$ that filters out both trivially mastered and overwhelmingly complex tasks. We schedule the training target g_t^* by

$$g_t^* = \arg\max_{g \in \mathcal{G}} \Big(a_g^{(t)} \cdot \mathbb{I}_{\text{ZPD}}(g, t) \Big), \tag{2}$$

where $\mathbb{I}_{\text{ZPD}}(g,t) = \mathbf{1}(\tau_{\text{low}} \le a_g^{(t)} \le \tau_{\text{high}})$. The active sampling distribution $\rho_t(x)$ is then deterministically routed to $\mathcal{D}_{g_t^*}$, keeping training within a dense reward landscape. The selection rule additionally handles cases where no candidate group lies within the ZPD. If all candidate groups satisfy $a_g^{(t)} > \tau_{\text{high}}$, training terminates since the curriculum has been fully mastered. Otherwise, the policy rolls back to the previously trained group g_{t-1}^* for additional consolidation, allowing the model to re-enter the ZPD before further progression.

The Sudoku solver is used only to construct the static groups $\mathcal{D}_g$ and their reference solutions. AdaC-GRPO does not query the solver during training. The scheduler uses the policy's online rollout accuracy $a_g^{(t)}$, so other tasks can define groups from structural metadata, human labels, or empirical pass rates.

4.2 Group Relative Policy Optimization for Strict Constraints

Constraint satisfaction is brittle: a single invalid assignment corrupts the global solution. We therefore use an outcome-based binary reward complemented by an auxiliary format term:

$$r(x, \mathbf{y}) = \underbrace{\mathbf{1}\big[\text{Grid}(\mathbf{y}) = \mathbf{y}^\star_{\text{grid}}(x)\big]}_{r_{\text{acc}}(x,\mathbf{y})} + \lambda \cdot r_{\text{format}}(\mathbf{y}), \tag{3}$$

where $r_{\text{format}}(\mathbf{y}) = |\{c \in \mathbf{y} : c \in \mathcal{P}\}|/|\mathbf{y}|$. $\mathcal{P}$ denotes the set of tokens conforming to the predefined parsable output format. The combination $r(x, \mathbf{y})$ drives the GRPO advantage in Eq. (4). However, the format term r_{format} is not used for ZPD selection, as we are more concerned with changes in problem-solving ability. We also explored lenient, dense intermediate rewards (e.g., partial assignment accuracy), but found they emit false-positive advantage signals that reinforce trajectories satisfying local but violating global constraints.

We optimize the policy with the standard GRPO framework (Shao et al., 2024). For each prompt $x \sim \rho_t(x)$, we sample a group of G outputs $\{\mathbf{y}_1, \ldots, \mathbf{y}_G\} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and compute the group-relative advantage

$$\hat{A}_i = \frac{r(x, \mathbf{y}_i) - \mu_G}{\sigma_G + \epsilon}, \tag{4}$$

where μ_G and σ_G are the mean and standard deviation of the group rewards.

Why impose a lower bound τ_{low}? Error-driven curricula such as AdaRFT (Shi et al., 2025) treat higher error as greater learning potential and route training toward the hardest instances. In binary-reward RLVR, sufficiently hard prompts often yield $r_{\text{acc}} = 0$ for all G rollouts, giving $\mu_G = \sigma_G = 0$ and $\hat{A}_i \equiv 0$. Such groups provide no task-success learning signal. DAPO (Yu et al., 2025) handles this by oversampling and retaining only groups containing both successful and failed rollouts. Its post-hoc filter ensures nonzero variance for retained groups, although discarded groups still consume rollouts. For a prompt with independent rollout success probability p, the probability of obtaining an informative group is

$$q(p) = 1 - (1-p)^G - p^G. \tag{5}$$

The expected number of raw rollouts required to obtain one retained group is therefore

$$\mathbb{E}[N_{\text{roll}}] = \frac{G}{q(p)} \approx \frac{1}{p}, \qquad Gp \ll 1. \tag{6}$$

This cost grows rapidly as p approaches zero. AdaC-GRPO uses online competence estimates to restrict $\rho_t(x)$ before generation, postponing groups whose estimated rollout accuracy falls below τ_{low} until their accuracy rises into the ZPD as the policy improves, thereby keeping the success probability of sampled tasks away from zero. If exactly one of the G rollouts succeeds, that trajectory is the sole source of positive credit, and its normalized advantage scales as $\mathcal{O}(\sqrt{G})$ ($\sqrt{G-1}$ under the population-standard-deviation convention). Since final-answer rewards do not assess intermediate reasoning, the update may reinforce a success that the model cannot reliably reproduce. Restricting training to the ZPD reduces dependence on such rare events.

5 Experiments

We structure our evaluation around four core research questions: (**RQ1**) Can FG-Sudoku accurately distinguish the reasoning capabilities of multimodal models across varying grid sizes and structural difficulties? (**RQ2**) Does AdaC-GRPO improve sample efficiency and final performance over vanilla GRPO? (**RQ3**) Is fine-grained difficulty stratification strictly necessary? (**RQ4**) Can trace alignment provide actionable diagnostics?

5.1 Experimental Setup

Baselines and Metrics. We evaluate a diverse set of frontier multimodal large language models (MLLMs), including proprietary systems (e.g., Gemini 2.5 Pro, GPT-4o) and open-weight alternatives (e.g., Qwen2.5-VL, InternVL3.5). The prompts used in the evaluation are listed in Appendix D. We report *Pass@1* as the primary metric.

AdaC-GRPO Implementation Details. We implement AdaC-GRPO on top of the open-source training codebase released by Video-R1 (Feng et al., 2025), which provides multimodal GRPO infrastructure (the Video-R1 fine-tuned *model checkpoint* is reported separately as a baseline in Table 1a). Prior to RL optimization, we cold-start the model's step-by-step reasoning format and rule adherence by supervised fine-tuning (SFT) on 300 Gemini-2.5-Pro-annotated traces. During AdaC-GRPO training, we sample $G = 8$ candidate completions per prompt, utilizing the combined binary accuracy and formatting reward defined in Section 4. Our curriculum schedule dynamically routes the active sampling distribution strictly to difficulty groups whose validation accuracy falls within the empirical ZPD bounds $[\tau_{\text{low}}, \tau_{\text{high}}] = [0.3, 0.8]$. We additionally provide threshold sensitivity and selection-rule ablations in Appendix B, to qualitatively determine the upper and lower bounds and the argmax selection mechanism; full hyperparameter configurations are also detailed there.

5.2 Main Results (RQ1 & RQ2)

The benchmark results (Table 1a) reveal a clear long-horizon reasoning bottleneck shared across the frontier. Addressing **RQ1**, while most frontier models perform competently on local 4×4 grids (Group A), even strongly capable models collapse to near-zero on the combinatorial 9×9 Group C. This sharp degradation as deductive cost grows isolates multi-step constraint propagation, rather than perception, as the limiting factor.

Addressing **RQ2**, AdaC-GRPO raises overall Pass@1 from 0.427 with GRPO to 0.586. DAPO provides a stronger baseline than vanilla GRPO: its dynamic sampling reaches 0.540 overall, but only 0.086 on Group C, compared with 0.219 for AdaC-GRPO. A rollout-efficiency analysis further shows that, upon entering Group C, 99.7% of DAPO rollout groups have zero accuracy-reward variance, compared with 54.7% for AdaC-GRPO. This shows that curriculum progression yields more

Model	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4
Block A: Proprietary Models												
GPT-4o (OpenAI, 2024)	0.729	0.598	0.377	0.571	0.171	0.094	0.048	0.042	0.000	0.000	0.000	0.000
o3 (OpenAI, 2025)	1.000	0.979	0.962	1.000	0.939	0.891	0.855	0.938	0.203	0.136	0.059	0.000
Claude-Sonnet-4 (Anthropic, 2025)	0.965	0.907	0.830	0.619	0.829	0.781	0.565	0.708	0.754	0.407	0.206	0.017
DeepSeek-V3 (DeepSeek-AI, 2024)	0.565	0.464	0.547	0.429	0.268	0.281	0.177	0.188	0.072	0.017	0.000	0.000
Gemini-2.5-Flash (Gemini Team, Google, 2025)	1.000	0.990	0.962	0.952	0.756	0.641	0.565	0.479	0.536	0.407	0.147	0.067
Gemini-2.5-Pro (Gemini Team, Google, 2025)	0.988	1.000	0.981	1.000	0.939	0.859	0.935	0.917	0.681	0.814	0.735	0.383
Block B: Open-weight Models												
LLaVA-NeXT-34B (Liu et al., 2024a)	0.024	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Video-R1 (Feng et al., 2025)	0.282	0.093	0.113	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
InternVL3-8B (Zhu et al., 2025)	0.008	0.006	0.002	0.005	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
InternVL3.5-8B (Wang et al., 2025)	0.765	0.588	0.453	0.333	0.220	0.062	0.048	0.021	0.029	0.000	0.000	0.000
Qwen2.5-VL-7B (Bai et al., 2025b)	0.624	0.361	0.264	0.143	0.061	0.016	0.000	0.000	0.000	0.000	0.000	0.000
Qwen3-VL-8B (Bai et al., 2025a)	0.906	0.784	0.642	0.476	0.634	0.406	0.177	0.104	0.058	0.051	0.000	0.000
Block C: Training Paradigms												
Qwen2.5-VL-7B + SFT	0.624	0.495	0.340	0.333	0.159	0.031	0.032	0.021	0.058	0.034	0.000	0.000
Qwen2.5-VL-7B + GRPO (Shao et al., 2024)	0.906	0.835	0.509	0.476	0.732	0.422	0.355	0.292	0.087	0.068	0.000	0.000
Qwen2.5-VL-7B + CurrReFT (Deng et al., 2025)	0.588	0.722	0.623	0.619	0.793	0.516	0.468	0.312	0.043	0.000	0.000	0.000
Qwen2.5-VL-7B + AdaRFT (Shi et al., 2025)	0.788	0.784	0.698	0.571	0.646	0.531	0.500	0.333	0.116	0.051	0.000	0.000
Qwen2.5-VL-7B + DAPO (Yu et al., 2025)	**0.976**	**0.948**	**0.792**	0.619	0.866	0.609	0.516	**0.438**	0.232	0.102	0.000	0.000
Qwen2.5-VL-7B + AdaC-GRPO **(Ours)**	**0.976**	0.928	0.774	**0.667**	**0.878**	**0.641**	**0.548**	0.396	**0.478**	**0.288**	**0.074**	**0.017**
Block D: Architecture Generalization												
InternVL3.5-8B + AdaC-GRPO **(Ours)**	0.976	0.897	0.792	0.714	0.854	0.609	0.581	0.417	0.420	0.322	0.059	0.033
Qwen3-VL-8B + AdaC-GRPO **(Ours)**	0.965	0.938	0.830	0.667	0.902	0.656	0.548	0.354	0.507	0.339	0.103	0.067

(a) Detailed per-category results.

Model	A	B	C	Total
Block A: Proprietary Models				
GPT-4o	0.5935	0.0978	0.0000	0.2304
o3	0.9842	0.9065	0.1017	0.6641
Claude-Sonnet-4	0.8867	0.7304	0.3557	0.6576
DeepSeek-V3	0.5118	0.2342	0.0233	0.2565
Gemini-2.5-Flash	0.9844	0.6291	0.2930	0.6355
Gemini-2.5-Pro	0.9921	0.9139	0.6562	0.8540
Block C: Training Paradigms				
Qwen2.5-VL-7B + SFT	0.4925	0.0704	0.0235	0.1954
Qwen2.5-VL-7B + GRPO	0.7616	0.4807	0.0391	0.4271
Qwen2.5-VL-7B + CurrReFT	0.6486	0.5549	0.0116	0.4050
Qwen2.5-VL-7B + AdaRFT	0.7501	0.5232	0.0430	0.4388
Qwen2.5-VL-7B + DAPO	**0.8984**	0.6367	0.0859	0.5404
Qwen2.5-VL-7B + AdaC-GRPO	0.8906	**0.6485**	**0.2189**	**0.5860**

(b) Aggregated group-level results.

Model	A	B	C	Total
Block B: Open-weight Models				
LLaVA-NeXT-34B	0.0080	0.0000	0.0000	0.0027
Video-R1	0.1523	0.0000	0.0000	0.0508
InternVL3-8B	0.0117	0.0000	0.0000	0.0039
InternVL3.5-8B	0.5979	0.1015	0.0078	0.2357
Qwen2.5-VL-7B	0.4104	0.0235	0.0000	0.1446
Qwen3-VL-8B	0.7698	0.3669	0.0274	0.3881
Block D: Architecture Generalization				
InternVL3.5-8B + AdaC-GRPO	0.8865	0.6447	0.2108	0.5807
Qwen3-VL-8B + AdaC-GRPO	**0.9024**	**0.6520**	**0.2578**	**0.6041**

(c) Aggregated group-level results.

Table 1: Benchmarking leading proprietary and open-weight multimodal models (Pass@1). Panel (a) reports per-tier results, while panels (b) and (c) report group-level averages. A, B, and C denote the 4×4, 6×6, and 9×9 subsets, respectively; 1–4 denote difficulty tiers. Block C compares training methods under the same Qwen2.5-VL-7B backbone, SFT warm start, reward, and rollout budget. **Bold** denotes the best result within Block C, while underlined values indicate the best result across all models.

informative rollouts when direct sampling rarely succeeds (Appendix E.3). AdaC-GRPO also outperforms curriculum-based methods such as CurrReFT and AdaRFT at every difficulty level, with the gap widening substantially on harder instances. Appendix B.5 reports variance across runs.

Disentangling the SFT warm-start. To isolate the contribution of the 300 Gemini-generated SFT traces from the RL gain, the `+SFT` row in Table 1a and 1b serve as the warm-start-only baseline. SFT alone achieves 0.023 on C-group — essentially identical to the untuned Qwen2.5-VL-7B baseline (0.000) and far below AdaC-GRPO (0.219). The hard-instance gain therefore cannot be attributed to teacher-generated CoT data; it stems from the curriculum-bounded RL phase.

Closing the gap to proprietary models. On Block C of Table 1b, AdaC-GRPO closes 62.2% of the overall accuracy gap between the untuned Qwen2.5-VL-7B baseline and Gemini-2.5-Pro (and 33.4% on C-group). On the hardest 9×9 tier specifically, AdaC-GRPO (0.219) approaches Gemini-2.5-Flash (0.293). This indicates that competence-driven ZPD scheduling can lift a 7B open-weight model toward mid-tier proprietary performance on long-horizon constraint reasoning.

5.3 Curriculum Trajectory

To understand how AdaC-GRPO scales the difficulty hierarchy, we monitor the online rollout accuracy on held-out validation subsets across training epochs (Figure 4). The trajectory shows a clear sequential dependency between subsets. By structurally grouping instances and dynamically activating subsets based on the $[\tau_{\text{low}}, \tau_{\text{high}}]$ threshold, AdaC-GRPO keeps the active policy within its ZPD (orange shaded region), supplying the dense gradient signals required to master increasingly complex global dependencies.

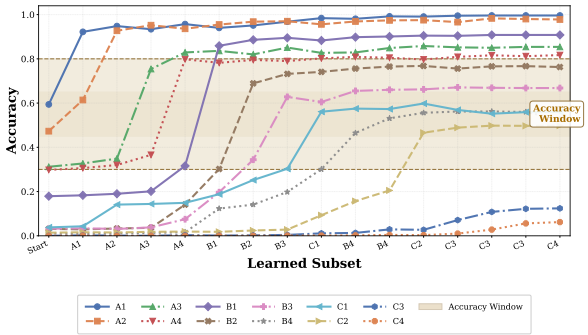

Figure 4: Trajectory of policy competence during curriculum-guided unit testing. The horizontal axis represents sequentially activated training subsets, while the vertical axis shows the rollout accuracy ($f = 1/4$). The orange shaded area denotes the globally defined Zone of Proximal Development (ZPD) bounds $[\tau_{\text{low}}, \tau_{\text{high}}]$.

5.4 Ablation and Analysis

Necessity of Fine-Grained Stratification (RQ3). To address **RQ3**, we ablate the fine-grained quantile stratification by partitioning instances solely by grid size (4×4, 6×6, 9×9). As shown in Table 2(a), this coarse-grained curriculum degrades performance, particularly on the C-group (0.219 → 0.012). Without fine-grained stepping stones, the structural difficulty gap between 6×6 and 9×9 grids is too large to bridge in one step; the model suffers reward starvation, failing to maintain a viable ZPD and halting optimization prematurely.

Binary vs. lenient rewards. In Section 4, we argued that lenient intermediate rewards, such as partial-grid accuracy, can assign positive learning signals to trajectories that are locally plausible but globally invalid. Table 2(b) supports this hypothesis: an AdaC-GRPO variant trained with partial rewards performs well on the simple A-group instances but drops to 0.000 on the C-group. In globally constrained tasks such as Sudoku, local correctness does not necessarily imply progress toward a globally feasible solution. Binary outcome rewards therefore better align the optimization signal with exact task success and avoid reinforcing partially correct but ultimately invalid trajectories, particularly when successful rollouts are rare on difficult instances.

(a) Impact of Curriculum Granularity

Curriculum Granularity	A	B	C	Total
Coarse-grained (size only)	0.6484	0.5547	0.0117	0.4049
Fine-grained (ours)	**0.8906**	**0.6485**	**0.2189**	**0.5860**

(b) Impact of Reward Formulation

Reward Formulation	A	B	C	Total
Lenient (partial) Reward	0.5860	0.1250	0.0000	0.2370
Binary Reward (ours)	**0.8906**	**0.6485**	**0.2189**	**0.5860**

(c) Train/Evaluation Modality

Train → Eval	A	B	C	Total
Untrained → Text	0.535	0.047	0.004	0.195
Untrained → Visual	0.410	0.024	0.000	0.145
Text → Text	**0.906**	0.582	0.125	0.538
Visual → Visual (ours)	0.891	**0.649**	**0.219**	**0.586**
Text → Visual	0.707	0.496	0.082	0.428
Visual → Text	0.859	0.547	0.109	0.505

Table 2: Ablation results. **(a)** Fine-grained tiers bridge the gap between grid sizes. **(b)** Partial reward fails on Group C. **(c)** Each trained condition uses one input form throughout training; cross-modal rows change only the evaluation input.

Visual vs. textual input modality. Table 2(c) disentangles training and evaluation modality. Without training, text input is consistently easier than the rendered board. Under matched train–evaluation modalities, visual training performs better overall and on Groups B and C, while text remains stronger on Group A. Cross-modal evaluation also shows stronger transfer from visual to text, suggesting that the two-dimensional layout provides additional benefit as constraint reasoning becomes more difficult.

Trace-Based Diagnostics (RQ4). Trace alignment shows that the baselines often lose track of the puzzle state and introduce duplicate digits within a row, column, or box. Relative to the SFT baseline on the same evaluated traces, AdaC-GRPO reduces the count of these solver-detected constraint conflicts by 67.4% (Each duplicate pair counts as one conflict). Its remaining errors are more often associated with higher-order constraint patterns such as *Hidden Pairs*. Appendix G gives the violation taxonomy and examples.

Method	Easy	Medium	Hard	Overall
ZebraLogicBench-ID (Lin et al., 2025)				
Base	0.453	0.047	0.014	0.165
GRPO	**0.656**	0.063	0.014	0.235
AdaC-GRPO	0.625	**0.109**	**0.042**	**0.250**
CMM-Math (Liu et al., 2024c)				
Base	0.526	0.480	0.410	0.469
GRPO	0.537	0.493	0.416	0.477
AdaC-GRPO	**0.540**	**0.508**	**0.427**	**0.487**

Table 3: Pass@1 after training GRPO or AdaC-GRPO from the same task-specific initialization. ZebraLogicBench uses structural-complexity tiers; CMM-Math uses its provided level labels.

Generalization across tasks. We train AdaC-GRPO and GRPO separately on each benchmark from the same task-specific initialization. AdaC-GRPO improves performance on the medium and hard tiers of both tasks. ZebraLogicBench contains only 64–72 examples in each of these tiers, so the absolute differences should be interpreted cautiously. CMM-Math contains roughly 1,800–2,200 examples per tier and exhibits the same trend with substantially larger evaluation sets. These results evaluate the task-level generalization of the scheduling strategy beyond Sudoku. Additionally, Appendix C evaluates zero-shot checkpoint transfer from the Sudoku-trained model to out-of-domain tasks and shows that AdaC-GRPO reduces the average reasoning-token usage by roughly 50%.

6 Conclusion

This work addresses the reward starvation challenge in applying RL to long-horizon visual reasoning. We introduce FG-Sudoku, a diagnostic benchmark isolating multi-step constraint bottlenecks. To overcome optimization collapse on complex tasks, we propose AdaC-GRPO, a competence-driven curriculum that confines active sampling to the model's Zone of Proximal Development (ZPD). Experiments show AdaC-GRPO stabilizes gradient signals and lifts Qwen2.5-VL-7B's C-group accuracy from a near-zero baseline to 0.219, substantially outperforming standard GRPO and existing curriculum-learning methods. It approaches Gemini-2.5-Flash on the hardest tier and closes 62.2% of the overall gap to Gemini-2.5-Pro. Future work will explore broader task transfer and principled multi-task curricula for real-world long-horizon reasoning problems.

Limitations

Compute and statistical coverage. We repeat GRPO and AdaC-GRPO across three training seeds, while the remaining training baselines are reported from single runs due to the high computational cost of multimodal RL. The hard tiers of ZebraLogicBench also contain relatively few test examples, leading to higher sampling uncertainty in the reported differences.

Difficulty partitions. AdaC-GRPO assumes access to a verifiable reward and a meaningful partition of the task space. Such partitions may be difficult to construct in open-ended domains where reliable metadata or stable empirical difficulty tiers are unavailable.

Scope of hard-case and cross-task evaluation. Performance remains low on the hardest C3 and C4 tiers. FG-Sudoku currently covers standard 4×4, 6×6, and 9×9 puzzles, without evaluating irregular variants or larger grids. Beyond Sudoku, our experiments include one text-based CSP benchmark and one multimodal mathematics benchmark; broader validation on open-ended visual reasoning tasks remains for future work.

Acknowledgments

We thank the anonymous reviewers for their constructive feedback. Xu Sun is the corresponding author of this paper.

References

Anthropic. 2025. Introducing claude 4.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-VL technical report. *arXiv preprint arXiv: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, and 8 others. 2025b. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *International Conference on Machine Learning*.

Bikram Pratim Bhuyan, Amar Ramdane-Cherif, Ravi Tomar, and Thipendra P. Singh. 2024. Neuro-

symbolic artificial intelligence: a survey. *Neural Computing and Applications*, 36:12809 – 12844.

Junhao Cheng, Yuying Ge, Teng Wang, Yixiao Ge, Jing Liao, and Ying Shan. 2025. Video-holmes: Can mllm think like holmes for complex video reasoning? *ArXiv*, abs/2505.21374.

Yew Ken Chia, Vernon Toh, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. 2024. Puzzlevqa: Diagnosing multimodal reasoning challenges of language models with abstract visual patterns. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16259–16273.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Karl Cobbe, Vineet Kosaraju, Mo Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations*.

DeepSeek-AI. 2024. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Huilin Deng, Ding Zou, Xinghao Zhao, Rui Ma, Yanming Guo, Yang Cao, and Yu Kang. 2025. Curreft: Overcoming training bottlenecks in small-scale vision-language models via curriculum reinforcement finetuning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12021–12032. Association for Computational Linguistics.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. 2025. Video-r1: Reinforcing video reasoning in mllms. *ArXiv*, abs/2503.21776.

Gemini Team, Google. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. Pmlr.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Xiaodong Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *ArXiv*, abs/2103.03874.

Kairui Hu, Penghao Wu, Fanyi Pu, Wangding Xiao, Yuanhan Zhang, Xiang Yue, Bo Li, and Ziwei Liu. 2025. Video-mmmu: Evaluating knowledge acquisition from multi-discipline professional videos. *ArXiv*, abs/2501.13826.

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

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

Jixuan Leng, Chengsong Huang, Langlin Huang, Bill Yuchen Lin, William W. Cohen, Haohan Wang, and Jiaxin Huang. 2025. Crosswordbench: Evaluating the reasoning capabilities of llms and lvlms with controllable puzzle generation. *Preprint*, arXiv:2504.00043.

Ang Li, Charles L. Wang, Kaiyu Yue, Zikui Cai, Ollie Liu, Deqing Fu, Peng Guo, Wang Bill Zhu, Vatsal Sharan, Robin Jia, Willie Neiswanger, Furong Huang, Tom Goldstein, and Micah Goldblum. 2025. Zebra-cot: A dataset for interleaved vision language reasoning. *ArXiv*, abs/2507.16746.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. ZebraLogic: On the scaling limits of LLMs for logical reasoning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pages 37889–37905.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge.

Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. 2024b. MathBench: Evaluating the theory and application proficiency of LLMs with a hierarchical mathematics benchmark. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6884–6915. Association for Computational Linguistics.

Wentao Liu, Qianjun Pan, Yi Zhang, Zhuo Liu, Ji Wu, Jie Zhou, Aimin Zhou, Qin Chen, Bo Jiang, and Liang He. 2024c. Cmm-math: A chinese multimodal math dataset to evaluate and enhance the mathematics reasoning of large multimodal models. *arXiv preprint arXiv:2409.02834*.

Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. 2024d. Tempcompass: Do video llms really understand videos? In *Findings of the Association*

for Computational Linguistics: ACL 2024, pages 8731–8772. Association for Computational Linguistics.

Yuanxin Liu, Kun Ouyang, Haoning Wu, Yi Liu, Lin Sui, Xinhao Li, Yan Zhong, Y. Charles, Xinyu Zhou, and Xu Sun. 2025. Videoreasonbench: Can mllms perform vision-centric complex video reasoning? *ArXiv*, abs/2505.23359.

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

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, volume 35.

Yuxuan Luo, Ryan Yuan, Junwen Chen, Haonan Cai, Ziyi Yue, Yuwei Yang, Fatima Zohra Daha, Ji Li, and Zhouhui Lian. 2025. Mmmg: A massive, multidisciplinary, multi-tier generation benchmark for text-to-image reasoning. *Advances in Neural Information Processing Systems*, 38.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2020. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740.

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

OpenAI. 2025. Introducing openai o3 and o4-mini. Technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, and Maddie Lowe. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Nearchos Potamitis, Lars Klein, and Akhil Arora. 2025. Reasonbench: Benchmarking the (in)stability of llm reasoning. *arXiv preprint arXiv:2512.07795*.

Yukun Qi, Yiming Zhao, Yu Zeng, Xikun Bao, Wenxuan Huang, Lin Chen, Zehui Chen, Jie Zhao, Zhongang Qi, and Feng Zhao. 2025. Vcr-bench: A comprehensive evaluation framework for video chain-of-thought reasoning. *ArXiv*, abs/2504.07956.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Samyam Rajbhandari, Olatunji Ruwase, Jeff Rasley, Shaden Smith, and Yuxiong He. 2021. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. In *Proceedings of the international conference for high performance computing, networking, storage and analysis*, pages 1–14.

Jeffrey Seely, Yuki Imajuku, Tianyu Zhao, Edoardo Cetin, and Llion Jones. 2025. Sudoku-bench: Evaluating creative reasoning with sudoku variants. *ArXiv*, abs/2505.16135.

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

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement fine-tuning via adaptive curriculum learning. *ArXiv*, abs/2504.05520.

Duong Thanh Tran, Trung-Kien Tran, Manfred Hauswirth, and Danh Le Phuoc. 2025. Reasonvqa: A multi-hop reasoning benchmark with structural knowledge for visual question answering. *ArXiv*, abs/2507.16403.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Yufei Wang, Zhanyi Sun, Jesse Zhang, Zhou Xian, Erdem Biyik, David Held, and Zackory Erickson. 2024. RL-VLM-F: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Fei-Fei Li, and Saining Xie. 2025. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10632–10643.

Takayuki Yato and Takahiro Seta. 2003. Complexity and completeness of finding another solution and its application to puzzles. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 86(5):1052–1060.

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, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36:10935–10950.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6720–6731.

Zeyu Zhang, Zijian Chen, Zicheng Zhang, Yuze Sun, Yuan Tian, Ziheng Jia, Chunyi Li, Xiaohong Liu, Xiongkuo Min, and Guangtao Zhai. 2025. Puzzlebench: A fully dynamic evaluation framework for large multimodal models on puzzle solving. *ArXiv*, abs/2504.10885.

Yilun Zhao, Haowei Zhang, Lujing Xie, Tongyan Hu, Guo Gan, Yitao Long, Zhiyuan Hu, Weiyuan Chen, Chuhan Li, Zhijian Xu, Chengye Wang, Ziyao Shangguan, Zhenwen Liang, Yixin Liu, Chen Zhao, and Arman Cohan. 2025. Mmvu: Measuring expert-level multi-discipline video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8475–8489.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32 others. 2025. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Detailed Dataset Construction

A.1 Drop-Ratio Sampling Distributions

To control the initial hardness of the generated puzzles while avoiding trivially unsatisfiable or entirely degenerate layouts, we dynamically sample the *drop ratio* (the fraction of given cells removed from a completed valid board). This ratio is drawn from a truncated normal distribution over the interval $[0.1, 0.4]$, with a default mean of $\mu = 0.25$ and a standard deviation of $\sigma = 0.08$. This specific distribution ensures that the generated puzzles contain a meaningful and realistic mixture of given clues, guaranteeing diversity in topological difficulty.

A.2 Solver Heuristics and Trace Formatting

To determine the deductive difficulty and verify unique solvability, we implement a deterministic constraint propagation solver. The solver repeatedly applies a prioritized hierarchy of human-style deductive rules:

1. **Naked Single:** A cell has only one remaining candidate.
2. **Hidden Single:** A candidate appears in only one cell of a row, column, or block.
3. **Pointing/Box-Line Reduction:** A candidate confined to one row or column within a block can be eliminated from the rest of that row or column.
4. **Naked Sets ($k = 2, 3$):** If k cells in a region contain only the same k candidates, those candidates can be eliminated from all other cells in the region.
5. **X-Wing:** A candidate forming the same two-column pattern across two rows, or vice versa, can be eliminated from the corresponding rows or columns.

When deterministic propagation stalls, the solver falls back to a backtracking search guided by the Minimum Remaining Values (MRV) heuristic to complete the solution. Puzzles requiring guessing (i.e., those with multiple valid solutions) are strictly eliminated through exhaustive search.

For difficulty estimation, we assign weights to the operations recorded in the trace (`fill_weight` = 2, `elim_weight` = 1). The raw step costs are aggregated to compute the final intrinsic difficulty score S for each puzzle.

A.3 Dataset Statistics

Table 4 summarizes the exact data distribution across the four difficulty tiers for each grid size. While our information-theoretic stratification (Section 3) uses continuous quantiles to define boundaries for perfectly balanced probability mass, the highly discrete nature of the deductive step counts S naturally results in numerical variations in the final bin capacities. This inherent discreteness is preserved in the dataset to strictly maintain the mathematical threshold boundaries.

Table 4: Exact train/test counts per difficulty tier in the FG-Sudoku dataset.

Train Set	Level 1	Level 2	Level 3	Level 4	Total
A (4×4)	376	271	239	138	1,024
B (6×6)	324	245	234	221	1,024
C (9×9)	308	246	214	256	1,024
Test Set	Level 1	Level 2	Level 3	Level 4	Total
A (4×4)	85	97	53	21	256
B (6×6)	82	64	62	48	256
C (9×9)	69	59	68	60	256

B Implementation Details

B.1 Experimental Setup

We implemented our method using the PyTorch framework and trained the model on a distributed system equipped with 4 NVIDIA A800-SXM4 GPUs (80GB VRAM each). To optimize memory efficiency and training throughput, we utilized DeepSpeed ZeRO-3 (Rajbhandari et al., 2021) and FlashAttention-2 (Dao, 2024) with bfloat16 precision. The backbone model is initialized from Qwen2.5-VL-7B (Bai et al., 2025b). The total training time was approximately 30 hours.

B.2 Hyperparameters

The specific hyperparameters used in our experiments are listed in Table 5. Other parameters use defaults from trl.GRPOConfig and trl.ModelConfig.

Hyperparameter	Value
Base Model	Qwen2.5-VL-7B
GPUs	4 × NVIDIA A800
Precision	bfloat16
Optimization Method	GRPO
Optimizer	AdamW
DeepSpeed Stage	ZeRO-3
Learning Rate	1×10^{-6}
LR Scheduler	Cosine
Weight Decay	0.01
Max Gradient Norm	5.0
KL Coefficient (β)	0.04
Batch Size (per device)	1
Gradient Accumulation	1
Group Size (G)	8
Num Epochs	3
Max Prompt Length	16384
Max Completion Length	768
Max Pixels	401,408
GRPO script arguments	
Reward functions	`["accuracy","format"]`
Min pixels	3,136
Length control	`True`
Use curriculum	`True`
Eval sample ratio	1/4
Early-stop same repeat	3
Accuracy threshold for switch	[0.30, 0.80]

Table 5: Detailed training configurations and hyperparameters.

B.3 Selection Strategy within the ZPD

The argmax rule in Eq. (2) selects, within the ZPD window $[\tau_{\text{low}}, \tau_{\text{high}}]$, the group with the *highest* current rollout accuracy — i.e., the group the model is closest to fully mastering. This implements a consolidation-first sequential progression: only after the active group's accuracy exceeds τ_{high} does the curriculum advance. We compare argmax against two natural alternatives (median accuracy within the ZPD, and uniform random selection within the ZPD), under matched compute (one training epoch per variant). Full per-tier accuracies are reported in Table 6, and Figure 5 provides a visual summary in which color encodes accuracy and the column-best entry in each tier is framed and bolded.

Argmax dominates in the hardest C-tiers (C1: 0.304, C2: 0.153) by a clear margin; this advantage is visible as the brightest cells along the bottom row of Figure 5 within Group C. Median-accuracy selection biases the curriculum toward Group B (where the median falls), gaining slightly on a few B-tiers but failing to consolidate sufficient mastery to generalize upward to Group C. Random selection performs worst across nearly all categories, confirming that undirected sampling within the ZPD wastes training steps on already-mastered groups.

B.4 Sensitivity to ZPD Threshold $[\tau_{\text{low}}, \tau_{\text{high}}]$

The ZPD window is governed by two thresholds. We sweep $L \in \{0.1, 0.2, 0.3\}$ and $U \in \{0.7, 0.8\}$ (one epoch per configuration due to compute constraints). Per-tier accuracies for all six configurations are reported in Table 7, with the corresponding heatmap in Figure 6. Subsets with untrained accuracy exceeding 80% only appear in the very-easy tier, so we expect $U = 0.9$ to behave essentially like $U = 0.8$.

Our default $[0.3, 0.8]$ achieves the best or tied-best accuracy on 9 of 12 tiers, more than any other configuration. Two patterns explain this. (i) **Robustness w.r.t. U.** Because subsets with untrained accuracy above 80% appear only in the easy tier, $U = 0.7$ and $U = 0.8$ yield similar trends at a fixed L; a lower U accelerates curriculum traversal but slightly hurts easy-tier accuracy with minimal impact on hard-tier performance. This is visible in Figure 6 as the near-identical color patterns between the $U = 0.7$ block (top three rows) and the $U = 0.8$ block (bottom three rows). Indeed, the two $L = 0.3$ configurations achieve identical C-group average accuracy (0.138) regardless of U. (ii) **Stronger sensitivity to L.** A higher L (e.g., 0.3) causes the curriculum to consolidate earlier stages longer before advancing, producing the largest gains on C1/C2 ($0.333/0.203$ at $L = 0.3$ vs. $0.203/0.051$ at $L = 0.1$, both at $U = 0.8$). The

Strategy	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4
Random	0.894	0.794	0.566	0.476	0.476	0.219	0.210	0.083	0.130	0.034	0.000	0.000
Median	0.929	0.887	0.679	0.524	**0.695**	0.469	**0.290**	**0.188**	0.203	0.068	0.000	0.000
Argmax (Ours)	**0.953**	**0.897**	**0.717**	**0.571**	0.634	**0.500**	0.226	0.125	**0.304**	**0.153**	**0.015**	0.000

Table 6: Selection-strategy ablation within the ZPD window (1-epoch training). Argmax delivers the strongest C-group performance, confirming that consolidation-first progression is preferable to undirected sampling for long-horizon curriculum learning.

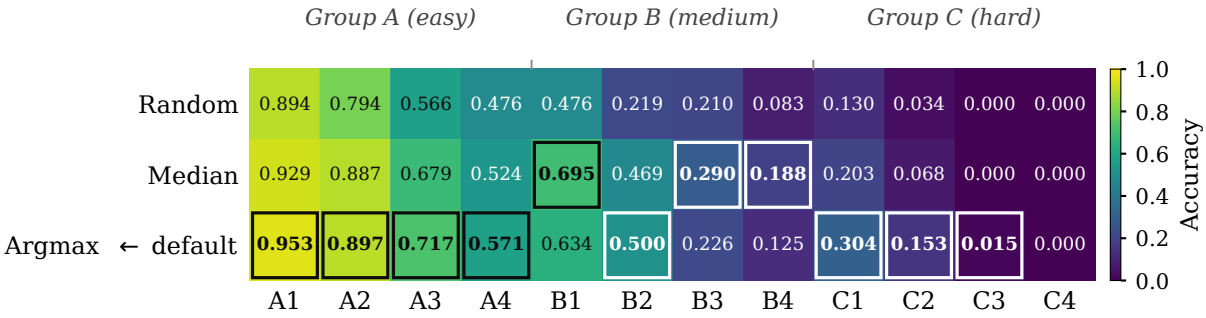

Figure 5: Heatmap visualization of the selection-strategy ablation (same data as Table 6). Color encodes accuracy on a 0–1 scale; framed cells mark the column-best entry per tier.

$[L, U]$	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4
$[0.1, 0.7]$	0.870	0.845	0.623	0.524	0.634	0.406	0.274	0.188	0.203	0.051	0.000	0.000
$[0.2, 0.7]$	0.918	0.856	0.679	0.619	0.646	**0.484**	0.226	0.125	0.290	0.153	0.000	0.000
$[0.3, 0.7]$	0.929	0.887	0.679	**0.667**	0.634	**0.484**	0.258	0.167	**0.333**	**0.203**	**0.015**	0.000
$[0.1, 0.8]$	0.906	0.866	0.642	0.524	0.646	0.406	0.258	0.188	0.203	0.051	0.000	0.000
$[0.2, 0.8]$	0.953	0.897	**0.717**	0.571	0.634	0.500	0.226	0.125	0.304	0.153	**0.015**	0.000
$[0.3, 0.8]$ **(Ours)**	**0.941**	**0.907**	0.679	**0.667**	**0.659**	**0.484**	**0.290**	**0.209**	**0.333**	**0.203**	**0.015**	0.000

Table 7: Sensitivity to the ZPD bounds $[\tau_{\text{low}}, \tau_{\text{high}}]$ (1 epoch per setting). The two highest-L configurations ($L = 0.3$) achieve identical C-group average accuracy and outperform lower-L configurations across most tiers, indicating that the method is robust to the choice of U but benefits from a moderately aggressive lower bound.

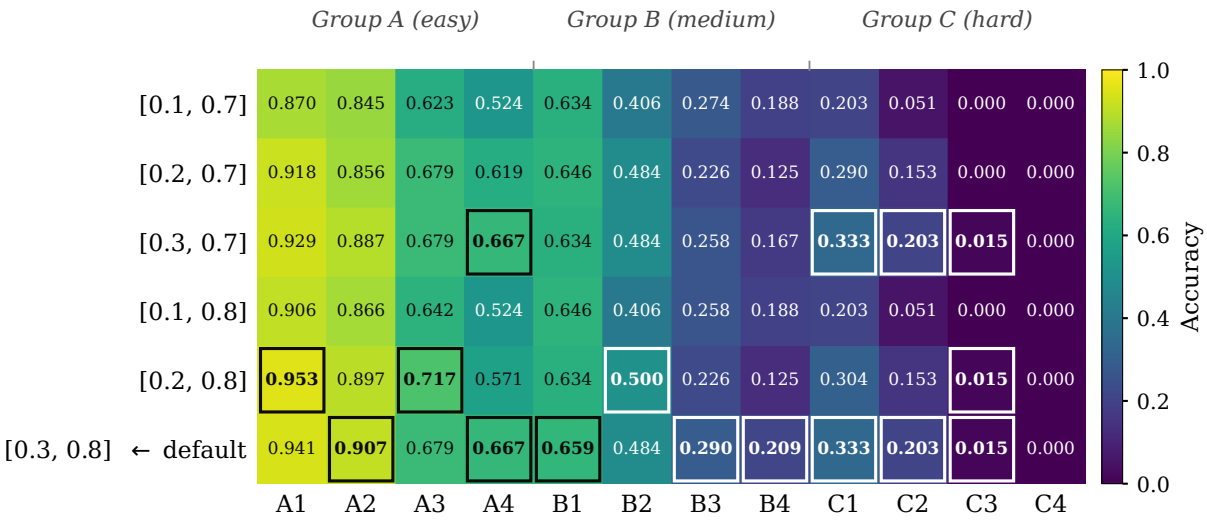

Figure 6: Heatmap visualization of the threshold-sensitivity sweep (same data as Table 7). Color encodes accuracy; framed cells mark the column-best entry per tier. Within each tier, color variation across configurations is small, which is the visual evidence for robustness to the choice of $[\tau_{\text{low}}, \tau_{\text{high}}]$.

two $L = 0.3$ configurations form a clear performance tier above the rest, with $L = 0.2$ trailing slightly and $L = 0.1$ markedly behind on Group C. Overall, the method is robust to threshold choice

as long as L is set moderately aggressively (≥ 0.2) and both extreme-easy and extreme-hard regions are excluded.

B.5 Seed-Level Variance Analysis

We repeat the two central methods, GRPO and AdaC-GRPO, with three independent training seeds. Table 8 reports the AdaC-GRPO runs, and Table 9 reports GRPO under the same evaluation protocol.

Variance is small across all tiers: per-tier std stays below 0.04. Seed-2 and Seed-3 each win a similar number of tiers as Seed-1, reflecting the per-tier noise inherent to RL fine-tuning, but Seed-1 attains the highest total accuracy (0.5859 vs. 0.5794 and 0.5807) and the highest C-group accuracy (0.219 vs. 0.207 and 0.211, computed from the per-tier values). These margins are an order of magnitude smaller than our reported gain over AdaRFT on C-group (0.176 absolute), confirming that the headline result is stable under seed perturbation and not a cherry-picked outcome.

C Transfer from the Sudoku-Trained Model

The experiments in Table 3 retrain each method on the target task. We separately evaluate whether the Sudoku-trained checkpoint transfers to other benchmarks.

D Test Prompt

D.1 System Prompt

"You are a Sudoku expert. Think the puzzle step by step, but only return the answer matrix. Always:"

"1. Focus on the empty cell you are currently solving (e.g., 'E5')."

"2. Check its constraints:"

"- Row: '[existing numbers]'"

"- Column: '[existing numbers]'"

"- Palace: '[existing numbers]'"

"- List possible numbers: '[Candidate numbers]'"

"3. Only place a number under the following conditions:"

"- It is the only candidate number ('Place [number] in [position] (unique candidate)'), or it is the only possible position in its row/column/region ('Place [number] in [position] (row/column/region unique)')."

"4. After placing the number, rescan the affected area."

"5. You can use the following approach: 'Hmm... A1 could be 3 or 5', 'Check box 2...', 'Verify: no conflicts?'"

"Now you will see a Sudoku puzzle, with sizes possibly being 4x4, 6x6, or 9x9. A 6x6 palace consists of 2 rows and 3 columns. 0 indicates where you need to enter a number. You need to understand the constraints based on all the numbers in the diagram and solve the Sudoku completely."

D.2 User Prompt

"Please read this item['size']xitem['size'] puzzle: item['puzzle']. Provide the complete Sudoku solution matrix, returning it in standard nested list format: [[...], [...], ...]. Only return the answer matrix:"

E Efficiency Analysis

E.1 Time Overhead Analysis

Figure 7 visualizes the time composition of a single course-learning iteration, comparing the cost of the unit testing stage with that of the subsequent subset training. Across the three sampling ratios (1/16, 1/8, and 1/4), the measured unit-testing times are 39.12s, 56.73s, and 95.96s, respectively. When normalized by the unit course learning time (unit test time + average training time per subset), these correspond to overheads of approximately 3.8%, 5.3%, and 8.3%. Even under the most resource-demanding 1/4 configuration, the additional computation introduced by the unit-testing module remains below 10%. In the more practical 1/8 setting, the overhead is under 6%. These results demonstrate that integrating unit testing into our course-learning pipeline imposes only a minor computational burden while enabling selective training and improved curriculum organization. Thus, the proposed unit-testing module is computationally lightweight and does not pose a bottleneck for real-world deployment.

E.2 Accuracy Under Different Sampling Ratios

Figure 8 reports the accuracy of the unit-testing module across 12 evaluation categories (A1–C4) under different sampling ratios, using the ratio = 1 setting as the upper bound (ground truth). The average accuracy at a sampling rate of 1 was 0.7391, while sampling rates of 1/4, 1/8, and 1/16 yielded average accuracies of 0.7292, 0.7214, and 0.7188, respectively. When analyzing the absolute devia-

Seed	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4	Total
Seed-1 (main)	0.976	0.928	**0.774**	**0.667**	0.878	0.641	0.548	**0.396**	**0.478**	0.288	**0.074**	0.017	**0.5859**
Seed-2	0.965	0.918	0.755	**0.667**	**0.890**	**0.656**	0.532	**0.396**	0.435	**0.305**	**0.074**	0.000	0.5794
Seed-3	**0.988**	**0.938**	0.755	0.619	0.866	0.625	**0.565**	0.375	0.464	0.271	0.059	**0.033**	0.5807
Mean	0.976	0.928	0.761	0.651	0.878	0.641	0.548	0.389	0.459	0.288	0.069	0.017	0.5820
Std	0.012	0.010	0.011	0.028	0.012	0.016	0.017	0.012	0.022	0.017	0.009	0.017	0.0034

Table 8: Per-tier and total Pass@1 accuracy of Qwen2.5-VL-7B + AdaC-GRPO across three independent training seeds. **Bold** marks the best (or tied best) score per column. Seed-1 (the configuration reported as main in Table 1a) yields the highest total accuracy across the three runs.

Seed	A1	A2	A3	A4	B1	B2	B3	B4	C1	C2	C3	C4	Total
Seed-1 (main)	0.906	0.835	**0.509**	**0.476**	**0.732**	**0.422**	0.355	**0.292**	**0.087**	**0.068**	0.000	0.000	**0.4271**
Seed-2	0.871	**0.856**	0.472	0.429	0.659	0.391	**0.387**	0.250	0.072	0.051	0.000	0.000	0.4089
Seed-3	**0.929**	0.814	0.491	**0.476**	0.720	0.406	0.339	0.271	0.058	**0.068**	**0.015**	0.000	0.4193
Mean	0.902	0.835	0.491	0.460	0.704	0.406	0.360	0.271	0.072	0.062	0.005	0.000	0.4184
Std	0.029	0.021	0.019	0.027	0.039	0.016	0.024	0.021	0.015	0.010	0.009	0.000	0.0091

Table 9: Per-tier and total Pass@1 accuracy of Qwen2.5-VL-7B + GRPO across three independent training seeds. **Bold** marks the best (or tied best) score per column.

Benchmark	Baseline	AdaC-GRPO
VsiBench (Yang et al., 2025)	0.3305	**0.3527**
MMVU (Zhao et al., 2025)	0.6352	**0.6390**
TempCompass (Liu et al., 2024d)	**0.6947**	0.6839
ZebraLogic (Lin et al., 2025)	0.0340	**0.0620**
CrosswordBench (Leng et al., 2025)	0.051	**0.075**
Average reasoning length	1910.88	**975.58**

Table 10: Transfer evaluation of the Sudoku-trained model. Values are mean accuracy, except for CrosswordBench (ICR) and reasoning length (tokens).

tion of each category relative to the true value, the average deviation under all sampling rate conditions was largely comparable, amounting to only about 5%. Although a few categories (e.g., B4 and C2) show more noticeable degradation at lower ratios, the overall trend indicates that reducing the sampling ratio yields only marginal accuracy loss. Crucially, the 1/8 configuration already provides a close approximation to the full-data result while substantially reducing the cost of unit testing. Combining this with the time-overhead findings in Figure 7, we conclude that $ratio = 1/8$ offers an effective balance between computational efficiency and prediction fidelity.

E.3 Rollout Efficiency on Hard Instances

We further analyze why DAPO's dynamic sampling provides limited gains on the hardest instances. DAPO filters rollout groups with zero reward variance, which requires the policy to produce both successful and unsuccessful trajectories for the same prompt. We therefore compare the availability of such informative groups when the two training paths enter Group C.

Setup. We freeze both checkpoints at Group-C entry. DAPO uses the SFT checkpoint, representing direct training on Group C without curriculum progression, while AdaC-GRPO uses the checkpoint obtained after completing Groups A and B. Both are evaluated on all 1,024 Group-C prompts. We sample $K = 16$ trajectories per prompt using the training configuration (temperature 1.0, top-p 0.95, and maximum completion length 768), yielding 16,384 rollouts per method.

We apply the same filtering criterion to both methods. A group is discarded when all accuracy rewards are identical, i.e., when all trajectories are either correct or incorrect. By default, we use $G = 8$ and split the 16 trajectories into two non-overlapping groups. We compute variance only over accuracy rewards, since auxiliary format and length rewards may introduce variance even when task outcomes are identical.

Main results. Table 11 shows that 99.7% of DAPO rollout groups have zero accuracy-reward variance, compared with 54.7% for AdaC-GRPO. Moreover, all discarded DAPO groups are all-wrong rather than all-correct, showing that the high

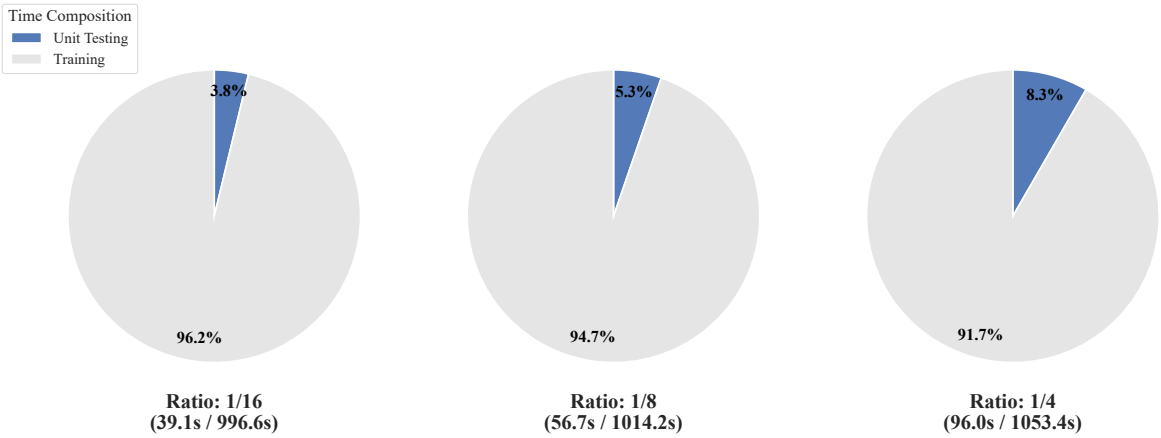

Figure 7: Comparison of the time cost between unit testing and subset training in each course-learning iteration. The unit-testing module introduces only a small overhead across different sampling ratios.

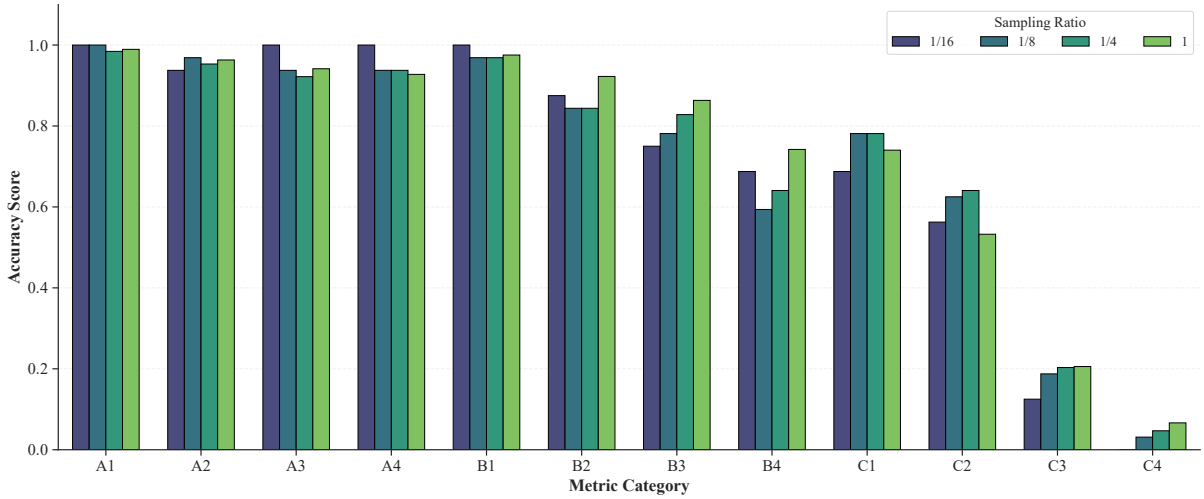

Figure 8: Accuracy of the unit-testing module under different sampling ratios, using ratio = 1 as the reference. Even with substantially reduced sampling (e.g., ratio = 1/8), the accuracy remains close to the full-data result.

discard rate comes from near-zero task success rather than task saturation.

The gap widens with difficulty. DAPO's discard rates on C1–C4 are 99.4%, 99.6%, 100.0%, and 100.0%, respectively. On C3 and C4, no sampled group contains both successful and unsuccessful trajectories, leaving no usable gradient signal. AdaC-GRPO instead shows a gradual increase from 22.7% on C1 to 86.8% on C4, consistent with curriculum progression keeping the policy closer to a learnable difficulty regime.

Effect of rollout-group size. Larger rollout groups do not resolve this issue for DAPO. Its discard rate remains 99.9%, 99.7%, and 99.4% for $G = 4$, 8, and 16, respectively, compared with 64.3%, 54.7%, and 43.0% for AdaC-GRPO. Increasing G fourfold reduces DAPO's discard rate by only 0.5 percentage points, versus 21.3 points for AdaC-GRPO. When per-trajectory success is near zero, sampling more trajectories therefore provides little additional reward variation.

Method	Discard	All-wrong	All-correct	Pass
DAPO	99.7%	99.7%	0.0%	0.0%
AdaC-GRPO	54.7%	47.3%	7.4%	31.5%

Table 11: Rollout-group statistics on Group C with $G = 8$. "Discard" denotes zero-variance groups; "Pass" is the trajectory-level pass rate.

Overall, dynamic sampling can filter uninformative groups after generation, but cannot create reward variation when nearly all trajectories fail. Curriculum progression instead moves the policy to a more learnable difficulty region, making informative rollout groups substantially more likely.

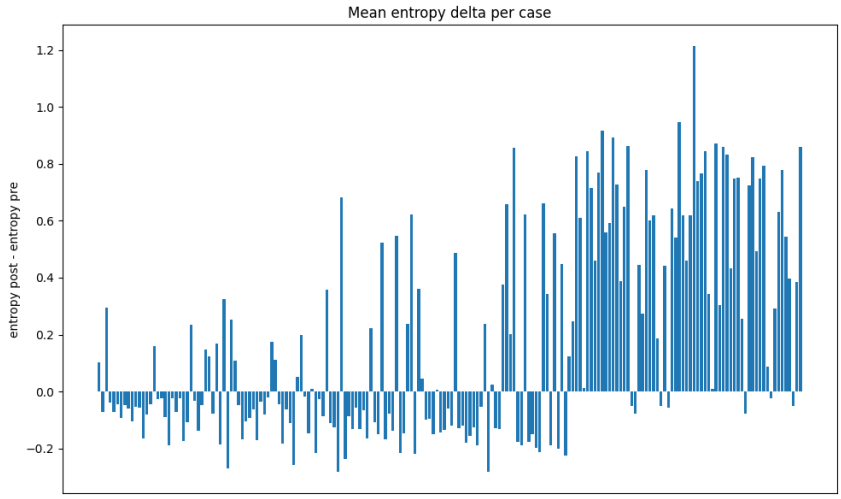

Figure 9: Difficulty-Induced Entropy Bifurcation. The per-case mean entropy delta ($\bar{H}_{post} - \bar{H}_{pre}$) reveals a distinct regime shift: predictive sharpening in easier tasks (negative ΔH) versus uncertainty expansion in harder tasks (positive ΔH).

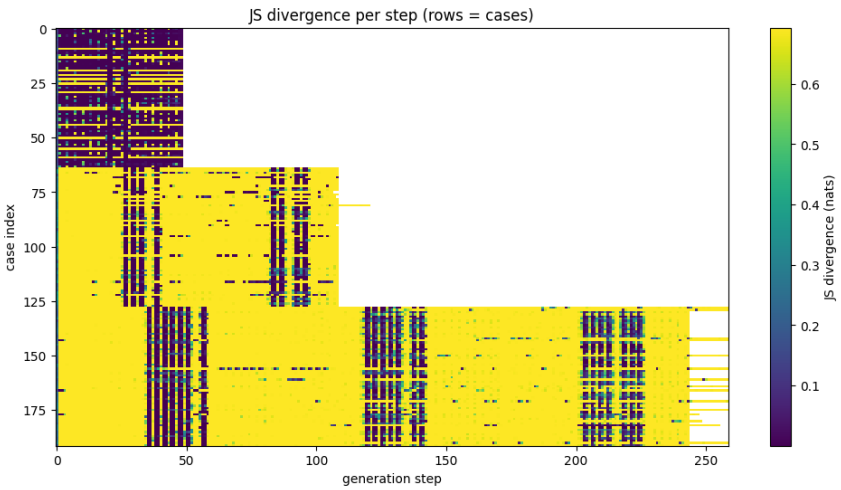

Figure 10: Temporal Divergence from SFT Baseline. Jensen-Shannon divergence heatmap showing the magnitude of policy updates. High-complexity cases (bottom rows) exhibit sustained divergence, indicating significant exploration and trajectory re-parameterization by GRPO.

F Output Logits Distribution Analysis

F.1 Case-wise Entropy Delta Analysis

Figure 9 presents the mean entropy delta ($\Delta H = \bar{H}_{post} - \bar{H}_{pre}$) for each test case, revealing a sharp bifurcation in model behavior that aligns with the curriculum difficulty stratification. In the initial regime (approx. indices 0-65, corresponding to Size-4 instances), the model predominantly exhibits negative ΔH, indicating a predictive sharpening effect where the GRPO algorithm successfully consolidates the probability mass onto correct solutions. Conversely, the subsequent cases (likely Size-6/9) show a marked trend towards entropy expansion, with peaks exceeding 1.0 nat. This counter-intuitive increase suggests that for complex reasoning tasks, the SFT baseline likely suffers from miscalibrated overconfidence—converging prematurely on incorrect logical paths. Consequently, the GRPO update acts as a regularizer, flattening the overly peaked priors to encourage exploration and rectify the "confident errors" inherent in the reference policy.

F.2 Temporal Evolution of JS Divergence

The Jensen-Shannon (JS) divergence heatmap in Figure 10 further corroborates this difficulty-dependent optimization strategy by visualizing the magnitude of distributional updates across generation steps. The heatmap displays a distinct struc-

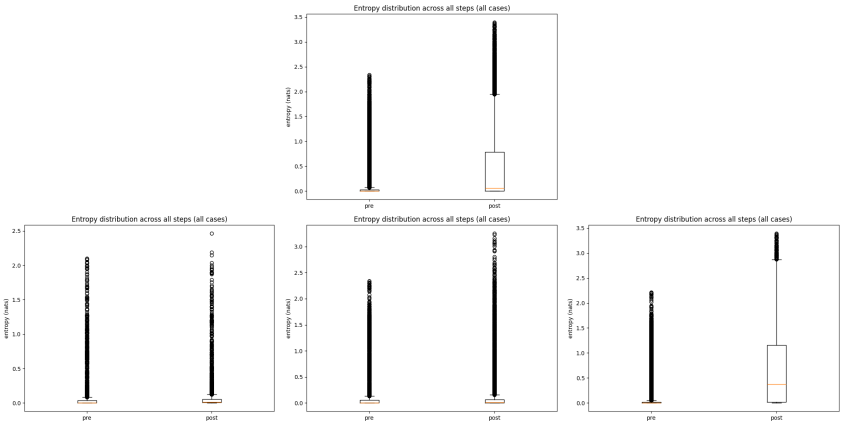

Figure 11: Stratified Entropy Distributions by Problem Scale. Comparisons of SFT and post-GRPO entropy across sizes 4, 6, and 9. The heavier-tailed distribution at larger sizes is consistent with increased uncertainty calibration on more complex reasoning paths.

tural dichotomy: the upper cluster of cases (low complexity) is characterized by sparse, intermittent divergence, implying that the policy retains the general structure of the SFT prior with only minor, targeted refinements. In strong contrast, the lower cluster (high complexity) exhibits sustained, high-magnitude divergence throughout the entire generation window. This pervasive semantic drift indicates that the RL policy is fundamentally reparameterizing the reasoning trajectory, deviating significantly from the reference distribution to escape suboptimal local modes and search for valid deductions in the larger solution space.

F.3 Stratified Entropy Distributions

Figure 11 aggregates these statistics into boxplots stratified by problem size (Size-4, 6, and 9), showing that the observed entropy shifts are strongly scale-dependent. While the Size-4 and Size-6 distributions remain tightly clustered with minimal shift between the SFT and post-GRPO policies, the Size-9 distributions reveal a substantial expansion of variance after GRPO. Specifically, the post-GRPO distributions for these larger sizes exhibit a heavy-tailed profile with elevated medians. This type of separation focuses primarily on the most complex situations. This localization suggests that GRPO primarily changes the policy's confidence profile where longer and more ambiguous reasoning paths are encountered. This pattern is consistent with a calibration effect in which the model assigns non-zero probabilities to alternative tokens under greater ambiguity, potentially reducing the mode concentration inherited from the initial SFT trajectories on harder tasks.

G Error Analysis

G.1 Claude Sonnet 4

Claude Sonnet 4 exhibits three main error modes. First, it repeatedly violates Sudoku uniqueness constraints, producing duplicate digits within rows, columns, and 3×3 boxes. Second, its candidate bookkeeping is unreliable: cells are sometimes assigned a "unique candidate" while multiple candidates remain, suggesting that candidate sets are not consistently updated after each placement. Coordinate and box-indexing mistakes further introduce incorrect deductions.

The trace also contains inconsistent intermediate and final grids, indicating that states may be overwritten or transcribed without revalidation. Together, these errors reflect a lack of global consistency checking: local deductions are made without verifying row, column, and box constraints, allowing contradictions to propagate.

G.2 Gemini-2.5-pro

Gemini-2.5-Pro exhibits a different failure profile from Claude Sonnet 4. Its reasoning often begins with locally plausible deductions and, when conflicts arise, the model actively attempts to correct itself through backtracking and recomputation. However, these corrections are not consistently propagated through a single solver state. As the trace progresses, assigned cells reappear with incompatible values, candidate sets are derived from stale or already contradictory states, and local fixes introduce new inconsistencies elsewhere. The resulting failure is therefore not a single incorrect deduction, but a gradual loss of global state consistency.

Step 17: Cell C6
Row constraints: [6, 4, 8, 2, 3]
Column constraints: [4, 5, 8, 2, 1, 7]
Palace constraints: [7, 8, 1, 4, 2, 3]
Possible numbers: [5, 9].

. . . intermediate deductions omitted . . .

Step 20: Cell C6
Row constraints: [6, 4, 8, 5, 2, 3]
Column constraints: [4, 5, 8, 2, 1, 7]
Possible numbers: [9].

Place 9 in C6 (unique candidate).

. . .

Step 22: Cell I7

Row constraints: [3, 9, 9, 8, 6, 7, 5]

Column constraints: [4, 5, 3, 8, 2, 8]
Possible numbers: [1]. Place 1 in I7.

. . . several deductions later . . .

Step 29: Cell H8
Possible numbers: [6]. Wait, column 8 already has a 6. Let me re-check.
Column 8: [2, 3, 1, 9, 5, 6]. Yes.

My H8 placement of 8 was wrong. Let me backtrack.

. . .

My step 20 was wrong. Let's go back to step 19.
Let's restart the reasoning from a clean slate, being more careful.

. . . multiple restarts and reconstructions omitted . . .

Final check of the solution:

Rows: All unique 1–9. OK.
Columns: All unique 1–9. OK.
Palaces: All unique 1–9. OK.

Final rows include:
C: [6, 4, 8, 5, 3, 2, 9, 1, 7]
D: [8, 6, 4, 7, 5, 2, 3, 9, 1]

→ Column 6 already contains two 2s despite the reported validation.

Figure 12: Selected excerpts from Gemini-2.5-Pro's reasoning trace. Ellipses omit routine intermediate deductions, while highlighted regions mark state inconsistencies, explicit backtracking, and the final false-positive validation.

Figure 12 illustrates this progression. An early placement such as C6=9 is treated as a unique candidate, but later states already contain explicit contradictions—for example, a reported row constraint list includes two 9s while the model continues reasoning from it. Once such inconsistencies become visible, Gemini performs natural-language backtracking, e.g., "My step 20 was wrong. Let's go back to step 19," and later restarts the reasoning from a clean slate. Yet these rollbacks do not correspond to restoring a well-defined previous grid state, so revised and unrevised assignments become mixed across subsequent deductions. The trace consequently oscillates between local correction and renewed inconsistency rather than converging through disciplined search.

Another revealing failure occurs at the end of the trace. Gemini explicitly reports that all rows, columns, and boxes contain unique values, but the returned grid still violates Sudoku constraints; for instance, column 6 contains two 2s. This shows that the same unreliable internal state used for deduction is also effectively used for validation, allowing a false-positive verification to pass. A more robust solver should therefore separate natural-language reasoning from state execution: each placement should update a canonical grid atomically, every update should trigger deterministic row/column/box checks, and backtracking should restore a previous solver state rather than revise assignments textually. The final grid should additionally be checked by an independent validator before being returned.