BLENDPO: CONTROLLING GUIDANCE STRENGTH IN REINFORCEMENT LEARNING WITHOUT CHANGING GUIDANCE CONTENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards provides no learning signal when sampled rollouts fail to produce a correct solution. A common approach is to condition rollouts on guidance, such as partial solutions or hints, during training. Stronger guidance makes successful rollouts more likely but can transfer less to the unguided setting used at test time, making guidance strength a key design choice. Solution prefixes allow fine-grained control by adjusting the prefix length. In contrast, for guidance without a natural ordering, such as abstract hints, the guidance strength is typically controlled only through coarse choices, such as selecting a hint level or deciding whether to provide a hint at all. We propose *BlendPO* (Blended-context Policy Optimization), which enables fine-grained control of guidance strength without changing guidance content by interpolating policy logits obtained with and without guidance. Rather than training with a single guidance strength, BlendPO samples responses at multiple interpolation strengths for each problem and updates the policy using these responses under the original unguided context. We first evaluate BlendPO on mathematical reasoning using abstract hints generated from reference solutions. BlendPO outperforms methods based on solution prefixes as well as those using the same abstract hints. We then extend BlendPO to visual reasoning with *visual states*, textual descriptions of the visual input that can be obtained from dataset metadata or the task environment without reference solutions. Across four visual reasoning task families, BlendPO consistently improves over GRPO and methods using the same visual states, demonstrating its potential to broaden the practical applicability of guided RL when reference solutions are unavailable.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has driven substantial progress in mathematical reasoning and code generation by training large language models on self-generated responses using rule-based rewards (Shao et al., 2024; DeepSeek-AI, 2025; Yu et al., 2025). However, this learning process becomes challenging on problems where the model rarely samples a correct response. In GRPO, a group of responses with identical rewards yields zero group-relative advantages and therefore no reward-based learning signal (Shao et al., 2024; Yu et al., 2025). To obtain a learning signal on such problems, a common approach is to provide additional, training-only privileged information (Vapnik & Vashist, 2009), such as a reference solution or a hint derived from it, in the rollout context to increase the likelihood of generating successful responses (Nath et al., 2025; Qu et al., 2026). We refer to RL that uses such information as *guidance* for rollout generation as *guided RL*.

Stronger guidance makes successful rollouts more likely but can transfer less to the unguided setting in which the policy is evaluated. Prior methods therefore control guidance strength to balance success and transfer. Methods using solution-prefix guidance enable fine-grained control of guidance strength by adjusting the prefix length (Zhang et al., 2025b; Amani et al., 2025; Qu et al., 2026; Setlur et al., 2026; Agrawal et al., 2026). Because a solution is an ordered sequence of reasoning steps, a shorter prefix reveals fewer steps and thus provides weaker guidance. Alternatively, abstract hints have been proposed to keep guided rollouts closer to the policy's own generation by providing

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.

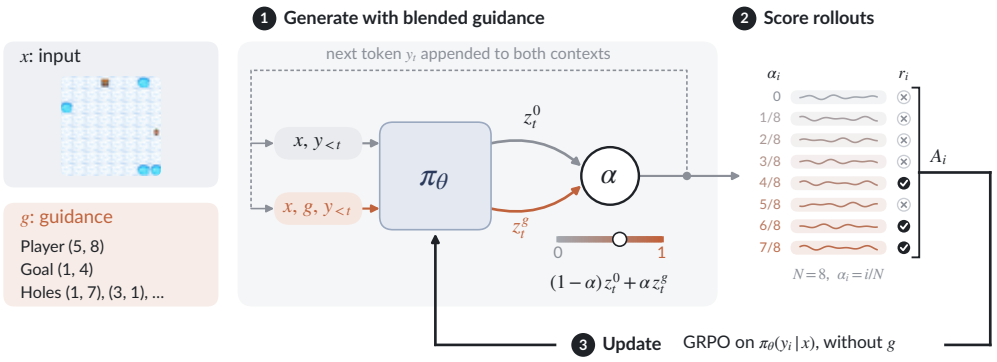


Figure 1: BlendPO training loop, illustrated with a maze input x and its visual state g. (1) The same policy computes next-token logits under the original context $(x, y_{<t})$ and the guided context $(x, g, y_{<t})$. BlendPO samples from the blended logits $(1-\alpha)z_t^0 + \alpha z_t^g$ and appends the sampled token to both contexts. (2) Responses to the same problem use strengths $\alpha_i = i/N$, fixed within each response, and receive verifier rewards r_i. (3) The group-relative advantages A_i update $\pi_\theta(y_i \mid x)$ with a GRPO surrogate without g, and inference uses x only. The rewards shown are illustrative.

only a concept or strategy rather than a solution trace (Wang et al., 2026b) (Figure 2a). However, success under hints may still not transfer to the unguided setting. In our experiments, providing hints without any strength control even degrades performance below that of the base model (Appendix D.1). Their strength therefore also needs to be controlled, but unlike solution prefixes, it is controlled only through coarse choices. HINT provides such hints only when all unguided responses in a group fail (Wang et al., 2026b), and Scaf-GRPO chooses among three levels of hints with increasing detail (Zhang et al., 2025a), which requires writing several hints for each problem. For guidance without a natural order, such as hints and input descriptions, strength cannot be reliably controlled through length, unlike for solution prefixes. Shortening them may remove a key relation and make the guidance nearly useless, or remove a redundant detail with little effect.

To enable fine-grained strength control for guidance without a natural order, we introduce *BlendPO* (Blended-context Policy Optimization), which separates guidance strength from guidance content. BlendPO interpolates the same policy's next-token logits under the original and guided contexts using a strength $\alpha \in [0, 1]$ and samples from the blended logits (Figure 1). Varying α from unguided ($\alpha = 0$) to fully guided ($\alpha = 1$) generation enables continuous control of guidance strength. We update the policy under the original context using a GRPO surrogate on responses generated with blended guidance. We consider *fixed-strength* allocation, which uses one strength for all responses, and a *within-group sweep*, which assigns different strengths across responses to the same problem, exposing each problem to multiple guidance strengths within a single group.

This control is especially useful in tasks without reference solutions, where solution prefixes cannot be constructed and the available guidance has no natural order. Visual reasoning is one such setting. Here, guidance can take the form of a *visual state*, a textual description of the objects, attributes, and relations in the visual input. This can be obtained from dataset metadata or the task environment without solving the problem. Examples include formal-language descriptions of geometry diagrams, captions of visual puzzles, object relations in spatial reasoning, and symbolic states in visual planning (Figure 2b–f). Even when neither a reference solution nor a visual state is available, constructing such guidance is typically cheaper than writing a reference solution.

We evaluate BlendPO in two settings. In mathematics, where reference solutions are available and existing strength controls can be compared, we use abstract hints generated from those solutions. BlendPO raises mean unguided MATH accuracy from GRPO's 74.83% to 79.40% and outperforms guided-RL baselines that use either solution-prefix guidance or the same hints. In visual reasoning, BlendPO using visual states improves unguided accuracy over GRPO and other baselines on geometry, visual puzzles, spatial reasoning, and visual planning, for example from 19.00% to 75.33% on 8×8 mazes. Ablations on 8×8 mazes show that both intermediate strengths enabled by logit blending and within-group strength variation contribute to these gains.

Our contributions are as follows.

- We introduce *BlendPO*, which enables fine-grained control of guidance strength through logit blending without modifying guidance content, and uses a within-group sweep to expose each problem to multiple guidance strengths (Section 3.2).
- In mathematics, BlendPO with abstract hints outperforms GRPO, solution-prefix methods, and methods using the same hints, and its gain persists within GRPO's estimated training-compute budget (Sections 4.2 and 4.4).
- Using visual states as guidance, BlendPO consistently improves over GRPO and baselines using the same visual states across four visual reasoning task families, demonstrating its potential to broaden the practical applicability of guided RL when reference solutions are unavailable (Sections 3.4 and 4.3).

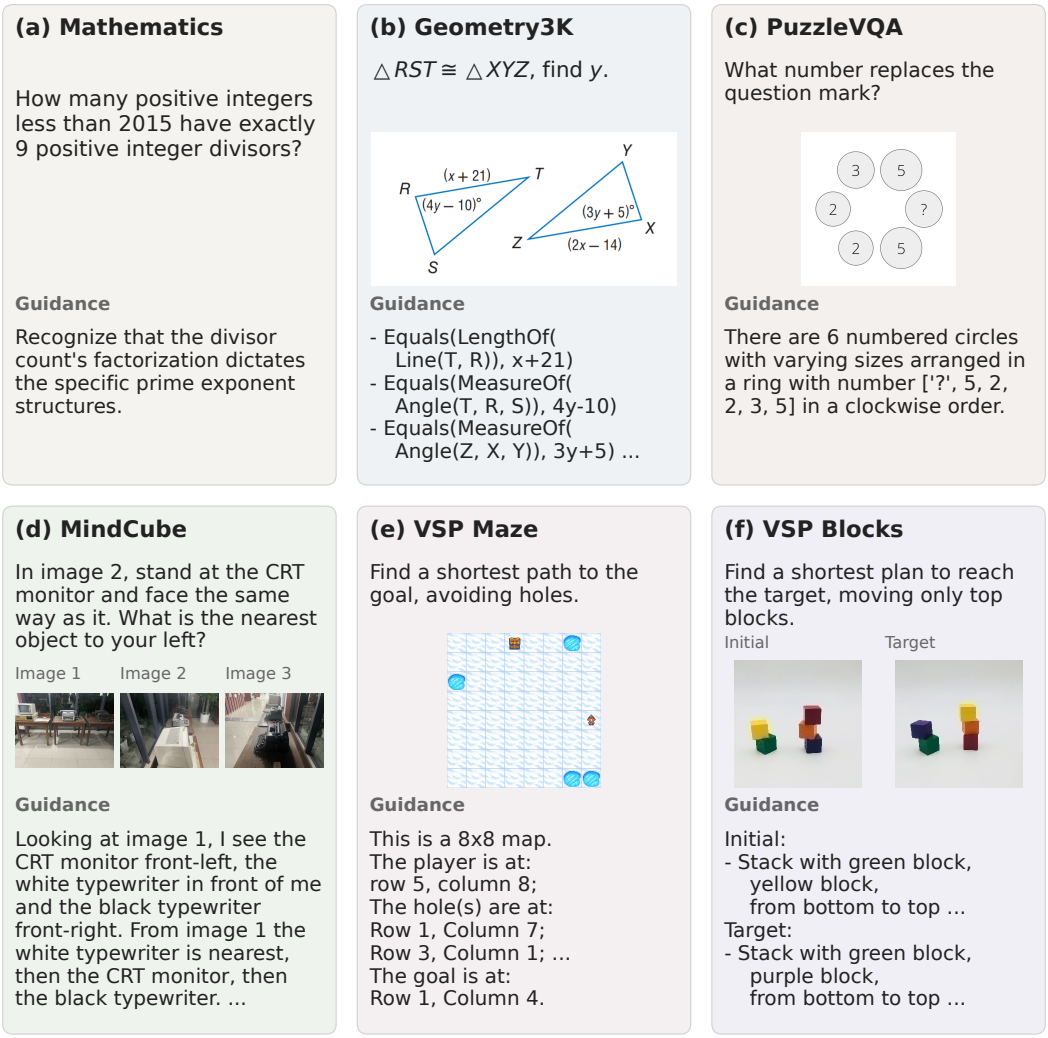


Figure 2: Examples of guidance used during training. Questions are shortened, and "..." marks omitted text.

2 Related Work

Learning with privileged information. Learning with privileged information uses additional information available only during training (Vapnik & Vashist, 2009). Generalized distillation extends

this idea by transferring knowledge from a teacher with access to privileged information to a student that receives only the original input (Lopez-Paz et al., 2016). In reinforcement learning, asymmetric actor–critic methods provide privileged state information to the critic but not the actor (Pinto et al., 2018). LEAP trains partially observed agents using feedback from an expert with access to privileged information (Choudhury & Sodhi, 2025). Like these approaches, BlendPO uses guidance available only during training and relies only on the original input at inference time.

Guidance for reinforcement learning. Guided-RL methods differ in how they control guidance strength. BREAD, POPE, QuestA, AdaBack, and PrefixRL condition rollouts on solution prefixes (Zhang et al., 2025b; Qu et al., 2026; Li et al., 2025; Amani et al., 2025; Setlur et al., 2026), which allows fine-grained control through the prefix length. BREAD and AdaBack adapt the prefix length, and POPE selects the shortest prefix with which the base model produces a successful rollout (Zhang et al., 2025b; Amani et al., 2025; Qu et al., 2026). Hint-based methods control strength through coarser choices. Scaf-GRPO chooses among three levels of hints with increasing detail (Zhang et al., 2025a). Guide-GRPO and HINT provide hints only when all unguided responses to a problem fail (Nath et al., 2025; Wang et al., 2026b). Other methods focus on how guidance is obtained. LUFFY incorporates off-policy expert traces (Yan et al., 2025), SAGE generates hints online (Liao et al., 2026), and HiLL trains a hint generator with a reward that favors hints whose success transfers to the policy without hints (Xia et al., 2026). Guide-GRPO and OC-GRPO update the policy under the original context and correct the mismatch between guided sampling and this update through token-level importance weighting (Nath et al., 2025; Agrawal et al., 2026). HINT instead reweights each group's update by an affinity score that reflects how compatible guided rollouts are with the policy's own distribution (Wang et al., 2026b). BlendPO also updates the policy under the original context, but enables fine-grained control of guidance strength through a logit blending ratio while keeping the guidance text unchanged, and omits this importance correction.

Privileged distillation for reasoning. On-policy distillation trains on the student's own response prefixes (Agarwal et al., 2024). Privileged variants supply the teacher view with reference solutions (OPSD), privileged information (π-Distill), or solution-derived skill cards (SKALD) (Zhao et al., 2026; Penaloza et al., 2026; Jiang et al., 2026). PAINT and β-OPSD interpolate privileged distillation targets (Tan & Hong, 2026; Xu et al., 2026). These methods use teacher token distributions as learning targets. BlendPO uses the guided distribution to generate rollouts and learns from verifier rewards, without a teacher-distribution matching loss.

Logit fusion for RL rollouts. The technical post *Learning from Mixed Rollouts* (Zhang et al., 2026) also generates RL rollouts from fused logits, but it fuses a frozen teacher's logits with the student's on the same input, anneals the mixing weight during training, and applies fused-behavior importance ratios. BlendPO instead obtains both sets of logits from the current policy, without and with training-only guidance, so the blending weight sets how strongly the guidance shapes generation. It also assigns different strengths within each group and updates under the original context without importance correction.

3 METHOD

RLVR provides limited learning signal on problems that the policy rarely solves, because sampled responses often fail and GRPO assigns zero group-relative advantages (Shao et al., 2024). We focus on *guided RL*, which addresses this issue by adding training-only guidance to the rollout context. Stronger guidance makes successful rollouts more likely but can transfer less to the unguided setting in which the policy is evaluated.

3.1 CONTROLLING GUIDANCE STRENGTH

To balance successful rollout generation and transferability to unguided settings, prior methods control the strength of guidance during generation. Methods using solution-prefix guidance provide a partial reference solution during rollout generation and control guidance strength by adjusting the prefix length (Zhang et al., 2025b; Amani et al., 2025; Qu et al., 2026).

Adjusting the length does not apply to guidance without a natural order, such as abstract hints and visual states. *Abstract hints*, such as HINT's Meta-Hints, suggest concepts, principles, or strategies for solving a problem without giving the concrete derivation (Wang et al., 2026b). Such guidance does not decompose into progressively revealed steps, so shortening it can remove essential information or leave an incomplete description, changing its content rather than its strength. For hints, existing methods therefore rely on coarse choices. Scaf-GRPO chooses among three levels of hints with increasing detail (Zhang et al., 2025a), which requires writing several hints for each problem. HINT provides a hint only when all unguided responses in a group fail (Wang et al., 2026b), so each problem is trained either without the hint or with the full hint. When a problem is rarely solved without the hint but success under the full hint transfers poorly, this binary choice offers no intermediate strength. We instead control guidance strength in logit space while keeping the guidance content unchanged.

3.2 Guidance control in logit space with BlendPO

We introduce ***BlendPO*** (Blended-context Policy Optimization), which controls the influence of guidance by blending logits from the original and guided contexts. Let x be the original input, g the training-time guidance, and θ_{old} the policy snapshot used to collect rollouts. At each decoding step, the same policy produces logits z_t^0 conditioned on $(x, y_{<t})$ and z_t^g conditioned on $(x, g, y_{<t})$. For guidance strength $\alpha \in [0, 1]$ and temperature $T > 0$, we sample from

$$\tilde{z}_t^{(\alpha)} = (1-\alpha) z_t^0 + \alpha z_t^g,$$
$$q_{\alpha,t}(v) = \frac{\exp\left(\tilde{z}_{t,v}^{(\alpha)}/T\right)}{\sum_u \exp\left(\tilde{z}_{t,u}^{(\alpha)}/T\right)}. \tag{1}$$

Each sampled token $y_t \sim q_{\alpha,t}$ is appended to both contexts, defining the response distribution $q_\alpha(y \mid x, g)$. When $\alpha = 0$, generation is unguided, while $\alpha = 1$ yields fully guided generation. Intermediate values give intermediate strengths without shortening the guidance, which allows fine-grained control. Appendix C.2 describes additional properties of this distribution.

When sampling $N > 1$ responses for a problem, guidance strengths can be allocated across rollouts in different ways. Each response $y_i \sim q_{\alpha_i}(\cdot \mid x, g)$ uses a strength α_i that remains fixed throughout its rollout, and all responses belong to the same GRPO reward group. For example, ***fixed-strength*** allocation assigns a common strength $\alpha_i = a$ with $a \in (0, 1)$, while a ***within-group sweep*** assigns $\alpha_i = i/N$ for $i = 0, \ldots, N-1$, so that guidance strength increases in equal increments across the group, starting from unguided generation. Appendix C.3 provides the allocation settings used in our experiments. Appendix D.1 evaluates the effectiveness of these allocations and compares a broader range of guidance allocation strategies.

3.3 Learning under the original context

BlendPO uses guidance to generate rollouts, then updates the policy under the original input x. Each response receives the task verifier reward $r_i = R(x, y_i)$, independent of its guidance strength. With group mean $\bar{r} = \left(\sum_{j=0}^{N-1} r_j\right)/N$, the response-level advantage is

$$A_i = \frac{r_i - \bar{r}}{\text{std}(r_0, \ldots, r_{N-1}) + \varepsilon}, \tag{2}$$

where $\varepsilon > 0$ ensures numerical stability. Following GRPO, we apply A_i to every token of y_i (Shao et al., 2024) and use a clipped surrogate with the original-context probability ratio

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}. \tag{3}$$

Because responses are sampled from the blended distribution but updated under the original context, prior work mitigates this mismatch with an importance weight (Nath et al., 2025; Agrawal et al., 2026), which for BlendPO is

$$w_{i,t} = \frac{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}{q_{\alpha_i,t}(y_{i,t} \mid x, g, y_{i,<t})}. \tag{4}$$

However, this weight also downweights tokens made more likely by guidance, potentially reducing the contribution of successful responses. We therefore omit it, as applying it with abstract hints and visual states lowered accuracy in our experiments (Appendix D.2).

3.4 Guidance without a natural order

Unlike shortening a solution prefix, BlendPO controls guidance strength without relying on the order of the guidance content. This is particularly useful when the only available guidance has no natural order. Visual reasoning is such an example where reference solutions are often unavailable but the visual input can still be described. We use *visual states*, textual descriptions of the objects, attributes, and relations in the visual input, which can be retrieved as dataset metadata or collected from the task environment without solving the problem. Examples include formal language annotations in geometry, image captions for visual puzzles, object relations in spatial reasoning, and symbolic states in visual planning (Figure 2b–f). We use them only as training-time guidance (Appendix B).

4 Experiments

We first compare BlendPO with existing methods for controlling guidance strength in mathematics (Section 4.2). We then extend BlendPO to visual reasoning where reference solutions are unavailable (Section 4.3). Finally, we ablate the within-group sweep and compare training and evaluation budgets (Section 4.4).

4.1 Experimental setup

We use full-parameter fine-tuning of Qwen3-1.7B (Qwen Team, 2025b) for mathematics and LoRA fine-tuning of Qwen3-VL-4B-Instruct (Qwen Team, 2025a) for visual tasks. All learned policies are evaluated without guidance. Unless stated otherwise, for trained models we report means and sample SDs over three seeds. Base denotes the frozen initial model. BlendPO uses the within-group sweep with $N = 8$ responses per problem unless stated otherwise (Section 3.2). Dataset construction, evaluation protocols, and baseline adaptations are detailed in Appendix A.

Datasets. For mathematics, we train on 2,038 problems from DAPO-Math-17K (Yu et al., 2025) that the base model rarely solves, failing in all four initial attempts but succeeding at least once in four additional attempts. We use verified GPT-OSS-120B (OpenAI, 2025) solutions from DAPO-MATH-17k-oss-reasoning to construct solution prefixes and to generate abstract hints offline (Appendix B.2). We evaluate on MATH (Hendrycks et al., 2021) and AIME 2024, 2025, and 2026. For AIME, we report Avg@16, combining the 2024 and 2025 problems into one evaluation set. For visual reasoning, we evaluate on Geometry3K (Lu et al., 2021), PuzzleVQA (Chia et al., 2024), MindCube (Wang et al., 2026a), BLINK multi-view, and VSP Maze and VSP Blocks from the Visual Spatial Planning (VSP) benchmark (Wu et al., 2025). The Maze settings use grid sizes of 5×5, 6×6, 7×7, and 8×8. For Blocks, 3, 4, and 5 indicate the minimum number of moves needed to reach the target configuration. For each dataset, training and evaluation questions are disjoint. MindCube's evaluation images nevertheless overlap with its training images, so we also evaluate the MindCube-trained policies on the BLINK multi-view subset without further training (Fu et al., 2024).

Baselines. In mathematics, we compare with methods using solution-prefix guidance (BREAD, POPE, and OC-GRPO) and methods using abstract hints (Scaf-GRPO, HINT, and OC-GRPO), running OC-GRPO with each type of guidance. Among methods using solution-prefix guidance, BREAD follows its adaptive prefix-search procedure (Zhang et al., 2025b), and the other methods use candidate prefixes constructed in 20% increments following Agrawal et al. (2026) (Appendix A.3). For Scaf-GRPO, we replace its original three-tier hint hierarchy with a single abstract hint per problem, identical to the one used by BlendPO. In visual reasoning, we use HINT and OC-GRPO, the two strongest abstract-hint baselines in mathematics, with visual states as guidance. Methods using abstract hints or visual states, including BlendPO, use the same guidance content while following their respective training procedures.

4.2 Comparison with existing strength controls in mathematics

Table 1: Mathematical reasoning: unguided accuracy (%). AIME uses Avg@16. Bold marks the highest mean in each column. *Scaf-GRPO uses the same abstract hints as BlendPO in place of its original three-tier hints.

Method	Training guidance	MATH	AIME 24–25	AIME 26
Base	None	70.44	11.88	8.96
GRPO (Shao et al., 2024)	None	74.83 ± 0.25	17.67 ± 0.42	11.46 ± 1.46
BREAD (Zhang et al., 2025b)	Solution prefix	74.58 ± 0.93	16.81 ± 0.97	12.71 ± 1.30
POPE (Qu et al., 2026)	Solution prefix	76.13 ± 0.13	19.83 ± 0.57	12.57 ± 0.84
OC-GRPO (Agrawal et al., 2026)	Solution prefix	76.96 ± 0.48	19.41 ± 0.89	13.61 ± 0.48
Scaf-GRPO* (Zhang et al., 2025a)	Abstract hint	74.53 ± 0.36	16.81 ± 0.47	12.57 ± 0.43
HINT (Wang et al., 2026b)	Abstract hint	76.85 ± 1.02	18.96 ± 1.95	12.50 ± 0.95
OC-GRPO (Agrawal et al., 2026)	Abstract hint	77.85 ± 0.88	20.63 ± 0.99	13.89 ± 1.32
BlendPO	Abstract hint	$\mathbf{79.40 \pm 0.39}$	$\mathbf{21.15 \pm 0.45}$	$\mathbf{16.18 \pm 0.32}$

Using abstract hints, BlendPO achieves unguided accuracy 79.40% on MATH, compared with 74.83% for GRPO, and consistently outperforms GRPO on both AIME evaluations (Table 1). It also outperforms baselines using solution-prefix guidance, as well as methods using the same abstract hints with coarse strength controls. The gain comes from controlling guidance strength rather than from the hints alone. Training with the same hints at full strength lowers MATH accuracy to 65.55%, below the base model, whereas BlendPO with a fixed intermediate strength ($\alpha = 0.25$) already reaches 78.84% (Appendix D.1).

4.3 Extending BlendPO to visual reasoning where reference solutions are unavailable

We next apply BlendPO with visual-state guidance, which is drawn from dataset metadata or the task environment without constructing a solution trajectory.

Table 2: Accuracy without any guidance (%) of methods trained with visual-state guidance across visual tasks. Bold marks the highest mean in each row.

Evaluation	Frozen	Trained policies (unguided)			
Dataset	Base	GRPO	HINT	OC-GRPO	BlendPO
Geometry3K	51.25	56.46 ± 0.58	58.02 ± 2.49	59.01 ± 1.50	$\mathbf{63.56 \pm 0.83}$
PuzzleVQA	39.75	52.08 ± 4.54	54.42 ± 1.77	53.33 ± 1.61	$\mathbf{65.25 \pm 1.30}$
MindCube	32.35	44.84 ± 2.47	45.18 ± 2.50	47.41 ± 2.67	$\mathbf{55.11 \pm 3.80}$
BLINK multi-view	29.32	41.60 ± 1.89	36.59 ± 3.86	45.61 ± 6.12	$\mathbf{51.88 \pm 1.99}$
Maze 5×5	12.00	22.67 ± 1.53	21.67 ± 2.08	28.67 ± 7.37	$\mathbf{69.67 \pm 4.73}$
Maze 6×6	10.00	6.00 ± 3.46	8.33 ± 3.06	13.67 ± 6.66	$\mathbf{53.33 \pm 4.04}$
Maze 7×7	3.00	8.00 ± 6.08	4.67 ± 1.15	4.00 ± 3.46	$\mathbf{67.00 \pm 6.24}$
Maze 8×8	3.00	19.00 ± 4.58	4.33 ± 0.58	6.00 ± 5.00	$\mathbf{75.33 \pm 5.13}$
Blocks, 3 moves	0.00	24.33 ± 11.59	4.00 ± 0.00	11.33 ± 10.21	$\mathbf{31.67 \pm 4.73}$
Blocks, 4 moves	0.00	2.00 ± 2.65	2.00 ± 2.00	4.33 ± 4.04	$\mathbf{15.33 \pm 3.06}$
Blocks, 5 moves	0.00	0.00 ± 0.00	0.67 ± 1.15	0.67 ± 1.15	$\mathbf{1.67 \pm 1.53}$

Across all four visual task families, BlendPO outperforms GRPO and the baselines trained with the same visual states in unguided accuracy (Table 2). These results demonstrate the potential of BlendPO to broaden the practical applicability of guided RL when reference solutions are unavailable. As in mathematics, the gain comes from controlling guidance strength rather than from the visual states alone. Visual states raise the base model's accuracy when provided at evaluation, for example from 51.25% to 64.89% on Geometry3K (Appendix B.3), but full-strength visual-state guidance during training falls below GRPO on Geometry3K and PuzzleVQA (Appendix D.1). We further compare BlendPO with prefix-based methods that use solutions generated from the guidance, which requires an additional generation step. On Geometry3K, we generate solutions from the

visual states and answers, and the 1,078 of the 2,101 training problems that yield a verified solution form the training pool for all Geometry3K runs (Appendix A). On this pool, BlendPO outperforms POPE and OC-GRPO using prefixes of the generated solutions (63.56% vs. 57.74% and 57.29%; Appendix A.4). This suggests that using guidance directly with fine-grained strength control is more effective than converting it into solutions for prefix-based methods.

4.4 Ablations and further analyses

Ablating the within-group sweep. BlendPO allocates guidance strength using a within-group sweep, which involves two components. Logit blending generates responses at intermediate guidance strengths, while within-group strength variation assigns a different strength to each of the eight responses for a problem. We ablate each component on Maze8 (Figure 3). Among our tasks, Maze8 provides visual-state guidance that most faithfully reflects the original input. Its visual state exactly transcribes the maze configuration, whereas some Geometry3K and MindCube visual states include information unavailable from the original input, and some PuzzleVQA visual states omit information needed to solve the problem (Appendix B.1). Maze8 therefore allows us to compare allocation strategies without confounding effects from imperfect guidance. To reduce sensitivity to the original 100-question evaluation set, we evaluate on 512 newly generated 8×8 mazes, disjoint from both training and the original evaluation set.

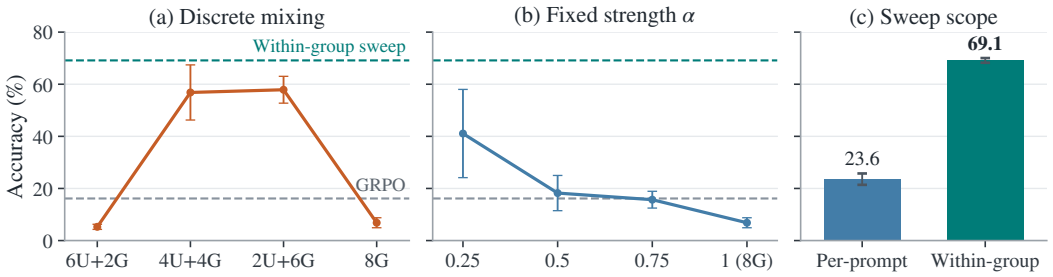


Figure 3: Within-group sweep ablation on 512 generated 8×8 mazes. (a) Discrete mixtures of unguided (U) and fully guided (G) rollouts in a group of eight. (b) One fixed strength for all rollouts, where $\alpha = 1$ is fully guided training (8G). (c) The sweep strengths assigned per prompt or within each group. Dashed lines mark GRPO and the within-group sweep.

Removing logit blending leaves discrete guidance within a group, with two, four, or six of the eight rollouts fully guided and the rest unguided (6U+2G, 4U+4G, and 2U+6G). Removing within-group strength variation keeps one strength per group, either a fixed $\alpha \in \{0.25, 0.5, 0.75\}$ or a per-prompt sweep. The per-prompt sweep gives all eight responses of a problem one strength from $\{0, 1/8, \ldots, 7/8\}$. Each strength goes to four randomly chosen problems in every 32-problem batch, so the batch contains the same strengths as the within-group sweep. Fully guided training (8G) removes both components (Appendix C.3).

The within-group sweep outperforms all ablated variants. Without logit blending, 4U+4G and 2U+6G outperform GRPO but remain below the within-group sweep, while 6U+2G falls below GRPO. Without within-group strength variation, only the lowest fixed strength outperforms GRPO, and the per-prompt sweep provides little improvement over GRPO despite using the same set of strengths as the within-group sweep. Fully guided training, which removes both components, also falls below GRPO. These results show that both components contribute on Maze8.

This conclusion is limited to Maze8, where guidance provides a much larger improvement over unguided generation than on the other tasks (Appendix B.3). On the other tasks in Appendix D.1, some single-strength allocations come close to the within-group sweep. We hypothesize that varying guidance strength within a group is most beneficial when guidance has a large effect on whether a response succeeds.

Training and evaluation budgets. BlendPO runs each response under both the original and guided contexts, requiring more computation than GRPO at the same training step. To test whether its advantage holds under matched compute, we compare the methods using GRPO's final mean

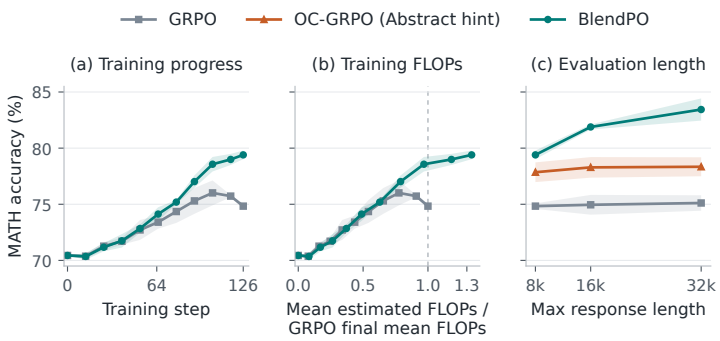


Figure 4: MATH accuracy versus (a) training steps, (b) estimated cumulative FLOPs, and (c) maximum response length. Bands show ± 1 sample SD. Step 0 is the shared base model. Panel (b) normalizes mean costs by GRPO's final mean FLOPs (dashed line). Panel (c) evaluates the same final checkpoints with maximum response lengths of 8k, 16k, and 32k, including abstract-hint OC-GRPO. Details appear in Appendices C.6 and D.3.

estimated training FLOPs as the budget (Figure 4a,b; Appendix C.6). We select the last BlendPO checkpoint whose mean FLOPs stay within GRPO's final mean (step 104). At this checkpoint, BlendPO scores 78.57% compared with 74.83% for final GRPO, a 3.74-point gain using 96.87% of GRPO's mean estimated FLOPs. At the common final training step, BlendPO uses 1.34 times the mean estimated FLOPs of GRPO.

BlendPO also benefits more from a longer maximum response length at evaluation (Figure 4c). Increasing the maximum length from 8k to 32k raises its mean MATH accuracy from 79.40% to 83.44%, while OC-GRPO, the strongest baseline, shows little additional improvement beyond 16k. Because OC-GRPO uses the same abstract hints, this difference suggests that the gain is not attributable to the hints alone. Appendix D.3 provides additional details on evaluation budgets.

5 Conclusion

We propose BlendPO, which enables fine-grained control of guidance strength without changing guidance content. It is well suited to hints, visual states, and other guidance without a natural order. BlendPO blends the logits of the guided and original contexts, sweeps guidance strength across rollouts within each group, and learns from the resulting responses under the original context. With abstract hints in mathematics, BlendPO outperforms methods using solution-prefix guidance and methods using the same hints, and its gain persists within GRPO's estimated training-compute budget. Extended to visual reasoning with visual states, which are available without solving the problem, BlendPO consistently improves unguided performance over GRPO and methods using the same visual states across four task families, demonstrating its potential to broaden the practical applicability of guided RL when reference solutions are unavailable. Future work includes testing BlendPO in additional domains and exploring guidance allocation strategies beyond a simple sweep.

Limitations. We analyze the within-group sweep in depth only on Maze8, and its allocation may not be optimal for every task. Some visual states come from existing dataset annotations, which can contain information unavailable from the input or omit information needed to solve the problem (Appendix B.1), which may limit the gains from visual-state guidance. Our experiments are limited to Qwen3 and Qwen3-VL models with at most 4B parameters. Finally, our compute comparison uses estimated FLOPs, which may not reflect wall-clock training cost.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, assist in writing proofs, propose or refine hypotheses, design or provide feedback on research methodology and experiments, implement methods, assist with translation, and interpret results. We have not used generative AI tools to generate synthetic data sets, develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, clean or reformat datasets, or support qualitative or thematic data analysis. Additionally, we used generative AI tools for brainstorming, discovering research topics and identifying gaps, searching for information, identifying, summarizing, and analyzing relevant literature, formatting references, creating or modifying figures, writing or editing software code such as evaluation and plotting scripts, drafting parts of the paper, and editing the paper to improve readability. We have reviewed all AI-assisted work: baseline implementations were compared against their official code, mathematical claims, derivations, and literature were verified directly by the authors, and all reported numbers were checked against the experiment logs. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

The authors have read and adhere to the ICLR Code of Ethics. All experiments use publicly available datasets and benchmarks, namely DAPO-Math-17K, MATH, AIME 2024–2026, Geometry3K, PuzzleVQA, MindCube, BLINK, VSP, and Reasoning Gym, and open-weight models, Qwen3-1.7B and Qwen3-VL-4B-Instruct. The study involves no human subjects, personally identifiable data, or private information, and collects no new data from people.

REPRODUCIBILITY STATEMENT

Appendix A describes the datasets, evaluation protocols, training settings (Tables 3–5), and baseline implementations. Appendix B describes how the abstract hints and visual states are constructed, and Appendix C gives the BlendPO sampling procedure, strength schedules, and optimization details. All trained results report means and sample SDs over three training seeds.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Priyank Agrawal, Ankur Samanta, Shervin Ghasemlou, Boris Vidolov, Jalaj Bhandari, Kavosh Asadi, Daniel Jiang, and Aditya Modi. Off-Context GRPO: Learning to Reason on Hard Problems using Privileged Information. *arXiv preprint arXiv:2607.19313*, 2026. URL https://arxiv.org/abs/2607.19313.

Mohammad Hossein Amani, Aryo Lotfi, Nicolas Mario Baldwin, Samy Bengio, Mehrdad Farajtabar, Emmanuel Abbe, and Robert West. RL for Reasoning by Adaptively Revealing Rationales. *arXiv preprint arXiv:2506.18110*, 2025. URL https://arxiv.org/abs/2506.18110.

Yew Ken Chia, Vernon Toh Yan Han, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. PuzzleVQA: Diagnosing Multimodal Reasoning Challenges of Language Models with Abstract Visual Patterns. *arXiv preprint arXiv:2403.13315*, 2024. URL https://arxiv.org/abs/2403.13315.

Sanjiban Choudhury and Paloma Sodhi. Better than Your Teacher: LLM Agents that learn from Privileged AI Feedback. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.05434.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal Large Language Models Can See but Not Perceive. In *European Conference on Computer Vision*, 2024. URL https://arxiv.org/abs/2404.12390.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving With the MATH Dataset. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

hiyouga. MathRuler. https://github.com/hiyouga/MathRuler, 2025.

Yubo Jiang, Fengying Xie, Zhiguo Jiang, and Haopeng Zhang. Distill Skills into Weights, Not Prompts: Abstract Skills as Privileged Signals for On-Policy Self-Distillation. *arXiv preprint arXiv:2608.09826*, 2026. URL https://arxiv.org/abs/2608.09826.

Jiazheng Li, Hongzhou Lin, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Yi Wu, and Jingzhao Zhang. QuestA: Expanding Reasoning Capacity in LLMs via Question Augmentation. *arXiv preprint arXiv:2507.13266*, 2025. URL https://arxiv.org/abs/2507.13266.

Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, and Jiang Bian. Self-Hinting Language Models Enhance Reinforcement Learning. *arXiv preprint arXiv:2602.03143*, 2026. URL https://arxiv.org/abs/2602.03143.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.03643.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable Geometry Problem Solving with Formal Language and Symbolic Reasoning. In *Proceedings of ACL-IJCNLP*, 2021. URL https://arxiv.org/abs/2105.04165.

Vaskar Nath, Elaine Lau, Anisha Gunjal, Manasi Sharma, Nikhil Baharte, and Sean Hendryx. Adaptive Guidance Accelerates Reinforcement Learning of Reasoning Models. *arXiv preprint arXiv:2506.13923*, 2025. URL https://arxiv.org/abs/2506.13923.

OpenAI. gpt-oss-120b & gpt-oss-20b Model Card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged Information Distillation for Language Models. *arXiv preprint arXiv:2602.04942*, 2026. URL https://arxiv.org/abs/2602.04942.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric Actor Critic for Image-Based Robot Learning. In *Robotics: Science and Systems*, 2018. URL https://arxiv.org/abs/1710.06542.

Yuxiao Qu, Amrith Setlur, Virginia Smith, Ruslan Salakhutdinov, and Aviral Kumar. POPE: Learning to Reason on Hard Problems via Privileged On-Policy Exploration. *arXiv preprint arXiv:2601.18779*, 2026. URL https://arxiv.org/abs/2601.18779.

Qwen Team. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Qwen Team. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025b. URL https://arxiv.org/abs/2505.09388.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Amrith Setlur, Zijian Wang, Andrew Cohen, Paria Rashidinejad, and Sang Michael Xie. Reuse your FLOPs: Scaling RL on Hard Problems by Conditioning on Very Off-Policy Prefixes. *arXiv preprint arXiv:2601.18795*, 2026. URL https://arxiv.org/abs/2601.18795.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Zafir Stojanovski, Oliver Stanley, Joe Sharratt, Richard Jones, Abdulhakeem Adefioye, Jean Kaddour, and Andreas Köpf. REASONING GYM: Reasoning Environments for Reinforcement Learning with Verifiable Rewards. *arXiv preprint arXiv:2505.24760*, 2025. URL https://arxiv.org/abs/2505.24760.

Zhiquan Tan and Yinrong Hong. PAINT: Partial-Solution Adaptive Interpolated Training for Self-Distilled Reasoners. *arXiv preprint arXiv:2604.26573*, 2026. URL https://arxiv.org/abs/2604.26573.

Vladimir Vapnik and Akshay Vashist. A New Learning Paradigm: Learning Using Privileged Information. *Neural Networks*, 22(5–6):544–557, 2009. doi: 10.1016/j.neunet.2009.06.042. URL https://doi.org/10.1016/j.neunet.2009.06.042.

Qineng Wang, Baiqiao Yin, Pingyue Zhang, Jianshu Zhang, Kangrui Wang, Zihan Wang, Jieyu Zhang, Keshigeyan Chandrasegaran, Han Liu, Ranjay Krishna, Saining Xie, Jiajun Wu, Li Fei-Fei, and Manling Li. MindCube: Spatial Mental Modeling from Limited Views. In *International Conference on Learning Representations*, 2026a. URL https://arxiv.org/abs/2506.21458.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Ma Shuguang, Fei Yu, and Yanghua Xiao. Don't Tell the Answer, Truly Guide the Reasoning During RL Rollouts. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3437–3455. Association for Computational Linguistics, 2026b. doi: 10.18653/v1/2026.findings-acl.169. URL https://aclanthology.org/2026.findings-acl.169/.

Qiucheng Wu, Handong Zhao, Michael Saxon, Trung Bui, William Yang Wang, Yang Zhang, and Shiyu Chang. VSP: Diagnosing the Dual Challenges of Perception and Reasoning in Spatial Planning Tasks for MLLMs. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025. URL https://arxiv.org/abs/2407.01863.

Yu Xia, Canwen Xu, Zhewei Yao, Julian McAuley, and Yuxiong He. Learning to Hint for Reinforcement Learning. *arXiv preprint arXiv:2604.00698*, 2026. URL https://arxiv.org/abs/2604.00698.

Jiawei Xu, Minghui Liu, Juzheng Zhang, Tom Goldstein, and Furong Huang. β-OPSD: Deriving with Policy Optimization, Training with Self-Distillation. *arXiv preprint arXiv:2607.28582*, 2026. URL https://arxiv.org/abs/2607.28582.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to Reason under Off-Policy Guidance. *arXiv preprint arXiv:2504.14945*, 2025. URL https://arxiv.org/abs/2504.14945.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering Complex Control in MOBA Games with Deep Reinforcement Learning. *arXiv preprint arXiv:1912.09729*, 2019. URL https://arxiv.org/abs/1912.09729.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Juzheng Zhang, Abhimanyu Hans, John Kirchenbauer, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. Learning from Mixed Rollouts: Logit Fusion as a Bridge Between Imitation and Exploration. *Notion Blog*, February 2026. URL https://juzhengz.notion.site/logit-fusion. Public technical post, February 25.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-GRPO: Scaffolded Group Relative Policy Optimization for Enhancing LLM Reasoning. *arXiv preprint arXiv:2510.19807*, 2025a. URL https://arxiv.org/abs/2510.19807.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. BREAD: Branched Rollouts from Expert Anchors Bridge SFT & RL for Reasoning. *arXiv preprint arXiv:2506.17211*, 2025b. URL https://arxiv.org/abs/2506.17211.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-Distilled Reasoner: On-Policy Self-Distillation for Large Language Models. *arXiv preprint arXiv:2601.18734*, 2026. URL https://arxiv.org/abs/2601.18734.

A EXPERIMENTAL SETUP

A.1 DATASETS AND EVALUATION

Table 3 lists the training steps and evaluation settings.

Table 3: Training steps and evaluation settings. Batch sizes appear in Table 4, and dataset sizes are described below. Token limits are per response.

Evaluation task	Training steps	Test size	Token limit
MATH (sparse-success DAPO subset)	126	5,000	8,192
AIME 2024–25 / 2026	126	60 / 30	8,192
Geometry3K	134	601	4,096
PuzzleVQA	40	400	2,048
VSP Maze, sizes 5–8	40	100 each	2,048
VSP Blocks, 3–5 moves	40	100 each	2,048
MindCube / BLINK multi-view	100	2,000 / 133	2,048

Mathematical reasoning. The mathematical training set is a sparse-success subset of DAPO-Math-17K (Yu et al., 2025). Starting from 12,775 problems with verified GPT-OSS-120B reference solutions from DAPO-MATH-17k-oss-reasoning, we retain only problems on which the Qwen3-1.7B base model fails in all four initial attempts but succeeds at least once in four additional attempts. MATH evaluation uses all 5,000 test problems (Hendrycks et al., 2021) with one sampled response per problem and thinking mode disabled. The scorer extracts and normalizes the final boxed answer.

AIME combines the 2024 and 2025 competitions into 60 problems and evaluates the 30 AIME 2026 problems separately. Each problem receives 16 samples of up to 8,192 tokens with thinking mode disabled. Avg@16 is the average of per-problem sample accuracies. The raw totals are 960 and 480 responses. The comparisons use step 126. A lexical audit finds no AIME 2024–25 or 2026 problem in the sparse-success subset. After lowercasing and stripping LaTeX commands, no evaluation problem has word-set Jaccard similarity above 0.5 with any training prompt. One AIME 2026 problem text does occur in the wider 12,775-problem DAPO pool, which is used only for subset selection. It was not selected into the sparse-success subset, so no trained policy saw it.

Geometry and visual puzzles. Geometry3K uses the released 601-problem test split (Lu et al., 2021) and MathRuler answer equivalence (hiyouga, 2025). Training draws from a 1,078-problem pool selected by the availability of a verified generated solution. For each of the 2,101 training problems, a Qwen3-VL-4B-Instruct checkpoint from a preliminary Geometry3K GRPO run, trained with the annotation in context, generated up to five candidate solutions given the image, the diagram's formal language annotation, and the answer. The first MathRuler-correct candidate was kept, covering 1,667 problems. Candidates that the same checkpoint judged to reproduce the private annotation or answer, or that matched explicit leak patterns, were removed, leaving 1,078 problems. The POPE and OC-GRPO prefix baselines use these solutions (Appendix A.4), whereas the main GRPO and BlendPO runs never expose them. Training runs for 134 steps with batch size 8. The input baselines use the same test split and 4,096-token limit. PuzzleVQA uses a category-balanced 400-problem evaluation with 20 questions from each of 20 categories (Chia et al., 2024) and 1,280 training questions. Its guidance supplies only the symbolic caption and excludes the explanation and deduction fields. Evaluation is greedy and scores the final multiple-choice answer.

Visually grounded planning. The maze experiments use the released 100-problem evaluation sets at sizes 5×5 through 8×8, with generated training mazes. Blocks evaluates 100 recombined state pairs at each minimum solution length of three, four, and five moves. The evaluation pairs are excluded from the corresponding training pools. These tasks use the VSP planning interfaces (Wu et al., 2025). Evaluation is greedy and requires reaching the target with a shortest valid action sequence. We report shortest-path accuracy for mazes and shortest-plan accuracy for Blocks.

Spatial reasoning. MindCube policies are compared at step 100 on the same internal 2,000-question MindCube evaluation (Wang et al., 2026a) and on the 133-question BLINK multi-view

subset (Fu et al., 2024). Both evaluations are greedy, use 2,048 tokens, and score the parsed option letter. MindCube reports the unweighted mean accuracy over the around, among, and rotation families, containing 667, 667, and 666 questions. The internal evaluation shares 638 scene IDs with the training pool, motivating the additional BLINK evaluation on different images. The two training pools contain the same 5,617 questions, original-context prompts, images, and answers in the same order. Training uses batch size 8. We also run an additional experiment on leg counting, one of the Reasoning Gym tasks, with Qwen3-1.7B (Appendix D.1).

A.2 Training settings and final checkpoints

Table 4: Core GRPO and BlendPO training settings. Each prompt contributes eight accepted training responses. HINT and Scaf-GRPO generate additional fully guided responses for groups whose unguided responses all fail, and BREAD performs additional prefix search. Learning rates are constant.

Training task	Prompts per step	Reported step	Learning rate
Sparse-success DAPO subset	32	126	10^{-6}
Geometry3K	8	134	2×10^{-5}
PuzzleVQA	32	40	2×10^{-5}
VSP Maze / Blocks	32	40	2×10^{-5}
MindCube	8	100	2×10^{-5}

Table 5 summarizes the fine-tuning and optimizer settings. Mathematical training updates all parameters, while the visual runs use LoRA adapters. Mathematical training responses are capped at 8,192 tokens. Geometry3K training uses 2,048 tokens and evaluation uses the longer limit in Table 3. For BlendPO, group advantages are standardized within each prompt, and its policy objective averages over valid response tokens and uses the dual clipping described in Appendix C.4. Within each task, the allocations and the GRPO control share the same sampling settings, optimizer, clipping, and regularization, and differ only in the intended context and blending rules.

Table 5: Fine-tuning and optimization settings for the main GRPO and BlendPO comparisons. Full FT updates all model parameters. LoRA updates adapters in the language decoder, with the vision tower frozen for visual tasks. Task-specific batch sizes and training steps appear in Table 4. Baseline-specific objective settings are described in Appendices A.3 and A.4.

Setting	Full FT	LoRA
Tasks	Mathematical reasoning	Visual tasks
Model	Qwen3-1.7B	Qwen3-VL-4B-Instruct
LoRA rank r	—	64
LoRA scaling α_{LoRA}	—	128
LoRA dropout	—	0
LoRA target layers	—	Attention and MLP projections
Optimizer	AdamW	AdamW
Learning rate	10^{-6}	2×10^{-5}
Learning-rate schedule	Constant	Constant
Warmup steps	0	0
Adam (β_1, β_2)	$(0.9, 0.999)$	$(0.9, 0.999)$
Weight decay	0.01	0.01
Maximum gradient norm	1.0	1.0
PPO epochs per rollout batch	1	1
PPO clipping (ϵ_-, ϵ_+)	$(0.2, 0.2)$	$(0.2, 0.2)$
Dual-clip parameter c	3	3
Gradient checkpointing	Enabled	Enabled

Batch size counts prompt rows processed per training step, including repeated uses of a question. POPE and OC-GRPO share an augmented training pool in which original and guided rows are counted separately, so equal steps and batch sizes across methods need not imply equal epochs. All methods within a task share the final training step in Table 4. Unless a table identifies an entry as a

representative run, trained results average training seeds 42, 43, and 44, with sample SD using the $n - 1$ denominator.

A.3 Mathematical baseline implementations

The RL baselines follow the data, training steps, batch sizes, seeds, and unguided evaluation protocol described above. Below we specify their guidance and update rules.

Shared prefixes for POPE and OC-GRPO. POPE and OC-GRPO use the same augmented training dataset, including the same original and prefix-guided rows. Prefixes are selected offline using an OC-GRPO-style rule (Agrawal et al., 2026), which tests 20, 40, 60, and 80 percent prefixes of each verified solution with eight base-model rollouts per level to find the shortest prefix yielding a correct response. The selected prefixes are fixed during training. POPE generates each response under its row's context, including the prefix for guided rows, and computes its loss under the same context (Qu et al., 2026). We thus replace POPE's own prefix selection with the shared prefixes above while keeping its update rule.

Scaf-GRPO with matched hints. We adapt Scaf-GRPO from the official implementation (Zhang et al., 2025a). As in Scaf-GRPO, the first 15% of training steps provide no hints. Afterward, when all unguided responses in a group fail, the group receives up to eight fully guided candidates using the identical abstract hint and private-thought prompt (Appendix B.2) as BlendPO. If a candidate succeeds, it replaces one original response. Otherwise, the original group is retained. Guided responses keep their guidance context in policy scoring, while ordinary responses use the original context. The eight accepted responses share group-relative rewards. This retains Scaf-GRPO's guided on-policy update for fully guided responses, unlike BlendPO, which updates all blended responses under the original context.

The adaptation replaces the original three-tier hint hierarchy with a single abstract hint per problem and uses per-update policy refresh and the GRPO control's optimizer and clipping. Fully guided candidates add generation cost beyond the eight accepted responses.

HINT. We adapt the official HINT implementation to the same data, original prompt, and abstract-hint private-thought context as BlendPO (Wang et al., 2026b). Each prompt first receives eight unguided responses. Only a group with eight failures is replaced by eight fully guided responses, and learning uses the original context. Training steps, batch size, and accepted responses are matched, but total generation and verifier calls are not, since each replaced group adds eight responses.

The adaptation retains AAPO's detached group weight with $\lambda = 1$ and the official code's linear ratio band $[0.8, 1.2]$, rather than the paper's symmetric log-ratio band, computed over all eight responses of a group. Rollouts refresh every two optimizer updates, and snapshot log probabilities use the original context. Generation uses an 8,192-token cap.

OC-GRPO. The OC-GRPO runs port the official implementation (Agrawal et al., 2026). Guided rows are sampled under the guided prompt (x, g) and updated under the original prompt x. The training batch pairs every response with x, so the trainable forward computes $\log \pi_\theta(y_t \mid x, y_{<t})$ and the PPO ratio $\rho_{i,t}$ of Equation 3. A second, gradient-free forward of the rollout snapshot under (x, g) supplies only the denominator of a detached per-token weight,

$$w_{i,t} = \mathrm{sg}\left[\mathrm{clip}\left(\frac{\pi_{\theta_{\mathrm{old}}}(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(y_{i,t} \mid x, g, y_{i,<t})},\, 0.01,\, 1\right)\right], \tag{5}$$

which multiplies the clipped token surrogate. Original rows use $w_{i,t} = 1$. The gradient of the negative policy surrogate is proportional to $-\sum_t w_{i,t} A_i \rho_{i,t} \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})$ inside the clipping region, and no gradient flows through $\pi_\theta(y \mid x, g)$. This matches the official prompt-masked mode, which restores the original prompt before the update and uses the guided-prompt forward only for the importance ratio. The weights average 0.98–0.99 over guided tokens, and 0.1–0.2% of guided tokens reach the 0.01 floor. These statistics describe the weights without isolating the effect of correction on learning. The augmented training data and selected prefixes are shared with POPE, as described above. The abstract-hint variant uses the same hint pool as BlendPO. Following OC-GRPO, its augmented training set adds a hint-guided copy of a problem only when

the base model solves it under the guided prompt, yielding 1,381 guided rows alongside the 2,038 original rows. Both guidance variants report seeds 42–44 at step 126. Training uses the GRPO control's learning rate and the official loss reduction. Both prefix baselines process 4,032 original or guided rows by the common final step.

BREAD. Our BREAD runs follow the paper's update equations and search procedure (Zhang et al., 2025b), using the pool's verified GPT-OSS-120B solutions as expert anchors. When a question's unguided rollouts fail, prefixes of the anchor are searched in paragraph units (at most ten) for the shortest one from which the policy succeeds. Rollouts are generated and scored under the question plus that prefix, with the loss on the continuation only. Prefix search adds generation beyond the eight accepted responses. The released code differs from the paper's equations in several places.

A.4 Visual baseline implementations

HINT with visual states. The Qwen3-VL-4B-Instruct HINT comparisons use BlendPO's training pools, original prompts, visual states, answers, and evaluation protocols (Appendix A). BLINK evaluates the MindCube-trained policies without additional training.

These runs provide guidance only when all unguided responses in a group fail, and use AAPO weighting and two-update rollout-policy refresh described in Appendix A.3, substituting visual states for mathematical hints. They share the corresponding BlendPO runs' learning rate, LoRA configuration, frozen vision tower, and 2,048-token rollouts. Fully guided regeneration adds cost beyond the eight accepted responses, so these comparisons are between complete recipes.

OC-GRPO with visual states. The state-guidance adaptation uses the same formal language annotations, symbolic captions, first-person scene descriptions, or planning states as BlendPO on the corresponding tasks. It constructs an augmented pool of original and guided rows and retains OC-GRPO's original-context updates with detached per-token weighting (Equation 5). These compare complete training recipes.

Solution-prefix baselines on Geometry3K. The Geometry3K pool carries a verified generated solution for each of its 1,078 problems (Appendix A), so prefix-based methods apply to this visual task. POPE and OC-GRPO share the augmented training data and offline-selected prefixes, following the prefix-selection procedure in Appendix A.3. Both use batch size 8 and the same final step (134) as GRPO and BlendPO, processing 1,072 rows each. Table 6 reports MathRuler accuracy at this final step.

Table 6: Geometry3K accuracy without any guidance (%): solution-prefix baselines and methods trained with visual states. Prefixes come from generated solutions (Appendix A).

Dataset	OC-GRPO (Prefix)	POPE (Prefix)	HINT (Visual state)	OC-GRPO (Visual state)	BlendPO (Visual state)
Geometry3K	57.29 ± 0.38	57.74 ± 1.75	58.02 ± 2.49	59.01 ± 1.50	$\mathbf{63.56 \pm 0.83}$

B Guidance construction

Guidance content and the resources used to prepare it are separate design choices. In Table 7, reference solutions concern construction of the guidance itself, not all resources used for dataset selection or reward verification.

B.1 Visual states

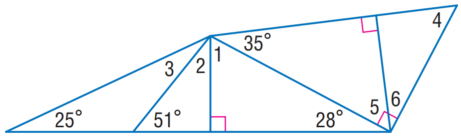
Geometry3K guidance uses formal language annotations. PuzzleVQA inserts a caption alongside the image, question, and answer choices. Maze guidance supplies exact symbolic state, including current and goal locations and hazardous cells. Blocks describes the initial and target stacks, ordered from bottom to top. These descriptions specify the problem without supplying its proof or action sequence.

Table 7: Guidance content and whether construction uses a reference solution.

Task	Guidance content	Reference solution
Geometry3K	Formal language annotation	No
PuzzleVQA	Symbolic visual caption	No
VSP Maze	Maze configuration	No
VSP Blocks	Initial and target configurations	No
MindCube	First-person scene relations	No
Sparse-success DAPO subset	Abstract mathematical direction	Yes

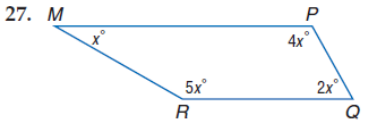
MindCube uses first-person scene descriptions of object and viewpoint relations. It does not append a question-specific derivation to the observations. This condition is evaluated through cross-benchmark transfer in Section 4.3.

Some visual states do not match the input (Figure 5). Geometry3K annotations can name points that do not appear in the diagram. In panel (a), the diagram marks angles only with the numbers 1–6 and degree values, whereas the annotation names twelve points (A–L) and lists 43 lines. In panel (b), the diagram labels the points M, P, Q, and R, whereas the annotation omits M and introduces a point X that does not appear in the diagram. The angle marked x° at M, which the solution requires, appears only as `Angle(R,X,P)`. PuzzleVQA captions can omit information needed to solve the problem. In panel (c), the caption lists the numbers in clockwise order and states only that the circles have varying sizes. The answer depends on the sizes. The circle marked ? matches the size of the circle labeled 2, so the answer is 2. MindCube descriptions can include objects that do not appear in the images. In panel (d), the description places a white-red cabinet and a light-colored sofa relative to both views, although neither object is visible in either view.



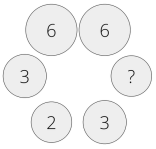
(a) Geometry3K: Find $m\angle 4$.

```
Equals(MeasureOf(Angle(F,E,I)), MeasureOf(angle 4))
Equals(MeasureOf(Angle(G,K,F)), 35)
PointLiesOnLine(I, Line(A,D))
...  (points A–L, 43 lines)
```



(b) Geometry3K: Find $m\angle R$.

```
Equals(MeasureOf(Angle(X,P,Q)), 4x)
Equals(MeasureOf(Angle(R,Q,P)), 2x)
Equals(MeasureOf(Angle(R,X,P)), x)
Equals(MeasureOf(Angle(X,R,Q)), 5x)
```



(c) PuzzleVQA. What is the missing number of the part denoted with a question mark? (Options: 4, 2, 1, 3) Answer: 2.

Caption. There are 6 numbered circles with varying sizes arranged in a ring with number ['?', 3, 2, 3, 6, 6] in a clockwise order.



View 1 View 2

(d) MindCube. In which direction did I move from the first view to the second view? (A) Diagonally forward and left; (B) Directly right; (C) Directly left; (D) Diagonally forward and right. Answer: A.

Description. Looking at view 1, I see the green ball in front of me, the school bag and TV cabinet front-right, the **white-red cabinet** behind me, the **light-colored sofa** front-left and the dark brown sofa in front of me. …

Figure 5: Visual states that do not match the input. (a, b) Geometry3K diagrams with excerpts of their formal language annotations: (a) the annotation names points that are not labeled in the diagram; (b) it replaces the labeled point M with an unlabeled point X. (c) A PuzzleVQA caption that omits the circle sizes that determine the answer. (d) A MindCube question and its first-person scene description; bold marks objects that appear in neither view.

Prompt interface. The guided context inserts the description into the original task prompt while retaining the answer-format instruction. PuzzleVQA introduces a "Symbolic description" before the question and choices. The maze template inserts an "Exact symbolic representation of the test maze" next to the test-image marker. Guidance is conditioning context and is excluded from the response loss.

B.2 Abstract mathematical hints

Mathematical hints are generated offline by Qwen3.8-27B from the problem and a verified reference solution. The generator uses a meta-hint prompt adapted from HINT (Wang et al., 2026b), which asks the generator to identify the central insight and express it in one short sentence. The prompt requests at most 20 words and asks the generator to omit the final answer, intermediate numerical results, and a step-by-step recipe. Among the 2,038 training problems, 9 hints contain a digit, in structural roles such as a 90-degree rotation or a 2-adic valuation rather than as computed values.

The mathematical guided view presents the hint as the model's initial private intuition. A system instruction asks the model to reason naturally without quoting or attributing it. The unguided view omits the intuition. Hint bodies average 18.2 tokens (median 18), and the full added context has a median of 88 tokens (Appendix B.4). The generator sees a verified solution, so this pipeline is not solution-free even though the policy never sees a trace. HINT instead prompts Qwen2.5-72B-Instruct with the question and its final answer to reconstruct the key insight, and restricts training to problems that the same model solves in all four attempts (Wang et al., 2026b). We do not test this construction.

Example (first training problem). The problem distributes 2,015 cards among seven boxes in a repeated back-and-forth traversal and asks which box receives the final card. The abstract hint is:

> The back-and-forth motion creates a periodic cycle whose length is twice the number of boxes minus two.

The example is selected by dataset order, not by rollout outcome. It states a principle and leaves the instance-specific derivation to the learner.

B.3 Base-model input evaluations

Table 8: Frozen Qwen3-VL-4B-Instruct input probes: accuracy (%). Base uses the original input; Base+guidance additionally receives the visual state at evaluation. Questions, decoding, and scoring follow Table 2. MindCube uses mean accuracy over question families.

Dataset	Base	Base+guidance
Geometry3K	51.25	64.89
PuzzleVQA	39.75	66.50
MindCube	32.35	61.41
Maze 5×5	12.00	87.00
Maze 6×6	10.00	63.00
Maze 7×7	3.00	46.00
Maze 8×8	3.00	48.00
Blocks, 3 moves	0.00	2.00
Blocks, 4 moves	0.00	1.00
Blocks, 5 moves	0.00	2.00

Table 8 compares original and guided inputs to the frozen visual base model on the visual tasks. These probes measure immediate use of the supplied visual states, rather than unguided learning after training.

The mathematical probe uses the same training problems, with eight responses per problem. Both conditions use ordinary single-context Qwen3-1.7B generation with matched sampling, an 8,192-token cap, and thinking disabled. The guided input adds both the hint and its private-thought fram-

Table 9: Frozen Qwen3-1.7B input probe on the training problems of the sparse-success subset: eight responses per problem, 8k tokens. Descriptive context comparison; this is not held-out MATH evaluation.

Input context	Accuracy (%)
Original prompt	22.28
Mathematical hint prompt	25.04

ing. This training-pool probe measures neither held-out MATH generalization nor the effect of hint content separately from framing.

B.4 Guidance length across training settings

Table 10: Additional guidance length in tokens. Each training-pool row is counted once. The solution-prefix row measures the guided contexts shared by POPE and OC-GRPO. Body excludes framing; ΔT includes all added chat instructions. Identical image tokens cancel. P90 is the 90th percentile.

Training condition	Pool rows	Guidance body	Net added context ΔT			
		Mean	Mean	Median	P90	Max
Geometry3K	1,078	108.0	119.0	81	174	1,980
PuzzleVQA	1,280	56.5	74.0	70	96.4	103
MindCube	5,617	392.0	433.0	521	577	913
Maze 5×5	1,280	78.1	87.1	83	107	139
Maze 6×6	1,280	95.1	104.1	107	131	179
Maze 7×7	1,280	116.7	125.7	123	155	195
Maze 8×8	1,280	140.1	149.1	147	179	235
Blocks, 3 moves	1,280	60.4	93.4	93	98	100
Blocks, 4 moves	1,280	58.2	91.2	90	97	100
Blocks, 5 moves	1,280	55.3	88.3	87	94	97
Mathematics, abstract hint	2,038	18.2	88.4	88	93	103
Mathematics, solution prefix	2,038	417.6	518.6	394.5	950	3,976

Table 10 tokenizes each training pool with the training model's chat template, with thinking disabled for Qwen3 text models. $\Delta T = T(x, g) - T(x)$ also counts added instructions and headers. Guidance length varies widely across tasks, from a median of 88 added tokens for mathematical hints to 521 for MindCube scene descriptions.

C BlendPO implementation details

C.1 Token-level interpolation

At a shared response prefix, let p_t^0 and p_t^g be the temperature-scaled next-token distributions under the two contexts. Substituting the softmax definition into Equation 1 gives

$$q_{\alpha,t}(v) = \frac{p_t^0(v)^{1-\alpha} p_t^g(v)^{\alpha}}{\sum_{u \in \mathcal{V}} p_t^0(u)^{1-\alpha} p_t^g(u)^{\alpha}}, \tag{6}$$

where $\mathcal{V}$ is the vocabulary. In particular, for tokens u and v with positive probability,

$$\log \frac{q_{\alpha,t}(v)}{q_{\alpha,t}(u)} = (1-\alpha) \log \frac{p_t^0(v)}{p_t^0(u)} + \alpha \log \frac{p_t^g(v)}{p_t^g(u)}. \tag{7}$$

The normalization constants cancel in the ratio. This identity describes how each next-token preference incorporates the original-context policy.

C.2 Conditional divergence at a fixed prefix

Fix a response prefix and let p^0 and p^g be the strictly positive, temperature-scaled softmax distributions from which the sampler draws. Set $\ell(v) = \log[p^g(v)/p^0(v)]$ and $Z(\alpha) = \sum_v p^0(v)e^{\alpha\ell(v)}$. Then $q_\alpha(v) = p^0(v)e^{\alpha\ell(v)}/Z(\alpha)$ and

$$\mathrm{KL}(q_\alpha\|p^0) = \alpha\mathbb{E}_{q_\alpha}[\ell] - \log Z(\alpha), \qquad \frac{d}{d\alpha}\mathrm{KL}(q_\alpha\|p^0) = \alpha\,\mathrm{Var}_{q_\alpha}[\ell] \geq 0. \tag{8}$$

The derivative follows from $\partial_\alpha \log Z = \mathbb{E}_{q_\alpha}[\ell]$ and $\partial_\alpha\mathbb{E}_{q_\alpha}[\ell] = \mathrm{Var}_{q_\alpha}[\ell]$. Thus, weakening guidance reduces conditional divergence from p^0 at this fixed prefix. This holds only at a fixed prefix, since different strengths also change the prefixes visited. It supports interpreting α as a per-token guidance strength.

C.3 Schedules and controls

Table 11: Guidance schedules for a group of eight responses. Mean α averages the assigned blending ratios over a batch.

Schedule	Strengths in a group	Mean α
GRPO	Eight responses at 0	0
Fixed strength	Eight responses at a	a
6U+2G	Six at 0, two at 1	$1/4$
4U+4G	Four at 0, four at 1	$1/2$
2U+6G	Two at 0, six at 1	$3/4$
Fully guided (8G)	Eight at 1	1
Per-prompt sweep	Eight at one $a \in \{0, 1/8, \ldots, 7/8\}$	$7/16$
Within-group sweep	$0, 1/8, \ldots, 7/8$	$7/16$

For fixed-strength allocation, a is held constant for every response. The mathematical, Geometry3K, PuzzleVQA, Maze8, and leg-counting ablations test $a \in \{0.25, 0.50, 0.75\}$. The per-prompt sweep assigns each problem one strength from $\{0, 1/8, \ldots, 7/8\}$ for all eight responses. Every batch assigns each of the eight strengths to the same number of randomly chosen problems, so its mean α matches that of the within-group sweep. In the within-group sweep, each response retains its assigned strength throughout generation. Discrete guidance chooses one context for a complete response. The comparisons hold the guidance, rollout count, batch size, and training steps fixed. The fully guided control uses ordinary single-context generation under the guided prompt and the same original-context update objective as BlendPO.

These comparisons address different aspects of strength allocation. Each fixed strength and the discrete mixture with the same mean ratio (0.25 and 6U+2G, 0.5 and 4U+4G, 0.75 and 2U+6G) compare intermediate interpolation with discrete context selection. All three pairs are tested on Maze8, and 0.5 and 4U+4G in every allocation experiment. The within-group sweep assigns all eight strengths to each problem's reward group, whereas the per-prompt sweep uses the same batch-level strengths but keeps one strength per group.

C.4 Optimization details

Using the ratios and advantages defined in Section 3.3, we form the clipped GRPO token surrogate (Shao et al., 2024; Schulman et al., 2017):

$$s_{i,t}(\theta) = \min\{\rho_{i,t}(\theta)A_i,\ \mathrm{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,A_i\}\,, \tag{9}$$

where $\epsilon > 0$ is the clipping parameter. Our implementation uses the trainer's dual clipping (Ye et al., 2019): for $A_i < 0$, replace $s_{i,t}$ by $\max\{s_{i,t}, cA_i\}$ with $c > 1$, and otherwise leave it unchanged. Denote the resulting value by $\tilde{s}_{i,t}$. For a minibatch $\mathcal{B}$ of sampled responses, the token-mean policy objective is

$$\widehat{J}(\theta) = \frac{\sum_{i\in\mathcal{B}}\sum_t m_{i,t}\tilde{s}_{i,t}(\theta)}{\sum_{i\in\mathcal{B}}\sum_t m_{i,t}} - \beta\widehat{K}_{\mathrm{ref}}(\theta), \tag{10}$$

where $m_{i,t}$ masks valid response tokens. The optional reference-policy KL term $\widehat{K}_{\mathrm{ref}}$ is evaluated under x, and its coefficient and the clipping settings follow the corresponding task's GRPO recipe. BlendPO adds no privileged-teacher distillation loss.

C.5 Sampling, optimization, and resource use

The paired branches share model parameters and maintain separate key–value caches. After sampling one token from the interpolated logits, both branches consume that token, preserving a shared response prefix. The zero-guidance response can use the ordinary single-context decoding path.

For learning, the rollout snapshot's response log-probabilities are recomputed under the original context. They provide the denominator of the policy-change ratio in Equation 3. The importance ratio in Equation 4 is omitted from the default recipe. Its numerator and denominator use the same temperature and response prefix.

Appendix D.2 tests the untruncated token-level ratio in mathematics and on Maze8.

Paired decoding requires additional context evaluations and cache storage. The primary comparisons match training steps and batch sizes (Appendix A), not computation. The following subsection quantifies the overhead in mathematics. Deployment uses only the original-context model.

C.6 Mathematical learning curves and computation

Figure 4 averages three Qwen3-1.7B seed pairs on the sparse-success subset, each with the same ten evaluated checkpoints. Within each pair, both methods use the same training questions at every update and the same 5,000-problem MATH evaluation and decoding. For the FLOPs panel, cumulative costs are averaged across seeds at each common step and divided by GRPO's final mean cost. All runs therefore use the same normalization denominator. BlendPO is charged no prefill reuse, whereas GRPO reuses prefill within each group.

We select the latest common evaluated BlendPO step whose three-seed mean cumulative FLOPs do not exceed GRPO's final three-seed mean under these asymmetric cache assumptions. Accuracy is not used for selection. This rule selects step 104 for all three seeds. Its mean cost is $2.000 \pm 0.052 \times 10^{18}$ FLOPs, compared with $2.065 \pm 0.184 \times 10^{18}$ for final GRPO at step 126, a ratio of means of 96.87%. The next evaluated BlendPO step, 117, exceeds this mean budget (2.438×10^{18} FLOPs). BlendPO at step 104 scores $78.57 \pm 0.65\%$, versus final GRPO's $74.83 \pm 0.25\%$, a 3.74-point difference in means (paired gain 3.73 ± 0.58 points). The budget constraint applies to the three-seed mean. Individual paired cost ratios at step 104 are 90.47%, 97.46%, and 103.84% for seeds 42–44. At the common final step 126, the ratio of mean costs is 1.34. Crediting prefill reuse to both methods, or to neither, gives step-104 ratios of 95.83% and 96.43% and final-step ratios of 1.32 and 1.33. Step 117 exceeds the budget under every scenario, so the selection is unchanged. Figure 4 connects means at common training steps. Shaded bands show accuracy sample SDs across seeds.

Costs vary across seeds with the generated response lengths, including their effects on generation, policy scoring, and updates. For example, GRPO's mean training-response lengths through step 126 are 2,951, 2,630, and 2,527 tokens for seeds 42–44. The corresponding BlendPO means through step 104 are 2,735, 2,603, and 2,656 tokens. Prompt lengths and attention length moments also enter the estimates.

FLOPs are analytical estimates of matrix multiplications from the logged prompt and response lengths of all 756 updates, covering generation, old-policy scoring, and the actor update, with both BlendPO generation branches charged even at zero guidance strength.

D Additional results

D.1 Guidance-strength allocation experiments

Table 12 compares the allocations in Appendix C.3 under matched data, guidance, batch size, training steps, sampling, reward, and original-context objective. Entries report means and sample SDs across training seeds 42–44. The fixed strengths are tested references, not validation-selected optima.

Table 12: Guidance-strength allocations across mathematical and visual reasoning tasks: unguided accuracy (%). AIME uses Avg@16; Geometry3K uses MathRuler. Fully guided (8G) and 4U+4G use discrete guidance.

	Mathematics			Visual reasoning	
Allocation	MATH	AIME 24–25	AIME 26	Geometry3K	PuzzleVQA
Unguided (GRPO)	74.83 ± 0.25	17.67 ± 0.42	11.46 ± 1.46	56.46 ± 0.58	52.08 ± 4.54
Fully guided (8G)	65.55 ± 2.12	7.85 ± 1.75	5.07 ± 1.58	43.48 ± 12.30	49.25 ± 1.95
BlendPO, fixed $\alpha = 0.25$	78.84 ± 0.81	21.94 ± 0.79	15.21 ± 1.46	60.95 ± 1.95	57.67 ± 1.46
BlendPO, fixed $\alpha = 0.5$	78.37 ± 0.78	19.65 ± 1.15	15.49 ± 3.30	60.73 ± 4.16	57.67 ± 3.41
BlendPO, fixed $\alpha = 0.75$	75.47 ± 1.89	16.28 ± 1.72	12.01 ± 1.89	62.62 ± 0.79	52.50 ± 2.95
BlendPO, per-prompt sweep	79.21 ± 0.74	21.56 ± 1.26	17.43 ± 0.63	61.12 ± 1.83	62.17 ± 4.30
4U+4G	78.29 ± 1.83	21.22 ± 1.89	15.70 ± 2.81	61.12 ± 0.82	61.25 ± 2.65
BlendPO, within-group sweep	79.40 ± 0.39	21.15 ± 0.45	16.18 ± 0.32	63.56 ± 0.83	65.25 ± 1.30

Mathematics. All allocations use the same hint text, framing, and final step (126). Fixed $\alpha = 0.25$ and 0.5, the per-prompt sweep, 4U+4G, and the within-group sweep improve over GRPO by similar margins, while fixed $\alpha = 0.75$ stays close to GRPO. Fully guided training falls well below GRPO on all three benchmarks.

Geometry3K. Geometry3K uses the selected 1,078-problem pool with batch size 8 for 134 steps. All BlendPO allocations and 4U+4G improve over GRPO. The within-group sweep scores highest, slightly above the strongest fixed strength. Fully guided training falls below GRPO and varies widely across seeds.

PuzzleVQA. All allocations use the same 400-question, image-only evaluation. The within-group sweep scores above every fixed strength, the per-prompt sweep, and 4U+4G. Fixed $\alpha = 0.75$ stays close to GRPO, and fully guided training falls below it.

Maze8. Figure 3 in Section 4.4 reports the Maze8 allocations. It evaluates final checkpoints on the same held-out 512-question set with unguided greedy decoding, reporting means and sample SDs across training seeds 42–44.

Leg counting. This additional experiment uses Reasoning Gym v0.1.25 (Stojanovski et al., 2025) with 1,280 training and 320 evaluation problems generated at seed 90,000, each listing 7–12 species with 2–20 instances per species. Qwen3-1.7B is trained with the LoRA settings in Table 5 for 40 steps with batch size 32, and evaluation is greedy at 512 tokens with exact integer-answer scoring. The guidance is a fixed table, shared by all training problems, that groups 37 species by leg count, from birds with two legs to woodlice with fourteen, with an instruction to multiply counts by the table entries and sum them. It supplies arithmetic structure and factual values but no instance-specific products, partial sum, or answer. Leg counting thus provides a controlled case in which part of the needed information appears only in the guidance, and evaluation removes the table. Table 13 compares the same allocations. Of the 320 evaluation problems, 85 include woodlice. GRPO solves none of them, whereas the within-group sweep solves 94.51% and 4U+4G 90.59%, so mixing strengths within a group carries a fact available only in the guidance over to the original context. The within-group sweep achieves the highest overall accuracy with the smallest variation across seeds, and 4U+4G comes close to it on both columns. Fully guided training and fixed $\alpha = 0.5$ and 0.75 fall below GRPO and rarely solve the woodlouse problems. Fixed $\alpha = 0.25$ and the per-prompt sweep improve over GRPO but stay below the within-group sweep, and the per-prompt sweep varies widely across seeds on the woodlouse subset. This pattern is consistent with the hypothesis in Section 4.4 that varying the strength within a group matters most when the guidance largely determines whether a response succeeds.

Table 13: Guidance-strength allocations on leg counting: unguided accuracy (%) with Qwen3-1.7B at the final checkpoint, mean ± SD across seeds 42–44. The woodlouse column is the subset of the 320 evaluation problems that include woodlice, whose leg count is given in the species table. Bold marks the highest score in each column.

Allocation	All (320)	With woodlouse (85)
Unguided (GRPO)	65.52 ± 9.92	0.00 ± 0.00
Fully guided (8G)	33.54 ± 3.21	0.39 ± 0.68
4U+4G	93.96 ± 0.79	90.59 ± 2.04
BlendPO, fixed $\alpha = 0.25$	82.92 ± 2.60	66.67 ± 5.56
BlendPO, fixed $\alpha = 0.5$	27.50 ± 2.67	2.75 ± 4.75
BlendPO, fixed $\alpha = 0.75$	28.85 ± 9.55	0.00 ± 0.00
BlendPO, per-prompt sweep	87.60 ± 6.96	70.98 ± 21.77
BlendPO, within-group sweep	$\mathbf{95.94 \pm 0.31}$	$\mathbf{94.51 \pm 1.80}$

D.2 Token-level importance correction

BlendPO omits the importance ratio in Equation 4. We compare it with a variant that multiplies the original-context token-level update terms by the untruncated ratio, in mathematics with solution prefixes and abstract hints and on Maze8 with visual states.

Table 14: Effect of token-level importance correction on BlendPO: unguided accuracy (%). Updates with correction multiply by the untruncated ratio in Equation 4. AIME uses Avg@16. Bold marks the better mean within each guidance condition.

Training guidance	Update	MATH	AIME 24–25	AIME 26
Solution prefix	w/ correction	$\mathbf{72.26 \pm 0.71}$	$\mathbf{13.85 \pm 0.73}$	$\mathbf{10.00 \pm 1.04}$
Solution prefix	w/o correction	60.67 ± 3.70	5.63 ± 2.53	3.68 ± 2.41
Abstract hint	w/ correction	73.51 ± 0.83	16.77 ± 0.75	10.97 ± 0.73
Abstract hint	w/o correction	$\mathbf{79.40 \pm 0.39}$	$\mathbf{21.15 \pm 0.45}$	$\mathbf{16.18 \pm 0.32}$

In mathematics, the comparison uses the training recipe of BlendPO in Section 4.2 (Table 14). The solution prefixes are the same offline-selected prefixes used by POPE and OC-GRPO (Appendix A.3). With solution prefixes, omitting the correction lowers mean MATH accuracy from the base model's 70.44% to 60.67%, while the correction raises it to 72.26%. With abstract hints, the correction reaches 73.51%, below GRPO's 74.83%, while omitting it reaches 79.40%.

On Maze8, this comparison uses the within-group sweep and multiplies the original-context token-level update terms by the importance ratio $w_{i,t} = \pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t}) / q_{\alpha_i,t}(y_{i,t} \mid x, g, y_{i,<t})$ from Equation 4. This directly weights tokens by their original-context versus blended sampling probabilities. Table 15 evaluates both arms at the final checkpoint on the same held-out 512-question diagnostic set as Figure 3.

Table 15: Maze8 token-level importance correction: unguided accuracy (%) at the final checkpoint on the same held-out 512-question diagnostic set as Figure 3. Both conditions use the within-group sweep. The corrected arm multiplies token-level update terms by Equation 4.

Update	Shortest path (%)
w/ correction	16.34 ± 9.04
w/o correction	69.14 ± 0.90

The correction substantially lowers mean shortest-path accuracy. With abstract hints and visual states, the correction lowers accuracy, whereas with solution prefixes it is needed to improve over the base model. These comparisons do not identify whether the loss under the correction arises from downweighting useful responses or from the variance of the weights.

D.3 MAXIMUM RESPONSE LENGTH AT EVALUATION

Table 16 gives the three-seed accuracy values in Figure 4c at 8k, 16k, and 32k for seeds 42–44. For each method and seed, all three lengths evaluate the same step-126 checkpoint on the same 5,000 MATH problems. The trained weights and decoding settings remain fixed, and only the maximum response length and corresponding model context limit change. OC-GRPO and BlendPO use abstract-hint training guidance. Generated-token counts are reported for the representative comparison below and are not included in the three-seed summary.

Table 16: MATH accuracy (%) at maximum response lengths of 8k, 16k, and 32k: mean ± sample SD across seeds 42–44 on 5,000 problems. OC-GRPO and BlendPO use abstract-hint training guidance; evaluation is unguided. For each method and seed, all lengths use the same step-126 checkpoint and decoding settings apart from the maximum response length and corresponding context limit (Appendix D.3). Bold marks the highest mean per length.

Policy	8k	16k	32k
GRPO	74.83 ± 0.25	74.95 ± 0.88	75.11 ± 0.69
OC-GRPO (abstract hint)	77.85 ± 0.88	78.29 ± 0.92	78.34 ± 0.85
BlendPO, within-group sweep	$\mathbf{79.40 \pm 0.39}$	$\mathbf{81.89 \pm 0.23}$	$\mathbf{83.44 \pm 0.99}$

Table 17: MATH accuracy (%) and mean generated tokens at step 126 (representative seed-42 runs; 5,000 problems). Only the maximum response length changes; training uses 8,192 tokens. Bold marks the highest accuracy per length.

	8k		16k		32k	
Policy	Acc.	Tokens	Acc.	Tokens	Acc.	Tokens
GRPO	75.12	1,512	75.02	1,298	75.12	1,319
BlendPO, within-group sweep	**79.82**	1,954	**82.12**	2,906	**84.52**	3,506
Fully guided (8G)	66.20	3,275	70.60	5,192	75.28	7,813

Table 17 compares representative seed-42 policies at step 126. The guided allocations share abstract hints, private-thought framing, and the original-context GRPO surrogate under the training settings in Appendix A. For each policy, weights, the 5,000 MATH questions, and decoding settings remain fixed, with one response per problem, and only the maximum response length and corresponding model context limit change.

At an 8k maximum response length, the fully guided policy reaches 66.20% at step 126, with 27.1% of responses reaching the limit. Extending the limit to 32k partly recovers accuracy to 75.28%, with 7,813 mean output tokens and 12.44% of responses still reaching the limit. It remains below the within-group sweep while using more than twice its mean output tokens. The within-group sweep uses 3,506 tokens at 32k, versus GRPO's 1,319. Its gains therefore also involve more realized inference computation, so a common maximum response length does not match inference FLOPs.