EFFICIENT LLM REASONING VIA ANSWER-GUIDED POSTERIOR-TO-PRIOR TRANSFER

Anonymous authors
Paper under double-blind review

ABSTRACT

Although large language models achieve strong reasoning capabilities through reinforcement learning, they often generate unnecessarily long reasoning trajectories, leading to substantial inference costs. Conventional length-based reward engineering curtails redundancy but overlooks the fundamental sampling bottleneck: correct-yet-concise trajectories remain severely sparse under question-only exploration, hindering further gains in reasoning efficiency. Inspired by cognitive science, where knowledge of outcomes suppresses ineffective trial-and-error, we leverage training-time reference answers as privileged signals to guide exploration. We theoretically prove that, under a non-negative covariance condition between correctness likelihood and comprehensive utility, the ideal Bayesian answer-conditioned posterior yields no lower expected utility than the question-only prior, with strict improvement when the covariance is positive, providing a principled foundation for answer-guided exploration. To bridge the posterior-prior gap and prevent the transfer of privileged shortcuts, we propose VPG-EA (Variational Posterior Guidance and Efficiency Awareness). Motivated by a variational lower-bound perspective, VPG-EA uses Cross-View Validation to select trajectories that remain predictive under the question-only view, and Advantage-Gated Posterior-to-Prior Transfer to selectively distill validated efficient reasoning patterns into the question-only prior. Experiments across 1.5B and 7B models demonstrate consistent accuracy-efficiency improvements: VPG-EA improves the comprehensive efficiency metric ε^3 by 8.5% and 17.2%, respectively, reduces token consumption by 29–44%, maintains or improves accuracy, and generalizes to non-mathematical scientific domains.

1 INTRODUCTION

Large language models (LLMs) have made remarkable strides in mathematical and logical reasoning through reinforcement learning (RL) (Lightman et al., 2024; Yu et al., 2025). However, these performance gains frequently come at a substantial inference-time cost. Reasoning models tend to generate unnecessarily long chain-of-thought (CoT) trajectories (Luo et al., 2025a; Yeo et al., 2025; Shen et al., 2025), particularly on relatively simple problems: they repeatedly verify intermediate conclusions, revisit alternative solution paths, or continue redundant exploration even after a valid answer has been derived. This overthinking phenomenon (Chen et al., 2024) inflates token consumption and inference latency, spurring a fast-growing line of research on reasoning efficiency optimization. A dominant paradigm in this line of work incorporates length-aware rewards or constraints into RL objectives, encouraging models to preserve reasoning correctness while curbing redundant computation (Luo et al., 2025a; Yeo et al., 2025; Shen et al., 2025).

However, this optimization problem is not merely a matter of choosing a stronger length penalty. From a broader RL perspective, efficient reasoning is inherently a multi-objective optimization problem, requiring the policy to jointly optimize task correctness and reasoning cost (Zhang et al., 2025), while these objectives may favor different behaviors. More importantly, reward design and exploration address two fundamentally different questions: reward design determines what learning signal sampled trajectories should receive, whereas exploration determines which trajectories are sampled and enter the learning process in the first place (Lin et al., 2026b). For efficient reasoning, a useful trajectory must satisfy both correctness and conciseness, making such dual-constraint trajectories significantly harder to discover under the standard question-only policy (Sui et al., 2025). In this setting, further refining the length-based reward can only change how existing samples are evaluated,

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.

but cannot directly compensate for the lack of effective trajectories being sampled. Therefore, efficient reasoning requires not only optimizing the reward function, but also improving the training-time sampling distribution.

To alleviate this exploration bottleneck, we draw inspiration from cognitive-science research: knowledge of outcomes enables humans to suppress ineffective trial-and-error attempts and reorganize more efficient reasoning processes (Hoffrage et al., 2000). Accordingly, we further investigate whether reference answers available during training can serve as a privileged guiding signal to help the model uncover efficient reasoning trajectories that are difficult to discover via question-only exploration (Zelikman et al., 2024; Chen et al., 2026a). We theoretically prove that under a mild correlation condition, the ideal answer-conditioned reasoning distribution yields higher expected comprehensive utility than the question-only prior, providing a principled theoretical foundation for answer-guided training-time exploration. Nevertheless, this distribution cannot be directly deployed for inference: reference answers are inaccessible at deployment time, and answer-conditioned exploration may produce shortcuts that rely on privileged information, which are hard to transfer to the question-only reasoning setting (Feng et al., 2024).

To bridge this gap, we propose VPG-EA (Variational Posterior Guidance and Efficiency Awareness), which treats the answer-conditioned posterior purely as a training-time guide while inference stays question-only. VPG-EA combines three modules: Posterior-Guided Exploration for discovering high-utility trajectories, Cross-View Validation for filtering inference-incompatible shortcuts, and Advantage-Gated Posterior-to-Prior Transfer for migrating validated patterns into the prior. Figure 1 summarizes the resulting improvements, and our contributions follow.

- We formalize efficient reasoning as expected utility optimization jointly accounting for correctness and efficiency, and prove that an answer-conditioned posterior achieves higher expected utility than the question-only prior under a mild correlation condition.
- We propose VPG-EA, a posterior-guided framework that transfers validated efficient reasoning patterns from an answer-conditioned posterior to the question-only prior through cross-view validation and advantage-gated posterior-to-prior transfer.
- Experiments on 1.5B and 7B models show that VPG-EA improves the comprehensive efficiency metric ε^3 by 8.5% and 17.2%, respectively. It reduces token consumption by 29–44%.

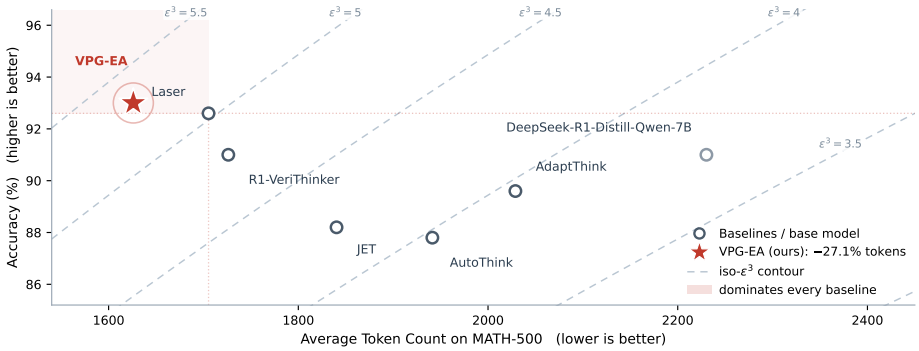

Figure 1: Efficiency landscape on MATH-500 with the 7B model. Dashed curves denote iso-ε^3 contours, where $\varepsilon^3 = \mathrm{ACC}^2/\mathrm{A.Tok}$; moving toward the upper-left corresponds to a better accuracy–efficiency trade-off. The shaded quadrant marks the region outperforming all baselines. VPG-EA achieves the highest accuracy and substantial token reduction.

2 Related Work

Efficiency Optimization via Reward Engineering With the emergence of overthinking in long CoT reasoning, recent studies primarily regulate reasoning length through RL reward engineering. One category designs refined length-penalty functions to guide shortening within standard RL frameworks (Luo et al., 2025a; Yeo et al., 2025; Arora & Zanette, 2025). Another explores difficulty-adaptive or multi-stage pruning, such as difficulty-budget calibration (Shen et al., 2025), decoupling

reasoning patterns from generation (Fang et al., 2025), and dynamic length scheduling (Hou et al., 2026; Tu et al., 2025; Aggarwal & Welleck, 2025) and self-aware boundary adaptation (Chen et al., 2026b). While these methods effectively shorten average CoT length, severe length penalization exacerbates the extreme sparsity of samples that are simultaneously correct and concise. This causes traditional RL algorithms relying on random sampling to face severe sample efficiency bottlenecks during exploration, making it extremely difficult to reach and learn the narrow efficient reasoning manifold through question-only exploration.

Reference-answer-guided Exploration Reference answers are also used as privileged guidance for exploration: STaR (Zelikman et al., 2024) and Nudging (Chen et al., 2026a) construct positive reasoning signals from correct answers, while ExPO (Zhou et al., 2025), RAVR (Lin et al., 2026b), and concurrent work (Zhou et al., 2026) formulate the guidance through probabilistic or variational objectives. However, these approaches primarily target reasoning capability or correctness, while the correctness-efficiency trade-off induced by overthinking remains less explicitly modeled. In contrast, we formulate efficient reasoning as expected utility over correctness and efficiency, use the answer-conditioned posterior as a variational proxy for high-utility reasoning, and selectively transfer inference-compatible patterns to the question-only prior through cross-view validation and advantage-gated posterior-to-prior transfer; Appendix A compares these five methods with ours, including a detailed comparison with the closest of them (RAVR).

3 Problem Formulation and Answer-Guided Exploration

Let $\mathcal{D} = \{(x_i, y_i^*)\}_{i=1}^N$ denote a verifiable reasoning dataset, where x is a problem and y^* is its reference answer. Given x, an LLM first generates a reasoning trajectory z and then produces an answer y. We factorize the generation process as $P_\theta(y, z|x) = P_\theta(y|x, z)\pi_\theta(z|x)$, where θ denotes the model parameters and $\pi_\theta(z|x)$ denotes the question-only reasoning distribution available at inference time.

3.1 Efficient Reasoning as Utility Maximization

Existing reasoning optimization methods mainly improve correctness, while recent efforts to alleviate overthinking further penalize excessively long reasoning paths (Luo et al., 2025a; Shen et al., 2025; Sui et al., 2025). However, directly imposing a strong length constraint introduces a more severe exploration problem: may occupy a sparse subset of the reasoning space (Gupta et al., 2022). Thus, the central objective is not merely to find correct reasoning, but to discover reasoning paths that achieve a favorable accuracy-efficiency trade-off. To explicitly characterize this trade-off, we define the correctness likelihood of a trajectory z as $L(z) = P_\theta(y^*|x, z)$, and introduce an efficiency function $H(z)$, which decreases as the reasoning length increases. The comprehensive utility of a trajectory is then defined as $S(z) = L(z)H(z)$. Accordingly, the desired inference policy is obtained by maximizing the expected utility under the question-only prior distribution that governs inference:

$$\theta^* = \arg\max_\theta \mathbb{E}_{z\sim\pi_\theta(\cdot|x)}[S(z)]. \tag{1}$$

Equation 1 defines the target objective, while the bottleneck arises because the high-utility region may have limited probability mass under the current question-only sampling distribution.

3.2 Answer-Guided Exploration via a Privileged Posterior

A reference answer provides privileged information during training that is unavailable at inference. We exploit this information to construct an answer-conditioned reasoning distribution $q_\theta^{\text{Bayes}}(z|x, y^*) = P_\theta(z|x, y^*)$. By Bayes' rule,

$$q_\theta^{\text{Bayes}}(z|x, y^*) = \frac{P_\theta(y^*|x, z)\pi_\theta(z|x)}{P_\theta(y^*|x)} \propto L(z)\pi_\theta(z|x). \tag{2}$$

Equation 2 defines the ideal Bayesian posterior induced by the factorized joint model. In practice, VPG-EA constructs an answer-conditioned proposal with a prompt and does not assume it to be

identical to this ideal posterior; Proposition 1 therefore characterizes the ideal reweighting mechanism underlying posterior-guided exploration. Conditioning on the reference answer reweights the prior by the correctness likelihood $L(z)$ rather than by the comprehensive utility $S(z)$, and we formalize this relationship next.

Proposition 1. Expected Utility Advantage of the Ideal Answer-Conditioned Posterior.

Assume $E_{z\sim\pi_\theta}[L(z)] > 0$. Then

$$E_{z\sim q_\theta^{\mathrm{Bayes}}}[S(z)] - E_{z\sim\pi_\theta}[S(z)] = \frac{\mathrm{Cov}_{z\sim\pi_\theta}(L(z), S(z))}{E_{z\sim\pi_\theta}[L(z)]}. \tag{3}$$

Therefore, $E_{q_\theta^{\mathrm{Bayes}}}[S(z)] \geq E_{\pi_\theta}[S(z)]$ whenever $\mathrm{Cov}_{\pi_\theta}(L, S) \geq 0$, with strict improvement when the covariance is positive.

The proof is provided in Appendix B. Proposition 1 characterizes when reference-answer conditioning provides a beneficial exploration bias under the efficiency-aware objective: when correctness likelihood is non-negatively correlated with comprehensive utility, the ideal reweighting induced by $q_\theta^{\mathrm{Bayes}}$ shifts probability mass toward trajectories of higher expected utility.

3.3 Variationally Motivated Posterior-to-Prior Transfer

The unavailability of the reference answer at inference time prevents direct deployment of the privileged posterior q_θ. Consequently, the practical objective is to transfer useful reasoning patterns identified through answer-guided exploration into the question-only distribution π_θ. A variational perspective (Blei et al., 2017) provides a natural structural motivation for this posterior-to-prior transfer. Let $J(\theta) = \mathbb{E}_{z\sim\pi_\theta(\cdot|x)}\left[L(z)H(z)\right]$ denote the expected efficient reasoning utility. We optimize its logarithm and introduce q_θ as an auxiliary distribution: $\log J(\theta) = \log \mathbb{E}_{z\sim\pi_\theta(\cdot|x)}[L(z)H(z)]$. Using a change-of-measure argument followed by Jensen's inequality, we obtain the following variationally motivated evidence lower bound on $\log J(\theta)$:

$$\log J(\theta) \geq \mathcal{L}_{\mathrm{V}} = \mathbb{E}_{z\sim q_\theta(\cdot|x,y^*)}\left[\log L(z) + \log H(z)\right] - D_{\mathrm{KL}}\left(q_\theta(z|x,y^*)\|\pi_\theta(z|x)\right). \tag{4}$$

The derivation is provided in Appendix C. Equation 4 suggests two complementary operations: first, using the privileged answer-conditioned distribution to explore reasoning trajectories with high correctness and efficiency; second, guiding the inference-time distribution to incorporate useful patterns discovered by the privileged distribution. Importantly, we use this lower bound as a structural motivation rather than claiming exact variational inference. This perspective motivates the posterior-to-prior transfer mechanism developed in the next section.

4 Method

Building on the formulation in Section 3, VPG-EA realizes answer-guided efficient reasoning through three stages: posterior-guided exploration, cross-view validation and efficiency-aware utility evaluation, and advantage-gated posterior-to-prior transfer, as outlined in Figure 2.

4.1 Posterior-Guided Exploration

We instantiate the posterior and prior using the same model parameters with differentiated prompts, whose templates are provided in Appendix D. The posterior stream receives a prompt conditioned on (x, y^*) and samples $z^{post} \sim q_\theta(\cdot|x, y^*)$, whereas the prior stream receives a question-only prompt and samples $z^{prior} \sim \pi_\theta(\cdot|x)$. The answer-conditioned stream is therefore treated as a practical privileged proposal inspired by the ideal Bayesian posterior, rather than as an exact realization of $q_\theta^{\mathrm{Bayes}}$. The prior stream represents the reasoning distribution available at inference time, and because reference answers are removed during evaluation, the final model operates solely under it.

4.2 Cross-View Validation and Efficiency-Aware Utility

Equation 4 motivates using the posterior to find useful trajectories and encouraging the prior to absorb them. Direct transfer is nonetheless unsafe: the posterior may exploit the reference answer to

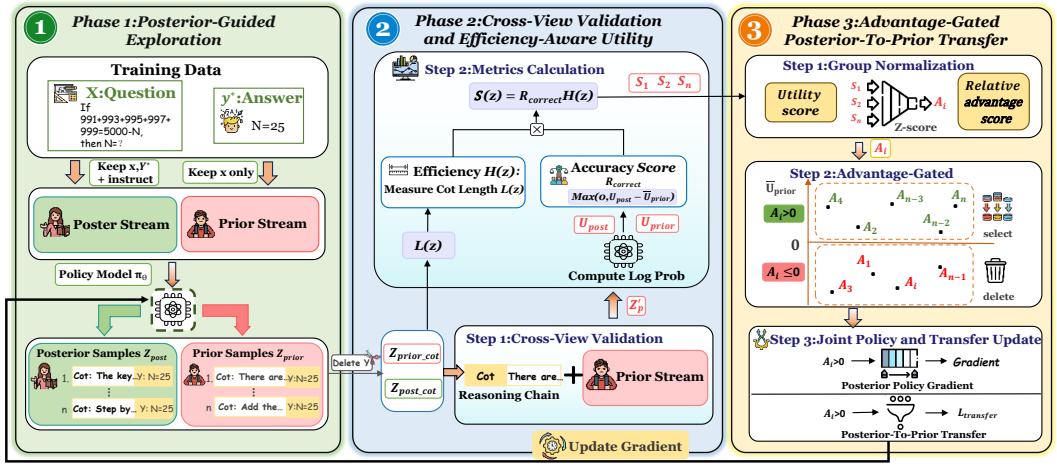

Figure 2: Overview of the VPG-EA framework. The architecture decouples training exploration from inference across three stages: privileged posterior-guided exploration, cross-view validation and efficiency-aware utility evaluation, which filters trajectories that are unreliable under the question-only view, and advantage-gated posterior-to-prior transfer. During inference, the reference answer is discarded, and the policy operates strictly under the question-only prior.

produce trajectories that are not reproducible under inference-time conditions (Feng et al., 2024), so we filter such trajectories by cross-view validation. For each posterior trajectory z^{post}, we remove its generated final answer and evaluate whether the same reasoning remains predictive of y^* under the question-only inference view: $U_{post} = \log P_\theta(y^*|x, z^{post})$. We compare this value against the mean answer likelihood of the concurrently sampled prior trajectories, $\bar{U}_{prior}$, and define:

$$R_{\text{correct}} = \max(0, U_{post} - \bar{U}_{prior}). \tag{5}$$

A posterior trajectory therefore receives a positive correctness signal only when its reasoning remains better than the prior's own samples under the question-only view. We then incorporate the efficiency requirement from Section 3.1 through a relative length coefficient:

$$H(z^{post}) = \left(\frac{L_{base}}{L_{z^{post}}}\right)^{\alpha}, \tag{6}$$

where L_{base} is the average prior trajectory length and α controls the sensitivity to reasoning length. The final posterior utility is

$$\hat{S}(z^{post}) = R_{\text{correct}} H(z^{post}). \tag{7}$$

The two requirements are therefore separated: validity under question-only conditioning, and conciseness. The resulting score acts as an inference-view compatibility filter, selecting trajectories that remain predictive under the question-only evaluation interface while also satisfying the efficiency criterion; it does not by itself establish answer-independence or prove that the prior can autonomously reconstruct the same trajectory.

4.3 Advantage-Gated Posterior-to-Prior Transfer

The lower bound in Equation 4 motivates transferring useful posterior trajectories toward the question-only prior. In practice, absolute utilities are not directly comparable across problems of different difficulty(Su et al., 2026). We therefore normalize the utilities within each sampled group:

$$A_i = \frac{\hat{S}_i - \text{mean}(\hat{S})}{\text{std}(\hat{S}) + \epsilon}. \tag{8}$$

Only posterior trajectories with positive relative advantage are used for posterior-to-prior transfer. Motivated by the negative KL term in Equation 4, we use an advantage-gated likelihood-transfer surrogate:

$$\mathcal{L}_{\text{transfer}} \approx \frac{1}{G} \sum_{i=1}^{G} \mathbf{1}(A_i > 0) \left[\text{sg} \left[\log q_\theta(z_i|x, y^*) \right] - \log \pi_\theta(z_i|x) \right], \tag{9}$$

where $\text{sg}[\cdot]$ denotes stop-gradient. This surrogate is not an unbiased Monte Carlo estimator of the KL divergence, because advantage gating and stop-gradient change both its sampling and its gradient structure. The gating prevents low-utility trajectories from being transferred, and the stop-gradient makes the explicit transfer term one-sided at the gradient level: the posterior guides the prior, so that the prior cannot distort the privileged exploration distribution through this term. Because the two streams share parameters, the overall optimisation is nonetheless not strictly unidirectional. Meanwhile, the posterior stream is optimized toward positive-advantage trajectories. Combining posterior exploration with posterior-to-prior transfer gives

$$\mathcal{L}_{\text{total}} \approx -\frac{1}{G} \sum_{i=1}^{G} A_i \log q_\theta(z_i|x, y^*) + \beta \mathcal{L}_{\text{transfer}}. \tag{10}$$

The first term searches for higher-quality efficient reasoning under the privileged posterior; the second transfers only validated improvements to the prior. VPG-EA thus implements the two operations identified in Section 3, and the complete training procedure is provided in Appendix E.

5 Experiments

5.1 Setup

Models and Baselines VPG-EA is instantiated using DeepSeek-R1-Distill-Qwen-1.5B and 7B (Guo et al., 2025) as the backbone models. Cross-architecture generalization is verified with DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025) (see Appendix F). These base models are also adopted by multiple strong baselines varying in training paradigms and datasets: AdaptThink (Zhang et al., 2025), Laser (Liu et al., 2026), AutoThink (Tu et al., 2025), and JET (Han et al., 2026) rely solely on RL; Thinkless (Fang et al., 2025) combines SFT with RL; and R1-VeriThinker (Chen et al., 2025) utilizes only SFT. Notably, AdaptThink, Laser, AutoThink, and Thinkless share the same training data source as the proposed method. Implementation details and baseline descriptions are provided in Appendix G.

Datasets and Evaluation For training, the DeepScaleR dataset (Luo et al., 2025b) is employed, which contains approximately 40,000 high-quality math question-answer pairs with difficulty grading. For evaluation, a diverse mathematical reasoning test set with increasing difficulty is constructed, including GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021), AIME 2024 [1], and AIME 2025 [2]. To verify generalization beyond mathematics, two non-mathematical domain datasets are introduced: GPQA-Diamond (Rein et al., 2023) and an MMLU-PRO subset (Wang et al., 2024) created by randomly sampling 50 questions from each non-mathematical discipline. Pass@1 (Chen et al., 2021) Accuracy (ACC) and Average Tokens (A.Tok) in the CoT are reported. To quantify the trade-off, the comprehensive metric $\varepsilon^3 = \frac{ACC^2}{A.Tok}$ (Lin et al., 2026a) is adopted, which penalizes redundancy while prioritizing correctness.

[1] https://huggingface.co/datasets/math-ai/aime24

[2] https://huggingface.co/datasets/math-ai/aime25

5.2 Comparative Experimental Results

Table 1 demonstrates VPG-EA's significant and consistent advantages across both parameter scales. On the 1.5B and 7B models, VPG-EA achieves average comprehensive metrics ε^3 of 4.10 and 5.44, outperforming the best baselines by 8.5% and 17.2% and the base models by 68.0% and 37.4%, respectively. Notably, these performance margins widen as task difficulty and model parameter scale increase, with VPG-EA attaining the highest ε^3 on six of the eight benchmark cells. This dominance persists across diverse training paradigms: while composite strategies show marginal gains, they significantly underperform VPG-EA, confirming that addressing the fundamental exploration bottleneck through a bottom-up RL paradigm yields far greater improvements than surface-level reward engineering or data-tuning combinations. VPG-EA also achieves state-of-the-art or competitive results across nearly all metrics against baselines sharing the identical RL setting and datasets, and it significantly outperforms the composite strategies.

Table 1: In-domain comparison on mathematical reasoning benchmarks. †denotes methods trained with the same data and RL paradigm. Paired significance tests for all comparisons are reported in Appendix K, and the inference and training cost of every method is reported in Appendix L, since an efficiency gain is only meaningful when it is not offset by the cost of producing it.

Method	GSM8K			MATH-500			AIME24			AIME25			Average		
	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑
DeepSeek-R1-Distill-Qwen-1.5B															
Original Model	77.26	756.53	7.89	75.00	3098.02	1.82	20.00	11524.67	0.03	16.67	12845.30	0.02	47.23	7056.13	2.44
Thinkless	**83.70**	645.84	10.85	81.20	1936.46	3.40	30.00	7004.40	0.13	16.67	7661.17	0.04	52.89	4311.97	3.61
JET	78.01	**505.85**	12.03	**83.80**	2397.60	2.93	26.67	**6159.30**	0.12	16.67	**5349.23**	0.05	51.29	**3603.00**	3.78
AdaptThink†	80.67	614.97	10.58	79.20	1960.60	3.20	23.33	7596.63	0.07	**26.67**	7365.07	**0.10**	52.47	4384.32	3.49
Laser†	79.15	666.62	9.40	81.60	1911.50	3.48	23.33	8291.07	0.07	20.00	6695.57	0.06	51.02	4391.19	3.25
AutoThink†	**83.70**	655.84	10.68	79.00	**1845.33**	3.38	26.67	6325.20	0.11	20.00	7602.73	0.05	52.34	4107.28	3.56
VPG-EA†	80.89	519.94	**12.58**	82.40	1891.80	**3.59**	**33.33**	6659.73	**0.17**	20.00	6678.70	0.06	**54.16**	3937.54	**4.10**
DeepSeek-R1-Distill-Qwen-7B															
Original Model	89.99	695.06	11.65	91.00	2230.23	3.71	53.33	8208.73	0.35	33.33	10355.10	0.11	66.91	5372.28	3.96
R1-VeriThinker	86.13	626.59	11.84	91.00	1726.20	4.80	53.33	6127.47	0.46	**36.67**	6364.10	**0.21**	66.78	3711.09	4.33
JET	87.87	558.62	13.82	88.20	1840.30	4.23	43.33	7181.60	0.26	33.33	9580.33	0.12	63.18	4790.21	4.61
AdaptThink†	90.98	687.24	12.04	89.60	2028.86	3.96	43.33	8044.07	0.23	**36.67**	7998.13	0.17	65.15	4689.58	4.10
Laser†	91.80	644.66	13.07	92.60	1705.36	5.03	40.00	**4575.00**	0.35	26.67	**5944.73**	0.12	62.77	**3217.44**	4.64
AutoThink†	89.46	809.37	9.89	87.80	1941.32	3.97	40.00	6125.50	0.26	30.00	6852.53	0.13	61.82	3932.18	3.56
VPG-EA†	**92.00**	**537.87**	**15.74**	**93.00**	**1626.07**	**5.32**	**56.67**	6352.83	**0.51**	**36.67**	6762.60	0.20	**69.59**	3819.84	**5.44**

5.3 Generalization Experimental Results

Table 2 shows that, although VPG-EA does not achieve the optimum on MMLU-Pro at the 1.5B scale, it approaches the strongest baselines, and that at 7B almost all metrics rank first or second. These results indicate that VPG-EA generalizes well and that its advantage grows with parameter scale: the model avoids catastrophic forgetting in non-mathematical disciplines and, by eliminating the redundancy caused by overthinking, improves direct answering in diverse scientific domains.

5.4 Ablation and Hyperparameter Studies

Figure 3 isolates each component of VPG-EA on the 1.5B model and traces the sensitivity of the efficiency weight α. Removing the explicit posterior-to-prior transfer term ($\beta = 0$) is the most damaging: on AIME24 the average token count rises to 9,464 while accuracy falls to 16.67%, and the comprehensive metric ε^3 collapses from 0.17 to 0.03, confirming that the privileged posterior is what makes high-utility trajectories reachable in the first place. Because the two streams share parameters, this ablation removes the explicit transfer loss but does not eliminate all posterior influence on the shared policy. Removing the efficiency term ($\alpha = 0$) trades length for almost no accuracy gain, inflating AIME24 to 8,626 tokens with $\varepsilon^3 = 0.06$. Removing cross-view validation produces the opposite—but equally harmful—failure: GSM8K outputs become *shorter* (448.85 tokens) while accuracy drops to 73.24%, and on AIME24 accuracy falls from 33.33% to 13.33% (ε^3: 0.17→0.03). A shorter yet much less accurate variant is the failure mode that cross-view validation is designed to block: without the question-only check, trajectories that reach the answer only because the reference

Table 2: Out-of-domain generalization on GPQA-Diamond and MMLU-Pro. †denotes methods trained with the same data and RL paradigm.

Method	GPQA-D			MMLU-Pro			Average		
	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑
DeepSeek-R1-Distill-Qwen-1.5B									
Original Model	29.79	7419.90	0.12	24.76	4168.47	0.15	27.28	5794.19	0.14
Thinkless	**32.82**	7465.22	0.14	**31.08**	3342.29	**0.29**	**31.95**	5403.76	**0.22**
JET	26.77	6140.98	0.12	25.08	2928.34	0.21	25.93	4534.66	0.17
AdaptThink†	28.70	**5106.90**	**0.16**	23.23	3022.77	0.18	25.97	**4064.84**	0.17
Laser†	30.80	6695.57	0.14	25.38	**2242.31**	**0.29**	28.09	4468.94	**0.22**
AutoThink†	26.77	5768.08	0.12	21.23	3544.66	0.13	23.97	4656.37	0.13
VPG-EA†	30.80	6237.98	0.15	25.23	3030.35	0.21	28.02	4634.17	0.18
DeepSeek-R1-Distill-Qwen-7B									
Original Model	36.86	6471.26	0.21	47.53	4115.27	0.55	42.20	5293.27	0.38
R1-VeriThinker	36.86	4060.43	0.33	43.85	4156.93	0.46	40.36	4108.68	0.40
JET	36.89	**2915.79**	**0.47**	**49.54**	3916.35	0.63	43.22	**3416.07**	**0.55**
AdaptThink†	**48.98**	7124.44	0.34	48.46	3696.78	0.64	**48.72**	5410.61	0.49
Laser†	33.83	3603.66	0.32	43.07	4029.99	0.46	38.45	3816.83	0.39
AutoThink†	34.85	4763.04	0.25	44.77	4270.04	0.47	39.81	4516.54	0.36
VPG-EA†	38.88	3892.62	0.39	48.77	**3342.44**	**0.71**	43.83	3617.53	**0.55**

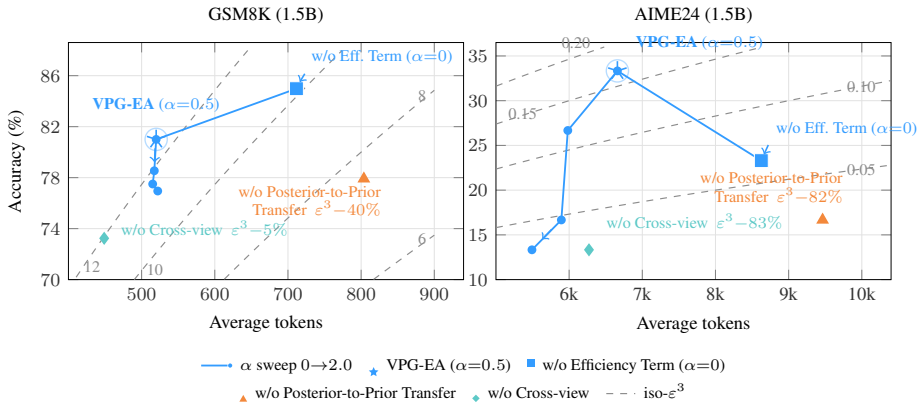

Figure 3: **Ablation and hyperparameter studies on 1.5B model.** Blue: α sweep ($\alpha = 0 \to 2.0$, default $\alpha = 0.5$ as $\star$, with the $\alpha = 0$ end of the sweep—removing the efficiency term—marked by a ■). Orange: w/o posterior-to-prior transfer (▲); teal: w/o cross-view validation (♦). Dashed curves are iso-ε^3 contours (up-left is better).

answer was visible are rewarded by the transfer term and absorbed by the inference-time policy. This is worst on the hardest benchmark.

6 Bottleneck Diagnostics

Figure 4 summarizes the diagnostics: panels (a–d) compare the posterior q with the prior π across the sampling budget, and panels (e–h) track the training dynamics.

Bottleneck. Across the sampling budget $N \in [1, 64]$, the posterior distribution q consistently outperforms the prior distribution π in both PASS@N and served length $\ell_p(N)$. As the number of samples increases, PASS@N under q improves while the served length further decreases. For example, on AIME 2024, the served length of q decreases from 8213 tokens at $N = 1$ to 3505 tokens at $N = 64$. This result indicates that the answer-conditioned posterior can discover correct and shorter reasoning trajectories more frequently than the question-only prior, while additional sampling under q can still uncover even shorter correct paths.

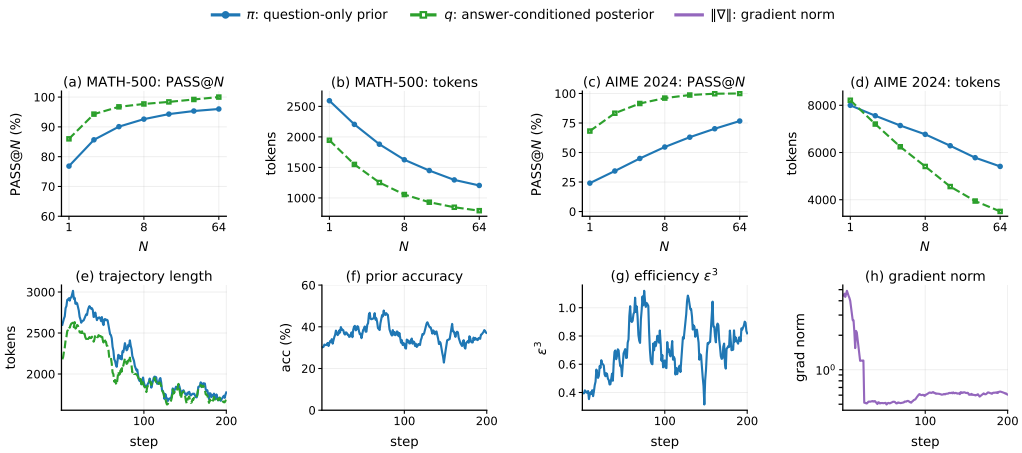

Figure 4: **Bottleneck diagnostics and training dynamics.** (a,c) PASS@N: probability that at least one of N samples is correct. (b,d) *Served length* $\ell_p(N) = \mathbb{E}_x \mathbb{E}_{z_1,\ldots,z_N \sim p}[\min_{i:\,\mathrm{correct}(x,z_i)} |z_i|]$, the answer a verifier serves when it keeps the shortest correct chain (∞ if none is correct), for the question-only prior π (solid) and the answer-conditioned posterior q (dashed). Panels (b,d) use only problems both distributions solve within the budget ($96/100$ MATH-500, $23/30$ AIME 2024), and ℓ_p is non-increasing in N by construction, so levels and rates are compared. (e–h) Training dynamics (10-step moving average, batch statistics): trajectory length, prior accuracy, efficiency proxy $\varepsilon^3 = \mathrm{acc}^2/(\mathrm{A.Tok}) \times 10^4$, and gradient norm (log scale).

Training Dynamics. During training, the trajectory length of the prior decreases from 2438 to 1823 tokens, while batch accuracy increases from 26.6% to 32.8% and the comprehensive efficiency metric ϵ_3 improves from 0.29 to 0.59. Meanwhile, the gradient norm decreases from 3.4 and stabilizes at around 0.6. These results show that the prior improves simultaneously in reasoning length, accuracy, and comprehensive efficiency, while the gradient norm remains stable in the later stages of training.

7 Limitations

Although VPG-EA synergizes efficiency and accuracy, limitations remain. First, the ideal Bayesian posterior of our analysis and the prompt-conditioned proposal used in practice are not assumed to be identical, and this gap is not quantified; Cross-View Validation assesses inference-view predictiveness but does not guarantee answer-independence or autonomous reconstruction of the same trajectory. Second, because the two streams share parameters, the stop-gradient makes only the explicit transfer term one-sided rather than the overall optimization strictly unidirectional. Finally, reward hacking remains a general limitation of proxy-based RL (Wen et al., 2025; Skalse et al., 2023), and the framework still depends on verifiable tasks with reference answers, limiting its use in open-domain settings.

8 Conclusion

This paper addresses the overthinking problem in LLM reasoning by identifying a sampling bottleneck: correct and efficient trajectories are sparse under question-only exploration. We prove that, under a mild correlation condition, the ideal Bayesian answer-conditioned posterior achieves higher expected comprehensive utility than the question-only prior. Based on this insight, VPG-EA combines posterior-guided exploration, Cross-view Validation, and Advantage-Gated Posterior-to-Prior Transfer to transfer validated patterns to the prior. Experiments across 1.5B and 7B models confirm consistent accuracy-efficiency improvements at both scales, with substantially lower token consumption.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the drafting and language editing of the manuscript, with the implementation of the training and evaluation code, and with generating the code that produces the data-driven figures. The research ideas, the experimental design, the framework figure, and all literature search and citation selection were carried out by the authors without generative AI assistance. We have reviewed all AI-assisted work: the text was checked and edited by the authors, LLM-generated code was inspected, executed, and validated against the reported results, and every data figure was regenerated and visually verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the efficiency of reasoning in publicly available language models using only publicly available benchmarks (GSM8K, MATH-500, AIME, GPQA-Diamond, MMLU-Pro). It involves no human subjects, no personal data, and no newly released datasets. We are not aware of ethical concerns specific to this work beyond the general dual-use considerations that apply to LLM research; the proposed method reduces inference cost and does not introduce new model capabilities.

REPRODUCIBILITY STATEMENT

We describe the complete training procedure in Appendix E, the prompt templates in Appendix D, the hardware, hyperparameters, and evaluation protocol in Appendix G and Appendix K.1, and provide complete proofs of all theoretical claims in Appendices B and C. All benchmarks are publicly available, and all results were produced with the decoding settings reported in Appendix K.1. Code and the evaluation harness will be released upon publication.

REFERENCES

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=4jdIxXBNve.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=AiZxn84Wdo.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Justin Chen, Xiangyu Peng, Prafulla Kumar Choubey, Kung-Hsiang Huang, Jiaxin Zhang, Mohit Bansal, and Chien-Sheng Wu. Nudging the boundaries of LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=hfNnQHkTtv.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Qiguang Chen, Dengyun Peng, Jinhao Liu, Huikang Su, Jiannan Guan, Libo Qin, and Wanxiang Che. Aware first, think less: Dynamic boundary self-awareness drives significant gains in reasoning efficiency in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30261–30269, 2026b.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Zigeng Chen, Xinyin Ma, Gongfan Fang, Ruonan Yu, and Xinchao Wang. Verithinker: Learning to verify makes reasoning model efficient. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Y1bt0YIS6Y.

Xiaoxue Cheng, Junyi Li, Zhenduo Zhang, Xinyu Tang, Xin Zhao, XinYu KONG, and Zhiqiang Zhang. Incentivizing dual process thinking for efficient large language model reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ezOfR26pGQ.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason E Weston. Chain-of-verification reduces hallucination in large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*, 2024. URL https://openreview.net/forum?id=ek3GqAs2uO.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ariVQf0KZx.

Jiaheng Feng, Mingxiao Feng, Haolin Song, Wengang Zhou, and Houqiang Li. Suf: Stabilized unconstrained fine-tuning for offline-to-online reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 11961–11969, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Abhishek Gupta, Aldo Pacchiano, Yuexiang Zhai, Sham Kakade, and Sergey Levine. Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity. *Advances in Neural Information Processing Systems*, 35:15281–15295, 2022.

Jinyi Han, Ying Huang, Ying Liao, Haiquan Zhao, Zishang Jiang, Xinyi Wang, Xikun Lu, Guanghao Zhou, Sihang Jiang, Jiaqing Liang, Weikang Zhou, Zeye Sun, Fei Yu, and Yanghua Xiao. Your models have thought enough: Training large reasoning models to stop overthinking. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=2u5ZRzDyS0.

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.

Ulrich Hoffrage, Ralph Hertwig, and Gerd Gigerenzer. Hindsight bias: A by-product of knowledge updating? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 26(3):566, 2000.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of LLMs via reinforcement learning. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=V51gPu1uQD.

Yao Huang, Huanran Chen, Shouwei Ruan, Yichi Zhang, Xingxing Wei, and Yinpeng Dong. Mitigating overthinking in large reasoning models via manifold steering. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=49Rc51iCso.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Junhong Lin, Xinyue Zeng, Jie Zhu, Song Wang, Julian Shun, Jun Wu, and Dawei Zhou. Plan and budget: Effective and efficient test-time scaling on reasoning large language models. In *International Conference on Learning Representations*, volume 2026, pp. 106869–106893, 2026a.

Tianqianjin Lin, Xi Zhao, Xingyao Zhang, Rujiao Long, Yi Xu, Zhuoren Jiang, Wenbo Su, and Bo Zheng. RAVR: Reference-answer-guided variational reasoning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18757–18775, 2026b.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hj9eKpqxQl.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. In *2nd AI for Math Workshop @ ICML 2025*, 2025a. URL https://openreview.net/forum?id=ioYybCRcyW.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, et al. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*, 3(5), 2025b.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023. URL https://doi.org/10.48550/arXiv.2311.12022.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, 2025.

Joar Max Viktor Skalse, Matthew Farrugia-Roberts, Stuart Russell, Alessandro Abate, and Adam Gleave. Invariance in policy optimisation and partial identifiability in reward learning. In *International Conference on Machine Learning*, pp. 32033–32058. PMLR, 2023.

Yuhao Su, Anwesa Choudhuri, Zhongpai Gao, Benjamin Planche, Van Nguyen Nguyen, Meng Zheng, Yuhan Shen, Arun Innanje, Terrence Chen, Ehsan Elhamifar, et al. Medgrpo: Multi-task reinforcement learning for heterogeneous medical video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2788–2798, 2026.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for large language models. *Trans. Mach. Learn. Res.*, 2025, 2025. URL https://openreview.net/forum?id=HvoG8SxggZ.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in r1-style models via multi-stage RL. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Hs3FrjwyVZ.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290, 2024.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Bowman, He He, and Shi Feng. Language models learn to mislead humans via RLHF. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xJljiPE6dg.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 2550–2575, 2023.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR. *arXiv preprint arXiv:2604.03128*, 2026.

Edward Yeo, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in LLMs. In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*, 2025. URL https://openreview.net/forum?id=AgtQlhMQ0V.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru 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, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. Star: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, volume 1126, pp. 0–55, 2024.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025.

Ruiyang Zhou, Shuozhe Li, Amy Zhang, and Liu Leqi. ExPO: Unlocking hard reasoning with self-explanation-guided reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=D1PeGJtVEu.

Xiangxin Zhou, Zichen Liu, Haonan Wang, Chao Du, Min Lin, Chongxuan Li, Liang Wang, and Tianyu Pang. Variational reasoning for language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=fGGcovg6oW.

APPENDIX CONTENTS

Appendix A. Positioning Relative to Prior Answer-Guided Work 15

Appendix B. Proof and Boundary Conditions for Proposition 1 16

B.1 Notation Setup 16

B.2 Accuracy Advantage under Answer-Conditioned Reweighting 16

B.3 Expected Utility Advantage and Covariance Condition 17

B.4 Summary of the Boundary Condition 18

B.5 Empirical Boundary of the Covariance Condition 18

Appendix C. Utility Lower Bound and Posterior-to-Prior Transfer 19

Appendix D. Prompts 21

D.1 Implementation Details of VPG-EA Prompts 21

D.2 Exploration of the Endogenous Attribution of Efficiency after Removing Instruction Bias 22

Appendix E. Algorithmic Procedure 23

Appendix F. Performance across Different Model Architectures 25

Appendix G. Experimental Details 26

Appendix H. Micro-behavioral Analysis of Reasoning Trajectory 27

Appendix I. Dynamic Analysis of Reasoning Trajectory Length and Posterior-to-Prior Transfer 28

Appendix J. Trade-off Analysis Under Difficulty Grading 28

Appendix K. Significance Analysis 28

K.1 Protocol and Tests 29

K.2 Aggregate Significance on GSM8K and MATH-500 30

K.3 Where the Efficiency Gain Comes From: a Channel Decomposition 30

K.4 Non-inferiority Against the Strongest Baselines 32

K.5 Statistical Power on AIME 32

K.6 Threats to Validity 32

Appendix L. Cost and Latency 33

Appendix M. Case Study 34

A Positioning Relative to Prior Answer-Guided Work

Table 3 compares five representative methods that use reference answers as a training-time privileged signal—STaR, Nudging (NuRL), ExPO, RAVR, and Variational Reasoning—along four dimensions: core objective, role of the privileged signal, shortcut/leakage defence, and weight update. All use the answer to obtain better samples or to define an answer-conditioned posterior, and the two most recent add a training-side guard against answer-dependent reasoning. None, however, puts reasoning length inside its objective, or tests whether each trajectory stays valid without the answer. As they optimise capability rather than this trade-off, we avoid them as baselines and compare against methods designed for it, which share our data source and RL paradigm where possible († in Table 1).

Relation to the closest work (RAVR). The overlap is plain: our teacher prompt follows RAVR's answer-conditioned think-aloud reconstruction template (Appendix D), and neither the privileged posterior nor the posterior-to-prior transfer is claimed as ours, as both already appear in this line of work. What differs is the objective, the guarantee, the mechanism keeping the transferred reasoning valid, and how it is applied.

(i) *Objective.* RAVR optimises a correctness-improvement reward, $R_{\text{impr}} = \max(0, \log \pi_\theta(y^*|x,z) - \mathbb{E}_{\pi_\theta} \log \pi_\theta(y^*|x,z'))$, whose length normalisation only makes answers comparable rather than rewarding brevity; we multiply the same signal by an efficiency coefficient $H(z)$, so a trajectory must be both answer-predictive and short to score highly (Equation 7). (ii) *Guarantee.* RAVR's guarantee holds for a correctness utility and does not extend to an efficiency-aware one: for the ideal Bayesian posterior, Proposition 1 requires a non-negative covariance between correctness likelihood and $S = L \cdot H$ (Appendix B.5 locates where it fails). (iii) *Mechanism.* RAVR suppresses answer-dependent reasoning through a leakage-aware reward; we keep a posterior trajectory only if it stays predictive of y^* under the question-only view. Without this filter, GSM8K outputs shorten to 448.85 tokens while accuracy falls to 73.24% (Figure 3)—unreliable under question-only evaluation. (iv) *Transfer.* RAVR reweights its KL samples by the utility reward; we gate transfer by the relative advantage of a trajectory passing this filter and make the explicit transfer term one-sided at the gradient level with a stop-gradient on the posterior—a design decision, not a theorem.

We do not reproduce RAVR and claim nothing about its performance. Our $\alpha = 0$ ablation isolates the objective type: a correctness-only objective shortens nothing (8,626 vs. 6,660 tokens on AIME24) and does not improve ε^3, and a different student instruction leaves token counts unchanged (516 vs. 520 on GSM8K) while degrading accuracy (78.46 vs. 80.89) (Appendix D.2). In short, we share the insight that the answer is a useful privileged signal, and add the efficiency-aware objective and guarantee, the inference-view compatibility filter, and gated posterior-to-prior transfer that an efficiency target still needs.

Table 3: Positioning of VPG-EA relative to five representative answer-guided methods.

Method	Core objective	Privileged signal role	Shortcut / leakage defence	Weight update
STaR (Zelikman et al., 2024)	Bootstrap reasoning ability	Answer as a hint for backward rationalisation of failed problems	Not addressed	Iterated SFT on rationales that reached the correct answer
Nudging (NuRL) (Chen et al., 2026a)	Learn from problems the model cannot solve	Offline high-level hint from the gold answer, injected for problems with zero pass rate	Empirical only: abstract hints generalise, more answer exposure degrades	Online RL (GRPO) with conditional hint injection
ExPO (Zhou et al., 2025)	Obtain positive samples for RL on hard problems	Self-explanations conditioned on the ground-truth answer	In-distribution samples with higher answer likelihood; no question-only-view test	RL (GRPO/DPO) with a self-explanation guidance term
RAVR (Lin et al., 2026b)	Make high-utility reasoning paths samplable	Answer-conditioned posterior; reward is the improvement over the prior	Leakage-aware reward suppresses post-hoc, answer-dependent reasoning	GRPO with a posterior–prior KL, one shared LLM
Variational reasoning (Zhou et al., 2026)	Tight variational bound over thinking traces	Answer conditions the variational posterior over traces	Forward KL prevents posterior collapse into leaking answer tokens	Forward-KL posterior, then weighted policy updates; no weight sharing
VPG-EA (ours)	**Cut trace length at preserved or improved accuracy**	**Answer-conditioned posterior used only in training and sharing parameters with the prior**	**Cross-view validation: a trajectory is transferred only if it stays predictive of the answer under the question-only view**	**Advantage-weighted policy gradient plus gated explicit posterior-to-prior transfer with one-sided gradient flow**

B Proof and Boundary Conditions for Proposition 1

In this appendix, we provide the proof of Proposition 1 in Section 3.2 and characterize the condition under which the answer-conditioned posterior achieves no lower expected efficiency-aware utility than the question-only prior.

Throughout this appendix, $q(z)$ denotes the ideal Bayesian posterior $q_\theta^{\text{Bayes}}(z|x, y^*)$ defined in Eq. (2).

B.1 Notation Setup

For clarity in derivation, we abbreviate the relevant variables as follows:

- Prior Distribution (Prior): $\pi(z) \triangleq \pi_\theta(z|x)$,
- Posterior Distribution (Posterior): $q(z) \triangleq \pi_\theta(z|x, y^*)$,
- Correctness Likelihood: $L(z) \triangleq P(y^*|x, z)$,
- Comprehensive Utility: $S(z) \triangleq L(z) \cdot H(z)$, where $H(z) > 0$ is the efficiency coefficient defined in Section 3.1.

According to Bayes' theorem, the relationship between the posterior distribution and the prior distribution is:

$$q(z) = \frac{\pi(z)L(z)}{P(y^*|x)}, \tag{11}$$

where the denominator $P(y^*|x)$ is the marginal likelihood, representing the probability of the model answering correctly directly. It can be written in the form of an expectation under the prior:

$$P(y^*|x) = \sum_{z \in Z} \pi(z)L(z) = E_{z\sim\pi}[L(z)]. \tag{12}$$

Therefore, $q(z) = \frac{\pi(z)L(z)}{\mathbb{E}_{z\sim\pi}[L(z)]}$. Thus, conditioning on the reference answer reweights the prior according to the correctness likelihood $L(z)$: trajectories with larger $L(z)$ receive proportionally greater probability mass under the posterior.

B.2 Accuracy Advantage under Answer-Conditioned Reweighting

Before deriving the comprehensive utility advantage, we first mathematically prove an important prerequisite corollary: the expected likelihood of deriving the correct answer using the Bayesian posterior distribution constructed with the reference answer is necessarily greater than or equal to that of the prior distribution.

Expanding the expected correctness likelihood under the posterior distribution:

$$E_{z\sim q(z)}[L(z)] = \sum_{z \in Z} q(z)L(z). \tag{13}$$

Substituting the Bayesian expansion from Section B.1 yields:

$$E_{z\sim q(z)}[L(z)] = \sum_{z \in Z} \left(\frac{\pi(z)L(z)}{E_{z\sim\pi}[L(z)]} \right) L(z) = \frac{\sum_z \pi(z)L^2(z)}{E_{z\sim\pi}[L(z)]} = \frac{E_{z\sim\pi}[L^2(z)]}{E_{z\sim\pi}[L(z)]}. \tag{14}$$

To compare the difference in expected accuracy between the posterior and the prior, we subtract the two:

$$E_{z\sim q(z)}[L(z)] - E_{z\sim\pi}[L(z)] = \frac{E_{z\sim\pi}[L^2(z)] - (E_{z\sim\pi}[L(z)])^2}{E_{z\sim\pi}[L(z)]}. \tag{15}$$

According to the variance formula $Var(X) = E(X^2) - (E(X))^2$ in probability theory, the numerator of the above equation exactly corresponds to the variance of the correctness likelihood $L(z)$ under the prior distribution:

$$E_{z\sim q(z)}[L(z)] - E_{z\sim\pi}[L(z)] = \frac{Var_{z\sim\pi}(L(z))}{E_{z\sim\pi}[L(z)]}. \tag{16}$$

Since the variance of any random variable is strictly non-negative ($Var \geq 0$), and the denominator expectation $E_{z\sim\pi}[L(z)] > 0$ (assuming there exists a solution with a non-zero correctness probability in the prior space), we reach the following mathematical conclusion:

$$E_{z\sim q(z)}[L(z)] \geq E_{z\sim\pi}[L(z)]. \tag{17}$$

This variance theorem establishes the inherent expected advantage of the true posterior distribution in terms of accuracy. However, the efficient reasoning objective of this paper requires not only correctness but also refinement of the reasoning process (introducing length decay $H(z)$). Therefore, whether the high expected correctness rate of the posterior can smoothly translate into the final high expected comprehensive utility $E_{z\sim q(z)}[S(z)]$ requires further covariance derivation.

B.3 Expected Utility Advantage and Covariance Condition

We need to compare the magnitude relationship between the posterior expected utility $E_{q(z)}[S(z)]$ and the prior expected utility $E_\pi[S(z)]$.

Step 1: Expand the posterior expectation By the definition of expectation, the expected utility under the posterior distribution is:

$$E_{z\sim q(z)}[S(z)] = \sum_{z\in Z} q(z)S(z) \tag{18}$$

Step 2: Substitute the Bayesian formula Substituting $q(z) = \frac{\pi(z)L(z)}{P(y^*|x)}$ into the above equation:

$$E_{z\sim q(z)}[S(z)] = \sum_{z\in Z} \left(\frac{\pi(z)L(z)}{P(y^*|x)}\right) S(z). \tag{19}$$

Step 3: Extract the constant term and regroup Since $P(y^*|x)$ is a constant with respect to the summation variable z, it can be factored outside the summation sign. The numerator part becomes the weighted expectation under the prior distribution:

$$E_{z\sim q(z)}[S(z)] = \frac{1}{P(y^*|x)} \sum_{z\in Z} \pi(z)(L(z)\cdot S(z)) = \frac{E_{z\sim\pi}[L(z)\cdot S(z)]}{P(y^*|x)}. \tag{20}$$

Step 4: Represent the marginal likelihood using the prior expectation Replacing the denominator $P(y^*|x)$ with $E_{z\sim\pi}[L(z)]$ (see Section B.1):

$$E_{z\sim q(z)}[S(z)] = \frac{E_{z\sim\pi}[L(z)\cdot S(z)]}{E_{z\sim\pi}[L(z)]}. \tag{21}$$

Step 5: Construct the inequality We need to prove $E_{q(z)}[S(z)] \geq E_\pi[S(z)]$. Substituting the result from Step 4 into the left side of the inequality, we must prove:

$$\frac{E_{z\sim\pi}[L(z)\cdot S(z)]}{E_{z\sim\pi}[L(z)]} \geq E_{z\sim\pi}[S(z)]. \tag{22}$$

Step 6: Transform into covariance form Since the likelihood probability $L(z) \geq 0$, its expectation $E_\pi[L(z)]$ is a positive number. Multiplying both sides of the inequality by $E_\pi[L(z)]$, the direction of the inequality remains unchanged:

$$E_{z\sim\pi}[L(z)\cdot S(z)] \geq E_{z\sim\pi}[L(z)]\cdot E_{z\sim\pi}[S(z)]. \tag{23}$$

Moving the right term to the left side:

$$E_{z\sim\pi}[L(z)\cdot S(z)] - E_{z\sim\pi}[L(z)]\cdot E_{z\sim\pi}[S(z)] \geq 0. \tag{24}$$

Step 7: Apply the definition of covariance According to the definition of covariance in statistics: for any random variables X and Y, $Cov(X,Y) = E[XY] - E[X]E[Y]$. Observing the expression in Step 6, the left side is exactly the covariance of $L(z)$ and $S(z)$ under the prior distribution π:

$$Cov_{z\sim\pi}(L(z), S(z)) \geq 0. \tag{25}$$

B.4 Summary of the Boundary Condition

Through the above derivation, we conclude: the necessary and sufficient condition for Proposition 1 to hold is that under the prior distribution $\pi_\theta(z|x)$, the correctness likelihood $L(z)$ of the reasoning path is non-negatively correlated with the comprehensive utility $S(z)$, namely $Cov_{z\sim\pi}(L(z), S(z)) \geq 0$.

It should be clarified that because the comprehensive utility $S(z) = L(z) \cdot H(z)$ introduces a length-decreasing efficiency term $H(z)$, this non-negative covariance condition is not absolutely guaranteed to hold under all parameter settings. In the high-quality mathematical reasoning tasks established in this paper, when the efficiency coefficient $H(z)$ is subject to reasonable constraints—such as a moderate decay hyperparameter—the likelihood gain brought by the model reaching the correct logical manifold will remain dominant. Under such circumstances, a correct but slightly lengthy high-quality sample will still have a utility $S(z)$ that strictly dominates a vast number of erroneous and brief invalid exploration samples. Within this boundary, $L(z)$ and $S(z)$ maintain a positive correlation, and the advantage of posterior guidance strictly holds mathematically. When the framework adopts extreme hard length truncation, or if the efficiency decay coefficient is excessively large, even if the model derives the correct answer (an increase in $L(z)$), the length cost it incurs will cause a precipitous drop in $H(z)$, resulting in a comprehensive utility $S(z)$ that is actually lower than that of an incorrect but brief path. At this point, $L(z)$ and $S(z)$ will exhibit a negative correlation, and the posterior distribution will lose its expected utility advantage.

B.5 Empirical Boundary of the Covariance Condition

Proposition 1 guarantees the advantage of answer-guided exploration only when $\mathrm{Cov}_{\pi_\theta}(L, S) \geq 0$, and the discussion above characterises that boundary analytically: the condition should hold under a moderate length decay and break down when the efficiency term dominates the likelihood gain. Whether it holds for the models we actually train, and where its boundary lies in practice, is an empirical question. We answer it by measuring the covariance directly on the prior distribution.

Protocol. For each evaluation question x we draw $K = 8$ independent trajectories $z_1, \ldots, z_K \sim \pi_\theta(\cdot \mid x)$ with the same student prompt, decoding temperature ($T = 1.0$) and top-p (0.95) used in training, i.e. the same decoding configuration that Proposition 1 refers to. The correctness likelihood $L(z) = P_\theta(y^* \mid x, z)$ is obtained by teacher-forcing the reference answer after the generated chain, and the efficiency-aware utility is $S(z) = L(z)\, H(z)$ with the efficiency coefficient used in every experiment of this paper ($\alpha = 0.5$, $H \in [0.5, 2.0]$, L_{base} the mean prior trajectory length of the corresponding dataset). We then compute $\mathrm{Cov}_x(L, S)$ *within each question*, over that question's K trajectories, and aggregate over questions: the fraction with $\mathrm{Cov}_x \geq 0$ (Wilson 95% CI) and the median within-question Spearman correlation ρ. Both are Monte-Carlo estimates of the population quantities from the $K = 8$ samples per question. We measure 330 questions (AIME24, MATH-500 and GSM8K) for the untrained prior and for two trained checkpoints of the 1.5B model. Aggregating over samples instead of questions would let question difficulty drive the correlation—easier questions are answered correctly more often *and* more compactly—and would therefore test a different statement than the one required by Proposition 1.

Numerical treatment. Within a question the reference answer y^* is fixed and $L(z) = P_\theta(y^* \mid x, z)$ is the sequence-joint likelihood of the answer tokens, i.e. the quantity we score is the *summed* token log-likelihood. To prevent numerical underflow on the long reasoning chains produced on AIME we apply the standard per-question max-shift in log space, $\log \tilde{L}(z_i) = \log L(z_i) - \max_{1\leq j\leq K} \log L(z_j)$, which is exactly a positive rescaling in probability space, $\tilde{L}(z_i) = c\, L(z_i)$ with $c = 1/\max_j L(z_j) > 0$. Writing $\tilde{S}(z_i) = \tilde{L}(z_i) H(z_i)$, bilinearity of covariance gives $\mathrm{Cov}_x(\tilde{L}, \tilde{S}) = c^2\, \mathrm{Cov}_x(L, S)$, so the shift preserves the sign of the within-question covariance exactly while keeping the values numerically well-conditioned. We deliberately do not replace the joint likelihood by a length-normalised one: averaging the log-likelihood over the answer tokens is the power transform $L \mapsto L^{1/n_{y^*}}$, whose effect on the sign of the covariance is not controlled.

The condition holds wherever the prior's likelihood actually varies. Table 4 and Figure 5 report the measurement. Overall the within-question sample covariance is non-negative for **82.4%** of questions for the untrained prior (95% CI $[78.0, 86.2]$), **85.8%** at step 80 ($[81.6, 89.1]$) and **97.9%**

for the later checkpoint ($[95.7, 99.0]$). The aggregate is, however, a mixture of two very different regimes, and splitting on whether the prior's likelihood varies across samples is far more informative than the aggregate itself.

Why some questions fail, and why the aggregate rises. Within a question the covariance decomposes exactly as

$$\mathrm{Cov}(\tilde{L}, \tilde{S}) = \underbrace{\bar{\eta}\,\mathrm{Var}(\tilde{L})}_{\text{likelihood spread} \geq 0} + \underbrace{\mu\,\mathrm{Cov}(\tilde{L}, \eta)}_{\text{length interaction}} + \underbrace{\mathbb{E}\big[(\tilde{L} - \mu)^2(\eta - \bar{\eta})\big]}_{\text{third moment}}, \tag{26}$$

with $\tilde{L} = L / \max_i L_i$, $\eta = H(z)$, $\bar{\eta} = \mathbb{E}[\eta]$, $\mu = \mathbb{E}[\tilde{L}]$ and $\tilde{S} = \tilde{L}\eta$. Since $\mathrm{Cov}(\tilde{L}, \tilde{S}) = c^2\,\mathrm{Cov}(L, S)$ with $c > 0$, the decomposition characterises the sign of the original covariance. The first term is non-negative; measured on our samples it dominates the other two by more than an order of magnitude (0.122 versus 0.004 and -0.003 for the untrained prior). The condition is therefore satisfied whenever the correctness likelihood has something to spread over—and it has not, precisely on the questions the prior has already mastered. On those, the $K = 8$ trajectories reach the reference answer with a standard deviation of $\log L$ below 0.1 (accuracy ≥ 0.95, chains of ≈ 650–960 tokens), so L is effectively constant, the covariance is decided by the length term acting on numerically negligible differences, and its sign is a coin flip: these questions satisfy the condition about half the time, with a mean covariance of ≈ 0.001.

Conditioning on the remaining questions, the condition holds for **100% of questions in all three states** (Wilson lower bounds 98.3%, 98.4% and 98.8%; last column of Table 4), the mean within-question covariance grows from 0.15 to 0.23, and *not one* failure lies outside the near-degenerate set. The rise of the aggregate figure with training is accordingly arithmetic rather than a change in the guarantee: the share of near-degenerate questions falls from 33% to 5%, so the mixture $0.667 \cdot 100\% + 0.333 \cdot 47.3\% = 82.4\%$ becomes $0.955 \cdot 100\% + 0.045 \cdot 53.3\% = 97.9\%$. We report these questions explicitly rather than excluding them, because they are also the regime in which the posterior has nothing left to reallocate and the guarantee matters least.

The empirical boundary in the efficiency exponent. The boundary discussion above concerns the strength of the length decay, which our objective controls through the single exponent α in $H(z)$. Because H depends on α only through $(L_{\text{base}}/|z|)^\alpha$, the condition can be re-evaluated for *any* exponent on trajectories that have already been sampled, at no additional sampling cost. Figure 6 sweeps $\alpha \in \{0, 0.25, 0.5, 0.75, 1, 1.5, 2, 3\}$, with and without the clamp, i.e. it traces how much length decay the posterior advantage tolerates. Two effects are visible. Without the clamp the condition degrades monotonically in α (from 82.4% at $\alpha = 0.5$ down to 73.6% at $\alpha = 3$ for the untrained prior), which is precisely the failure predicted in the boundary discussion. With the clamp used in this paper the same sweep is flat or improving (85.8% up to 96.4%), because bounding $H(z) \in [0.5, 2.0]$ prevents a handful of very short trajectories from dominating the utility. (The sweep evaluates all questions with a single pooled L_{base}, the batch-average convention of Algorithm 1, whereas Table 4 uses each dataset's own mean length; this is why the pooled $\alpha = 0.5$ value exceeds the per-dataset aggregate reported above.) The clamp is therefore not merely a numerical safeguard: it is what keeps the condition—and with it the guarantee of Proposition 1—valid at aggressive length decay. All experiments in this paper use $\alpha = 0.5$, which sits well inside the region where the condition holds even unclamped; this is consistent with the hyperparameter study in Figure 3, where larger α traded accuracy away for only marginal token reductions.

C Utility Lower Bound and Posterior-to-Prior Transfer

In this appendix, we provide a detailed derivation of the ELBO proposed in Section 3.3 of the main text.

Original Objective Our objective is to maximize the logarithmic expected utility under the prior distribution $\pi_\theta(z|x)$:

$$\log J(\theta) = \log \mathbb{E}_{z \sim \pi_\theta(\cdot|x)}[S(z)]. \tag{27}$$

where the comprehensive utility is $S(z) = P_\theta(y^*|x, z) \cdot H(z)$.

Table 4: Within-question verification of the covariance condition at the operating point $\alpha = 0.5$, for 330 evaluation questions and $K = 8$ prior samples per question, using each dataset's mean prior trajectory length as L_{base}; the aggregate rows pool the per-question values. "% (non-deg.)" conditions on questions whose prior likelihood is not near-degenerate (std. of $\log L \geq 0.1$ across the K samples).

Training state	Dataset	#Q	$\text{Cov}_\pi(L, S) \geq 0$ (%)	95% CI	median ρ	% (non-deg.)
Init	AIME24	30	100.0	[88.6, 100.0]	0.98	100
	MATH-500	100	90.0	[82.6, 94.5]	0.80	100
	GSM8K	200	76.0	[69.6, 81.4]	0.54	100
	All	330	**82.4**	[78.0, 86.2]	0.69	100
Trained (ckpt 80)	AIME24	30	100.0	[88.6, 100.0]	0.95	100
	MATH-500	100	91.0	[83.8, 95.2]	0.87	100
	GSM8K	200	81.0	[75.0, 85.8]	0.71	100
	All	330	**85.8**	[81.6, 89.1]	0.79	100
Trained (ckpt 200)	AIME24	30	100.0	[88.6, 100.0]	0.98	100
	MATH-500	100	99.0	[94.6, 99.8]	0.98	100
	GSM8K	200	97.0	[93.6, 98.6]	0.86	100
	All	330	**97.9**	[95.7, 99.0]	0.93	100

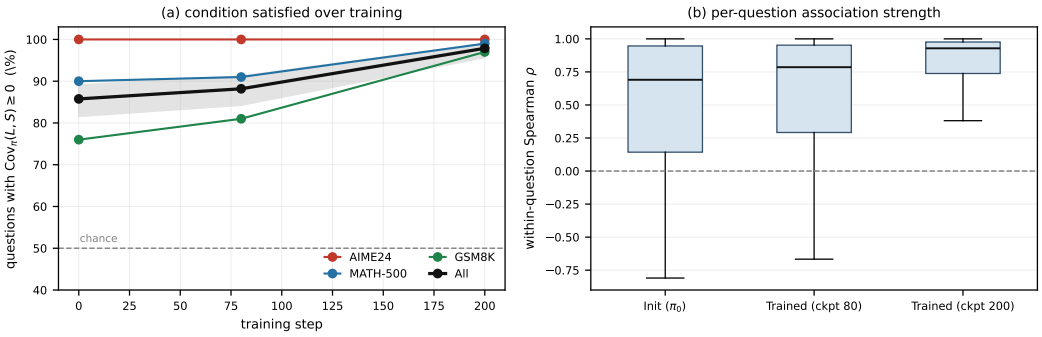

Figure 5: The covariance condition measured per question. **(a)** Share of evaluation questions whose within-question covariance $\text{Cov}_x(L, S)$ is non-negative, with Wilson 95% intervals; the dashed line is chance. **(b)** Distribution of the within-question Spearman correlation between L and S. The aggregate curve of panel (a) uses a single pooled L_{base} (the batch-average convention of Algorithm 1), the per-dataset curves use each dataset's own mean length, and the aggregate rows of Table 4 pool the per-question values of the latter.

Importance Sampling Transformation Because high-utility reasoning trajectories are sparse under the question-only prior π_θ, directly optimizing the aforementioned objective can be highly sample-inefficient, particularly during the early stages of training. We therefore introduce the answer-conditioned posterior distribution $q_\theta(z|x, y^*)$ and utilize the importance sampling identity to transform the objective function:

Step 1: Expectation Expansion According to the definition of expectation, the objective function is expanded into a summation form over the discrete space $\mathcal{Z}$:

$$\log J(\theta) = \log \sum_{z \in \mathcal{Z}} \pi_\theta(z|x) \cdot S(z). \tag{28}$$

Step 2: Identity Insertion and Importance Sampling Multiply and divide by the posterior distribution $\pi_\theta(z|x, y^*)$ (abbreviated as $q(z)$) inside the summation:

$$\log J(\theta) = \log \sum_{z \in \mathcal{Z}} q(z) \frac{\pi_\theta(z|x)}{q(z)} \cdot S(z). \tag{29}$$

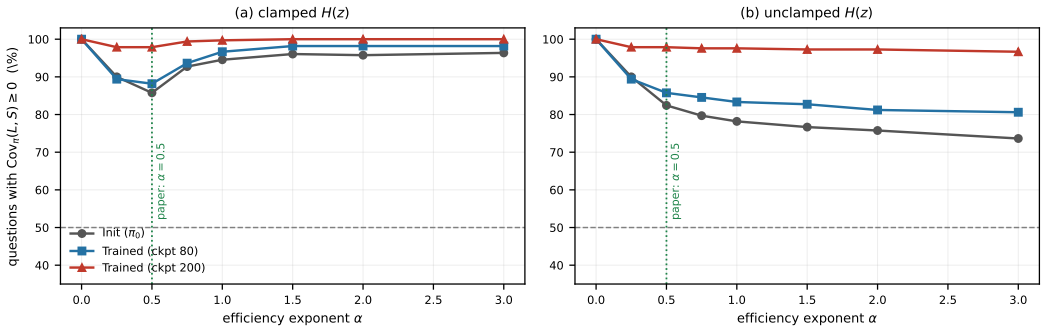

Figure 6: Empirical boundary of the covariance condition in the efficiency exponent α, re-evaluated on the trajectories already sampled for each state, **(a)** with and **(b)** without the clamp on $H(z)$. The dotted line marks $\alpha = 0.5$, the value used in all experiments of this paper. Without the clamp the condition degrades monotonically in α, as predicted by the boundary analysis; the clamp keeps it intact.

At this point, the sampling distribution has shifted from the prior π_θ to the posterior $q(z)$, which can be rewritten as an expectation under the posterior:

$$\log J(\theta) = \log \mathbb{E}_{z\sim q(z)} \left[\frac{\pi_\theta(z|x)}{q(z)} \cdot S(z) \right]. \tag{30}$$

Step 3: Applying Jensen's Inequality Since the logarithmic function $\log(\cdot)$ is a strictly concave function, according to Jensen's inequality $\log(\mathbb{E}[X]) \geq \mathbb{E}[\log(X)]$, we can move the logarithmic operation inside the expectation, we obtain the following lower bound on the objective:

$$\log J(\theta) \geq \mathbb{E}_{z\sim q(z)} \left[\log \left(\frac{\pi_\theta(z|x)}{q(z)} \cdot S(z) \right) \right]. \tag{31}$$

Step 4: Log Expansion & Rearrangement Using the logarithmic property $\log(abc/d) = \log a + \log b + \log c - \log d$, and expanding $S(z)$ into $P_\theta(y^*|x, z) \cdot H(z)$:

$$\begin{aligned} \text{ELBO} &= \mathbb{E}_{z\sim q(z)} \left[\log \frac{P_\theta(y^*|x, z) \cdot H(z) \cdot \pi_\theta(z|x)}{q(z)} \right] && (32) \\ &= \mathbb{E}_{z\sim q(z)} \left[\log P_\theta(y^*|x, z) + \log H(z) + \log \pi_\theta(z|x) - \log q(z) \right] && (33) \\ &= \mathbb{E}_{z\sim q(z)} \left[\log P_\theta(y^*|x, z) + \log H(z) \right] - \mathbb{E}_{z\sim q(z)} \left[\log q(z) - \log \pi_\theta(z|x) \right]. && (34) \end{aligned}$$

Step 5: Final Formula Utilizing the definition of KL divergence $D_{\mathrm{KL}}(q||\pi) = \mathbb{E}_q[\log q - \log \pi]$, we obtain the final optimization objective:

$$\text{ELBO} = \mathbb{E}_{z\sim q(\cdot|x,y^*)} \left[\log P_\theta(y^*|x, z) + \log H(z) \right] - D_{\mathrm{KL}}(q(z)||\pi_\theta(z|x)). \tag{35}$$

D Prompts

D.1 Implementation Details of VPG-EA Prompts

Student Prompt (Prior)

<|im_start|>system
A conversation between user and assistant. The user asks a question, and the assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process is enclosed within <think></think> tags, i.e., <think>This is my reasoning.</think> This is my answer.<|im_end|>

<|im_start|>user
{Question}<|im_end|>
<|im_start|>assistant

Teacher Prompt (Posterior)

<|im_start|>system
A conversation between user and assistant. The user asks a question, and the assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process is enclosed within <think></think> tags, i.e., <think>This is my reasoning.</think> This is my answer.<|im_end|>
<|im_start|>user
Given the following question and its reference answer, your task is to produce a step-by-step explanation that logically leads to the reference answer, written in the style of a first-person think-aloud monologue. You are encouraged to draw on the reference answer for internal guidance to help structure and support your reasoning, but the final monologue must read as a genuine, first-encounter, real-time discovery, without mentioning or implying any prior access to the reference answer.
Question: {Question}
Reference Answer: {Reference Answer} 1. Output only the first-person, think-aloud monologue. Do not include any preface, summary, or restatement of these instructions.
2. Maintain the tone of a focused individual thinking to themself. Avoid meta-commentary like "for the first time," and any phrasing that reveals simulation.
3. Do not mention, imply, or hint at prior access to the Reference Answer in the monologue. Avoid phrases like "according to the answer..." or "to get to that answer...", and any euphemism that signals foreknowledge.
4. Do not merely restate the final answer in the monologue; articulate the reasoning pathway with sufficient intermediate steps, rationale, decision points, verification, and any necessary error-correction or backtracking.<|im_end|> <|im_start|>assistant

D.2 Exploration of the Endogenous Attribution of Efficiency after Removing Instruction Bias

Student Prompt (Prior)

<|im_start|>system
You are a student solving a problem. You must show your work.
First, provide your step-by-step reasoning process enclosed in <think></think> tags.
Second, provide your final answer enclosed in \boxed{}.
<|im_end|>
<|im_start|>user
{Question}
<|im_end|>
<|im_start|>assistant

Teacher Prompt (Posterior)

<|im_start|>system
You are an expert teacher providing a model solution. You are given a question and its correct answer.
Your task is to generate a **concise, direct, and efficient** step-by-step solution that logically leads *directly* to the given answer.
Avoid any unnecessary exploration, self-correction, or reflective 'think-aloud' monologues. Focus *only* on the core, necessary logical steps.
The reasoning process must be enclosed in <think></think> tags. The final answer must be enclosed in \boxed{}.
<|im_end|>
<|im_start|>user
Question: {Question}

Answer: {Answer}
<|im_end|>
<|im_start|>assistant

When evaluating the efficient reasoning performance of the VPG-EA framework across multiple benchmarks, a core question requires exploration: does the reduction in model output length originate from the RL algorithm successfully locating and learning sparse, highly efficient samples, or is it merely because the model is fitting a certain instruction bias that implies a reduction in output? To eliminate the interference caused by such instructions, we designed a set of extreme comparative prompt experiments to verify the true source of the model's efficiency after training with the VPG-EA framework. Our original instructions (see D.1) not only do not require the model to remain concise, but rather explicitly encourage trial-and-error and exploration. We require the model to elaborate on the reasoning path in detail, including sufficient intermediate steps, underlying principles, decision points, verification processes, and any necessary error-correction or backtracking. As a baseline for the comparative prompts (see D.2), we constructed an extremely rigid efficiency instruction, requiring the model to generate a concise, direct, and efficient step-by-step solution, avoiding any unnecessary exploration, self-correction, or reflectiveness.

The experimental results, as shown in Table 5, indicate that the Instruction-Based Prompts achieve a shorter average output length on GSM8K and AIME24; however, this length advantage is accompanied by significant performance degradation: the accuracy on AIME24 plummets from 33.33% to 26.67%, and there are also varying degrees of decline on GSM8K and MATH-500. This demonstrates that while rigid length-constrained instructions can mechanically truncate the output, the cost is the destruction of the necessary computational depth required for the model to process complex reasoning tasks. In contrast, VPG-EA achieves a comprehensive lead in accuracy while maintaining a similar length level, proving that its length optimization is not a simple shortening, but a selective refinement. The aforementioned experiments powerfully prove that the effectiveness of VPG-EA does not rely on fitting efficient instructions. In our framework, even if the system instructions explicitly encourage divergence and trial-and-error, the model still achieves significant refinement in the final output. Therefore, the VPG-EA framework genuinely empowers the large model to find a logically rigorous and redundancy-free efficient manifold; this is a cognitive upgrade at the algorithmic level, rather than remaining at the level of superficial compliance to instructions.

Clarification on Evaluation Protocol and Data Leakage Prevention To address potential concerns regarding data integrity, we explicitly clarify the usage of the Answer field in the prompts. While the Posterior template in Appendix D.2 includes a reference answer field to facilitate the construction of the posterior distribution during the training phase, this field was strictly omitted during the evaluation of all benchmarks (GSM8K, MATH-500, AIME, etc.). During inference, the model was provided only with the Question and the system instructions to ensure a zero-leakage, fair comparison with all baseline models. The performance gains observed in VPG-EA stem from the internalized efficient reasoning manifold learned during posterior-to-prior transfer, rather than any form of ground-truth injection during testing.

Table 5: Performance comparison between instruction-based efficiency prompts and the VPG-EA framework.

	GSM8K			MATH-500			AIME24		
Method	ACC ↑	A.Tok ↓	ε^3 ↑	ACC ↑	A.Tok ↓	ε^3 ↑	ACC ↑	A.Tok ↓	ε^3 ↑
VPG-EA	**80.89**	519.94	**12.58**	**82.40**	**1891.80**	**3.59**	**33.33**	6659.73	**0.17**
Instruction-Based Prompts	78.46	**516.16**	11.93	81.60	1965.03	3.39	26.67	**5708.50**	0.12

E Algorithmic Procedure

As outlined in Algorithm 1, to align the theoretical derivation with practical RL training, we introduce the following key processes in our engineering implementation:

First, to ensure numerical stability during training, we apply a clamp operation to the efficiency coefficient $H(z)$. In practical sampling, the model occasionally generates abnormally short outputs,

Algorithm 1 VPG-EA Training Procedure

Input: Dataset $\mathcal{D} = \{(x, y^*)\}$, Initial Policy π_θ, transfer coefficient β, Generation batch size G
Output: Optimized policy parameters θ^*
Initialize policy parameters θ
while not converged **do**
  Sample a batch of question-answer pairs $(x, y^*) \sim \mathcal{D}$
  // Phase 1: Posterior-Guided Exploration
  Sample G prior paths: $\mathcal{Z}_{\text{prior}} = \{z_{\text{prior}}^{(1)}, \ldots, z_{\text{prior}}^{(G)}\} \sim \pi_\theta(\cdot \mid x)$
  Sample G posterior paths: $\mathcal{Z}_{\text{post}} = \{z_{\text{post}}^{(1)}, \ldots, z_{\text{post}}^{(G)}\} \sim q_\theta(\cdot \mid x, y^*)$
  Compute average prior length: $L_{base} \leftarrow \frac{1}{G} \sum_{j=1}^{G} |z_{prior}^{(j)}|$
  // Phase 2: Cross-View Validation & Efficiency-Aware Utility
  Compute prior baseline: $\bar{U}_{\text{prior}} \leftarrow \frac{1}{G} \sum_{j=1}^{G} \log P_\theta(y^* \mid x, z_{\text{prior}}^{(j)})$
  for $i = 1, \ldots, G$ **do**
    $U_{\text{post}}^{(i)} \leftarrow \log P_\theta(y^* \mid x, z_{\text{post}}^{(i)})$
    $R_{\text{correct}}^{(i)} \leftarrow \max\left(0, U_{\text{post}}^{(i)} - \bar{U}_{\text{prior}}\right)$
    $H_i \leftarrow \text{Clamp}\left(\left(\frac{L_{\text{base}}}{L_{z_{\text{post}}^{(i)}}}\right)^{\alpha}, H_{\min}, H_{\max}\right)$
    $\hat{S}_i \leftarrow R_{\text{correct}}^{(i)} \cdot H_i$
  end for
  Compute relative advantages $A_i \leftarrow \frac{\hat{S}_i - \mu_s}{\sigma_s + \epsilon}$ for $i \in \{1, \ldots, G\}$
  // Phase 3: Advantage-Gated Posterior-to-Prior Transfer
  $\mathcal{L}_{\text{post_pg}} \leftarrow -\frac{1}{G} \sum_{i=1}^{G} A_i \log q_\theta(z_{\text{post}}^{(i)} \mid x, y^*)$
  $\mathcal{L}_{\text{transfer}} \leftarrow \frac{1}{G} \sum_{i=1}^{G} \mathbb{I}(A_i > 0) \left[\text{sg}(\log q_\theta(z_{\text{post}}^{(i)} \mid x, y^*)) - \log \pi_\theta(z_{\text{post}}^{(i)} \mid x)\right]$
  $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{post_pg}} + \beta \mathcal{L}_{\text{transfer}}$
  Update parameters: $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_{\text{total}}$
end while
return θ^*

which causes a surge in the ratio $L_{base}/L_{z_{post}}$ and consequently triggers reward explosion. The introduction of the clamp mechanism effectively prevents such abnormal samples from dominating the gradient update, ensuring that the model does not deviate from the fundamental optimization of logical correctness while pursuing output conciseness. Specifically, in our main experiments, we explicitly set the bounds to $H_{min} = 0.5$ and $H_{max} = 2.0$ to balance efficiency incentives and training stability.

Second, during posterior-to-prior transfer, we apply a stop-gradient (sg) operation to the posterior distribution $q(z)$. This blocks the posterior branch within the explicit transfer loss, so the term acts on the prior distribution π_θ alone. Because the two streams share parameters, this does not imply globally unidirectional parameter updates: the posterior policy-gradient term also modifies the shared parameters and therefore affects both views in subsequent iterations.

Finally, regarding the coefficient β of the transfer term, we set $\beta = 1$ in the main experiments as the reference weighting of the explicit transfer surrogate; it is treated as a control coefficient rather than as a value uniquely determined by the lower bound. This parameter is retained in the algorithm procedure primarily to serve as a control variable for ablation studies (setting $\beta = 0$ in the ablation variant that removes the explicit transfer term).

F Performance across Different Model Architectures

To verify the cross-architecture generalization capability of VPG-EA, we conduct experiments on DeepSeek-R1-Distill-Llama-8B and compared it with advanced baselines at this parameter scale: Manifold Steering (Huang et al., 2025) By employing the mean difference method and PCA analysis, this approach extracts and purifies the low-dimensional manifold direction representing overthinking from the model's activation space, and subsequently applies ablation interventions to the residual stream activations at each layer, thereby mitigating redundant reasoning internally within the model. ACPO (Cheng et al., 2025) This method introduces explicit system-aware tokens (e.g., <fast_think>) for SFT cold start, and subsequently integrates an online token length budget mechanism and a composite reward function within the GRPO framework, enabling the model to achieve dynamic switching between fast and slow thinking modes and adaptive allocation of cognitive resources based on task difficulty. The results are shown in Table 6. Except for Base and VPG-EA, which are locally evaluated, the data for the remaining baselines are cited from their original papers.

The experimental results indicate that VPG-EA also achieves a good balance between efficiency and accuracy on Llama-8B. On GSM8K, VPG-EA improves the accuracy of the base model by 1.29%, and the average token consumption decreases by 199.18, representing a reduction of 22.71%; although Manifold Steering has lower token consumption (542), its accuracy significantly degrades. On MATH-500, VPG-EA optimizes the average token consumption from 3292.8 to 2536.2, with a simultaneous improvement in accuracy, yielding an overall performance superior to ACPO and comparable to R1-VeriThinker-8B. On AIME24, the base model consumes 12,977.53 tokens to achieve 33.33% accuracy, whereas VPG-EA raises accuracy to 46.70% while using 4,752 fewer tokens (36.6%). Although some baselines report higher accuracy, VPG-EA achieves better computational efficiency while ensuring relatively high accuracy, indicating that its core mechanism can guide the model to autonomously internalize the efficient reasoning manifold, realizing a dual improvement in correctness and efficiency without relying on specific architectures.

Table 6: Cross-architecture generalization experimental results of VPG-EA.

Model	GSM8K			MATH-500			AIME24			Average		
	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑	ACC↑	A.Tok↓	ε^3 ↑
Original Model	87.87	877	8.80	87.40	3292.80	2.32	33.33	12977.53	0.09	69.53	5715.78	3.74
Manifold Steering*	82.80	**542**	**12.65**	88.00	2873.00	2.70	**50.00**	9457.00	0.26	73.60	4290.67	**5.20**
R1-VeriThinker*	-	-	-	**89.90**	2953.00	2.74	46.90	11285.00	0.19	-	-	-
ACPO*	-	-	-	87.00	**2232.00**	**3.39**	43.30	**7405.00**	0.25	-	-	-
VPG-EA	**89.16**	677.82	11.73	89.60	2536.20	3.17	46.70	8225.07	**0.27**	**75.14**	**3813.03**	5.05

Note: * denotes results cited directly from their original papers, as their model checkpoints are not publicly available. To ensure statistical fairness, average performance metrics are computed exclusively for models evaluated across all benchmarks. We omit cross-task aggregate comparisons for baselines with incomplete data.

G EXPERIMENTAL DETAILS

Hardware and Software Platforms The training infrastructure is based on a single-node configuration, equipped with four NVIDIA RTX Pro 6000 Blackwell (96GB) high-performance GPUs with a unified CUDA version of 12.4. The system runs on Ubuntu 22.04, utilizing a Python environment (version 3.12.3) and PyTorch 2.5.1. This study adopts the vLLM (0.7.3) framework for efficient batch decoding and employs the DeepSpeed framework during the training phase to optimize memory.

Training Configurations The experiment utilizes the AdamW optimizer, with the learning rate set to 1e-6 and a warmup ratio of 0.2. The global effective batch size is 64, and the sampling temperature is set to 1.0, paired with 4 prior and 4 posterior sampling examples respectively. Differentiated parallel strategies are applied based on model scales: the 1.5B model uses DeepSpeed ZeRO-1, while the 7B and 8B models enable DeepSpeed ZeRO-3 to optimize memory distribution. Additionally, the hyperparameters α and β are fixed at 0.5 and 1.0, respectively, and the efficiency clamp bounds $[H_{min}, H_{max}]$ are set to $[0.5, 2.0]$ (as detailed in Appendix E). Regarding the computational cost, a single complete training run for the 1.5B model on 4 $\times$ NVIDIA RTX Pro 6000 GPUs takes approximately 13.6 hours.

Evaluation Configurations The vLLM framework is adopted to accelerate inference, with the temperature set to 0.6, Top-p sampling set to 0.95, and the maximum output length set to 16K tokens.

Choice of the efficiency metric We adopt $\varepsilon^3 = \mathrm{ACC}^2/\mathrm{A.Tok}$, the Efficiency-Aware Effectiveness score introduced by Lin et al. (2026a), which measures reasoning quality and computational cost jointly rather than as two separate axes. The quadratic accuracy term is a deliberate design choice of that metric: relative to the conventional accuracy-per-token ratio $\mathrm{ACC}/\mathrm{A.Tok}$, it weights correctness more heavily and thereby discourages degenerate strategies that shorten outputs at the expense of accuracy. This weighting matches the objective of this work, in which reasoning length must be reduced while accuracy is *preserved*: because a one-point loss of accuracy is penalised roughly twice as strongly as an equivalent relative increase in length, the metric is a conservative choice for our claims rather than one that flatters them. We nevertheless treat the functional form as a design choice rather than a ground truth, and we therefore (i) report ACC and A.Tok for every configuration (Tables 1 and 2), (ii) test accuracy, output length and ε^3 separately in Appendix K, where accuracy is statistically non-inferior to the strongest baselines while output length is 7.9–37.3% shorter, and (iii) show in Appendix K.3 that most of the ε^3 gain is attributable to the brevity channel rather than to the accuracy channel. As a final check of metric dependence, we recomputed the aggregate ranking under the conventional $\mathrm{ACC}/\mathrm{A.Tok}$ ratio on the same GSM8K+MATH-500 aggregate: VPG-EA remains first at both scales (0.0678 vs. 0.0651 for the strongest 1.5B baseline, and 0.0855 vs. 0.0785 at 7B), so our conclusions do not hinge on the quadratic weighting.

Baseline Introduction

- AdaptThink (Zhang et al., 2025): This method utilizes RL to endow the model with dynamic decision-making capabilities, enabling it to skip lengthy reasoning processes for simple problems and generate answers directly.
- Laser (Liu et al., 2026): This method introduces a length-based reward shaping mechanism, dynamically penalizing excessively long chains of thought during RL training to guide the model to shorten the reasoning length while maintaining accuracy.
- AutoThink (Tu et al., 2025): This method adopts a multi-stage RL framework, aiming to balance computational overhead and performance on problems of varying difficulty through adaptive reasoning strategies.
- JET (Han et al., 2026): Based on the DAPO RL framework, this method combines trajectory rollout generation and a progressive early-stopping truncation strategy to construct short reasoning trajectories, and designs a composite reward mechanism to drive training, allowing the model to actively terminate unnecessary reasoning processes.
- Thinkless (Fang et al., 2025): This method constructs long-short response pairs through knowledge distillation and decouples the optimization process of reasoning mode selection and answer generation.

- R1-VeriThinker (Chen et al., 2025): This method proposes a verification-based pruning strategy, utilizing an auxiliary verifier to identify and remove redundant steps in the reasoning path.

H MICRO-BEHAVIORAL ANALYSIS OF REASONING TRAJECTORY

To decode how VPG-EA reshapes model behavior, we follow the taxonomy from (Hou et al., 2026; Sui et al., 2025) and categorize reasoning tokens into three groups using a rule-based keyword dictionary (see Table 7): (i) Soliloquize & Thinking, (ii) Check & Confirm, and (iii) Summary & Calculation. Figure 7 reveals that while both VPG-EA and Laser effectively suppress tokens associated with the overthinking phenomenon (Soliloquize & Thinking), their reallocation strategies diverge. Laser maintains logical coherence through verbose Summary & Calculation steps. In contrast, VPG-EA aggressively compresses redundant summaries and selectively reallocates computational budget to the Check & Confirm phase. This design aligns with established findings that explicit procedural verification is essential for rigorous reasoning (Weng et al., 2023; Dhuliawala et al., 2024). VPG-EA outperforms Laser because it strengthens logical verification at key nodes while compressing non-essential information. This enables VPG-EA to produce more rigorous outputs through leaner reasoning trajectories.

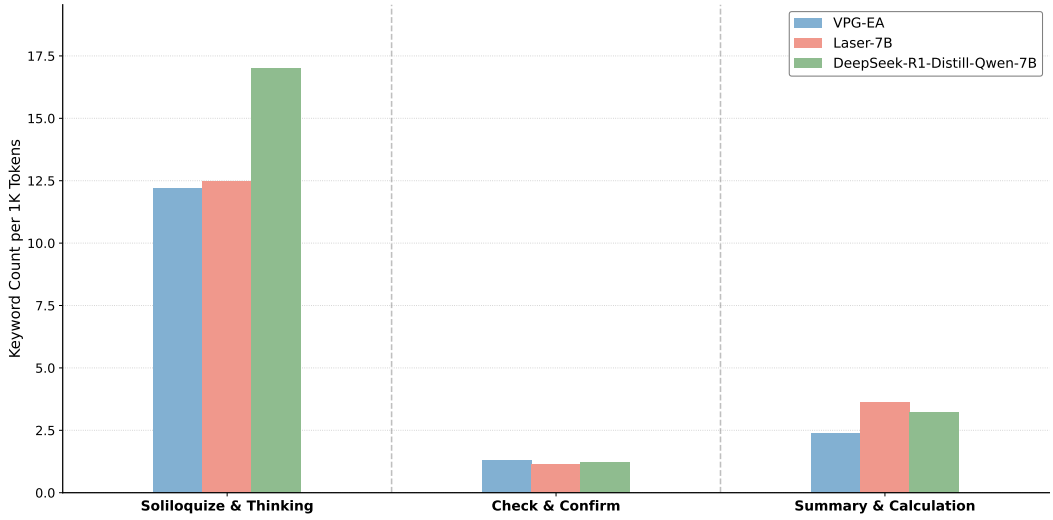

Figure 7: Frequency distribution of inference keywords per thousand tokens for each model under the MATH-500 benchmark.

The conceptual taxonomy of categorizing reasoning behavior into distinct cognitive phases is inspired by established frameworks in existing literature (Hou et al., 2026; Sui et al., 2025). To operationalize this taxonomy for the micro-behavioral analysis above, we developed a rule-based dictionary of explicit transitional phrases. These keywords were utilized to match and quantify the corresponding tokens generated within the `<think>...</think>` trajectories. The complete mapping between the behavior categories and their respective keywords is detailed in Table 7.

Table 7: Keyword mapping for reasoning behavior categorization.

Behavior Category	**Keywords**
Soliloquize & Thinking	`Wait`, `But`, `wait`, `Hold on`, `Alternatively`
Check & Confirm	`Let me confirm`, `Double-check`
Summary & Calculation	`Remember`, `Let me compute`, `Therefore`

I Dynamic Analysis of Reasoning Trajectory Length and Posterior-to-Prior Transfer

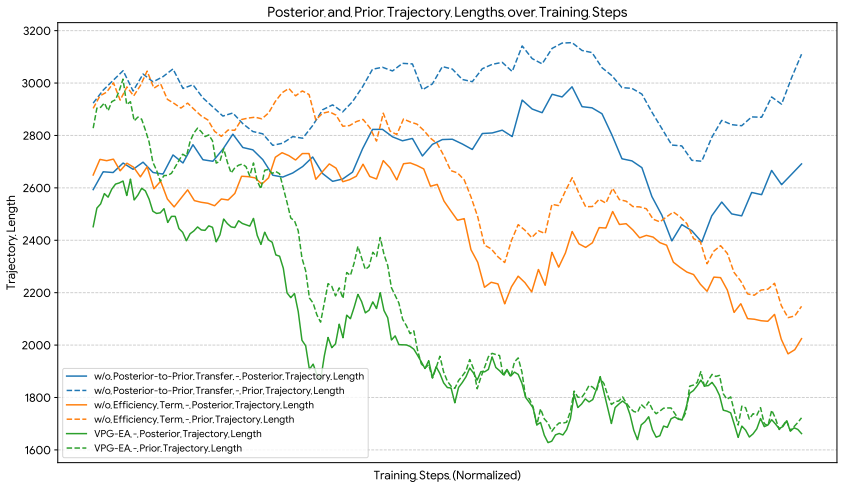

Figure 8: Comparison of training trajectory lengths between VPG-EA and its ablation variants (1.5B model).

Figure 8 tracks the prior and posterior reasoning trajectory lengths during the 1.5B model's training to analyze each module's impact. Removing the explicit posterior-to-prior transfer term substantially weakens the reduction of the prior trajectory length, indicating that the explicit transfer mechanism contributes to efficient adaptation of the prior. Conversely, retaining the transfer term but removing the efficiency constraint allows the prior trajectory to decrease and approach the posterior, but the convergence is slow and unstable. Under the complete VPG-EA framework, both trajectories drop sharply and jointly converge. As a bottom-up RL paradigm, VPG-EA utilizes posterior-to-prior transfer to provide additional guidance and overcome inefficient exploration, while the efficiency constraint accelerates convergence. In late training stages, the prior trajectory drops below the posterior, which is consistent with the intended role of the explicit transfer mechanism in moving efficient reasoning patterns toward the inference-time prior.

J Trade-off Analysis Under Difficulty Grading

Figure 9 evaluates model performance across the five difficulty levels of MATH-500. On levels 1 to 4, VPG-EA maintains high accuracy with shorter CoT outputs. At the highest difficulty, VPG-EA achieves a 5% higher accuracy than Laser for a marginal cost of roughly 100 tokens, whereas Laser's accuracy degrades below that of the base model. Notably, other same-paradigm baselines suffer from both longer CoT lengths and lower accuracies on hard samples. This demonstrates that VPG-EA is highly robust to difficulty variations, successfully decoupling reasoning length from accuracy without sacrificing performance on complex problems.

K Significance Analysis

All claims in the main experiments (Table 1) are supported by paired significance tests. This appendix specifies the testing protocol (Appendix K.1), reports aggregate tests over GSM8K and MATH-500 for both model scales (Appendix K.2), decomposes the improvement of ε^3 into an accuracy and a brevity channel (Appendix K.3), tests accuracy non-inferiority against the strongest baselines (Appendix K.4), analyses the statistical power limits of AIME (Appendix K.5), and lists threats to validity (Appendix K.6).

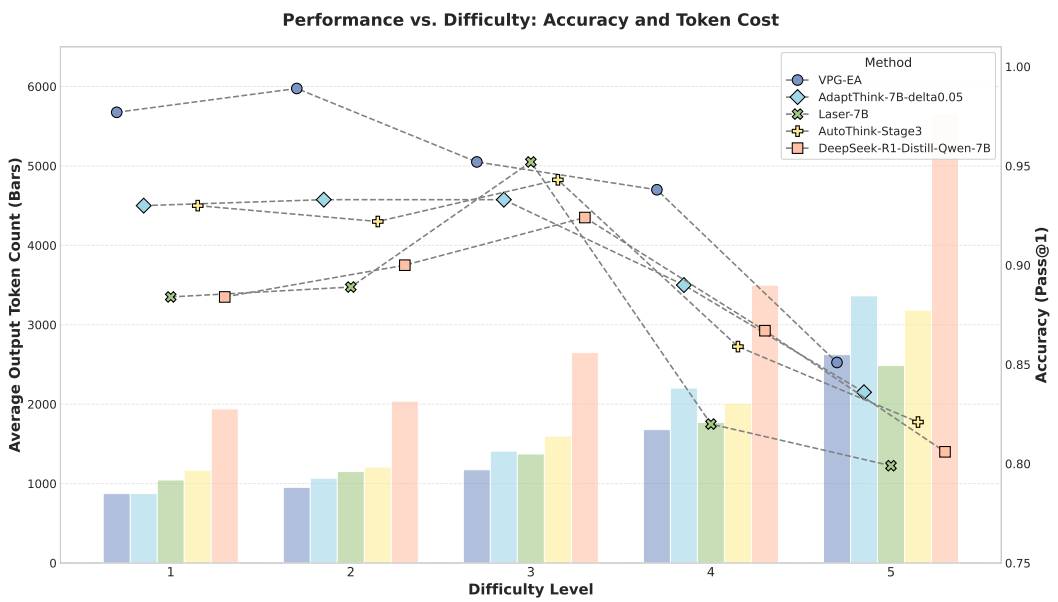

Figure 9: 7B baseline performance across MATH-500 difficulty levels (1-5). Line: Accuracy; Bar: average output length.

K.1 Protocol and Tests

Paired evaluation. Every comparison in Table 1 uses the identical question set for both models, with identical decoding (temperature 0.6, top-p 0.95, maximum output length 16,384 tokens, vLLM). We verified that the stored per-question records of all runs share the same question ordering, so tests can be paired item-wise.

Accuracy: exact McNemar test. Accuracy is binary per question, and the two models are evaluated on the same items; we therefore use McNemar's exact test on the discordant pairs (n_{10}, n_{01}), i.e., the two-sided binomial tail with $p = \frac{1}{2}$. This is the appropriate test for paired binary outcomes and requires no asymptotic approximation.

Tokens and ε^3: paired bootstrap. Output length and $\varepsilon^3 = \mathrm{ACC}^2/\mathrm{A.Tok}$ are continuous quantities with large per-question variance. We report 95% percentile intervals and two-sided p-values from a paired question-level bootstrap with $B = 10{,}000$ resamples: questions are drawn with replacement, and the statistic is recomputed for both models on the same resampled items.

Aggregates. Aggregated results use the same convention as the "Average" columns of Table 1, i.e. an unweighted mean over benchmarks; the aggregates tested here are taken over GSM8K and MATH-500 only, because AIME does not have sufficient sample size for inferential claims (Appendix K.5). For aggregate tests we use a *stratified* bootstrap that resamples questions within each benchmark and then averages across benchmarks, so that a change in test-set composition is propagated through the aggregate.

Multiplicity. The comparison against the per-scale strongest baseline is the pre-specified primary comparison and is reported unadjusted; comparisons against the remaining baselines are secondary. Applying Holm's step-down correction across the six comparators within each scale leaves all twelve $\Delta\varepsilon^3$ results significant at $\alpha = 0.05$ (the least significant comparison has raw $p = 0.021$).

Reported quantities. We report effect sizes together with uncertainty: ΔACC in percentage points, ΔA.Tok in percent, and $\Delta\varepsilon^3$ in absolute units, each with a 95% confidence interval.

Table 8: Paired significance analysis at the 1.5B scale, aggregated over GSM8K and MATH-500 (macro-average over the two benchmarks). Δ is VPG-EA minus the comparator. Accuracy is tested with McNemar's exact test; tokens and ε^3 with a 10,000-sample paired question-level bootstrap (95% percentile intervals). † marks the strongest baseline at this scale. $^*p < 0.05, ^{**}p < 0.01$, $^{***}p < 0.001$.

	ΔACC (pp)		ΔA.Tok (%)		$\Delta\varepsilon^3$	
Comparator	Δ [95% CI]	p	Δ [95% CI]	p	Δ [95% CI]	p
JET†	$+0.76\,[-1.39, +2.92]$	0.493	$-16.9\,[-22.0, -11.5]$	$< 0.001^{***}$	$+0.61\,[+0.10, +1.14]$	0.021^{*}
Thinkless	$-0.79\,[-3.08, +1.42]$	0.497	$-6.6\,[-11.8, -0.9]$	0.022^{*}	$+0.96\,[+0.47, +1.46]$	$< 0.001^{***}$
AutoThink	$+0.27\,[-2.01, +2.61]$	0.811	$-3.5\,[-9.3, +2.5]$	0.239	$+1.05\,[+0.60, +1.51]$	$< 0.001^{***}$
AdaptThink	$+1.72\,[-0.60, +4.02]$	0.145	$-6.2\,[-15.2, +2.8]$	0.182	$+1.20\,[+0.69, +1.71]$	$< 0.001^{***}$
Laser	$+1.28\,[-0.89, +3.48]$	0.259	$-6.5\,[-12.1, -0.7]$	0.029^{*}	$+1.65\,[+1.18, +2.13]$	$< 0.001^{***}$
Base model	$+5.53\,[+3.20, +7.91]$	$< 0.001^{***}$	$-37.3\,[-41.9, -32.6]$	$< 0.001^{***}$	$+3.24\,[+2.77, +3.73]$	$< 0.001^{***}$

Aggregated over $n = 1{,}819$ evaluation questions. Positive ΔACC and $\Delta\varepsilon^3$, negative ΔA.Tok favour VPG-EA.

Table 9: Paired significance analysis at the 7B scale, aggregated over GSM8K and MATH-500. Format as in Table 8.

	ΔACC (pp)		ΔA.Tok (%)		$\Delta\varepsilon^3$	
Comparator	Δ [95% CI]	p	Δ [95% CI]	p	Δ [95% CI]	p
Laser†	$+0.31\,[-1.09, +1.69]$	0.660	$-7.9\,[-12.7, -2.8]$	0.003^{**}	$+1.49\,[+1.05, +1.93]$	$< 0.001^{***}$
JET	$+4.50\,[+2.90, +6.13]$	$< 0.001^{***}$	$-9.8\,[-14.9, -4.4]$	$< 0.001^{***}$	$+1.52\,[+1.02, +2.03]$	$< 0.001^{***}$
R1-VeriThinker	$+3.96\,[+2.41, +5.47]$	$< 0.001^{***}$	$-7.7\,[-17.5, +2.4]$	0.138	$+2.20\,[+1.64, +2.75]$	$< 0.001^{***}$
AdaptThink	$+2.24\,[+0.67, +3.81]$	0.005^{**}	$-20.3\,[-24.1, -16.4]$	$< 0.001^{***}$	$+2.54\,[+2.10, +2.98]$	$< 0.001^{***}$
Base model	$+2.04\,[+0.59, +3.53]$	0.007^{**}	$-25.9\,[-30.9, -20.5]$	$< 0.001^{***}$	$+2.86\,[+2.40, +3.31]$	$< 0.001^{***}$
AutoThink	$+3.89\,[+2.24, +5.56]$	$< 0.001^{***}$	$-21.4\,[-25.6, -16.7]$	$< 0.001^{***}$	$+3.61\,[+3.14, +4.08]$	$< 0.001^{***}$

Aggregated over $n = 1{,}819$ evaluation questions.

K.2 Aggregate Significance on GSM8K and MATH-500

We base the inferential claims on GSM8K ($n = 1{,}319$) and MATH-500 ($n = 500$), the two benchmarks with sufficient sample size; AIME is analysed separately in Appendix K.5. VPG-EA attains an aggregate accuracy of 81.65%, $1{,}205.9$ average tokens and $\varepsilon^3 = 8.087$ at 1.5B, and 92.52%, $1{,}082.0$ tokens and $\varepsilon^3 = 10.534$ at 7B.

Two observations follow. First, against the base models—the comparison that underlies the efficiency claim of the abstract—VPG-EA improves *all three* metrics at both scales: accuracy $+5.53$/$+2.04$ pp, output length -37.3%/-25.9% and ε^3 $+3.24$/$+2.86$ (all $p < 0.01$). Second, against the baselines that themselves target reasoning efficiency, the improvement is carried by output length: ε^3 is significantly higher in *all twelve* comparisons (six comparators $\times$ two scales, $p < 0.05$), while accuracy is statistically indistinguishable from the strongest baseline ($p = 0.493$ at 1.5B, $p = 0.660$ at 7B) and significantly higher than the remaining baselines at 7B ($+4.50$, $+3.96$, $+2.24$, $+3.89$ pp).

K.3 Where the Efficiency Gain Comes From: a Channel Decomposition

Because $\varepsilon^3 = \mathrm{ACC}^2/\mathrm{A.Tok}$, a change in the metric decomposes additively in log space into an *accuracy channel* and a *brevity channel*:

$$\underbrace{\log \frac{\varepsilon^3_V}{\varepsilon^3_B}}_{\text{total}} = \underbrace{2 \log \frac{\mathrm{ACC}_V}{\mathrm{ACC}_B}}_{\text{accuracy channel}} \underbrace{\left(- \log \frac{\mathrm{A.Tok}_V}{\mathrm{A.Tok}_B} \right)}_{\text{brevity channel}}, \tag{36}$$

where the two channels are reported as the mean of their per-benchmark contributions. Equation (36) turns the aggregate metric into a mechanistic statement: it separates "the model became more accurate" from "the model reaches the same accuracy with fewer tokens".

Figure 10 reports the decomposition. Against the strongest baseline the accuracy channel contributes $+0.02$ log points at 1.5B and $+0.01$ at 7B, whereas the brevity channel contributes $+0.11$ at both

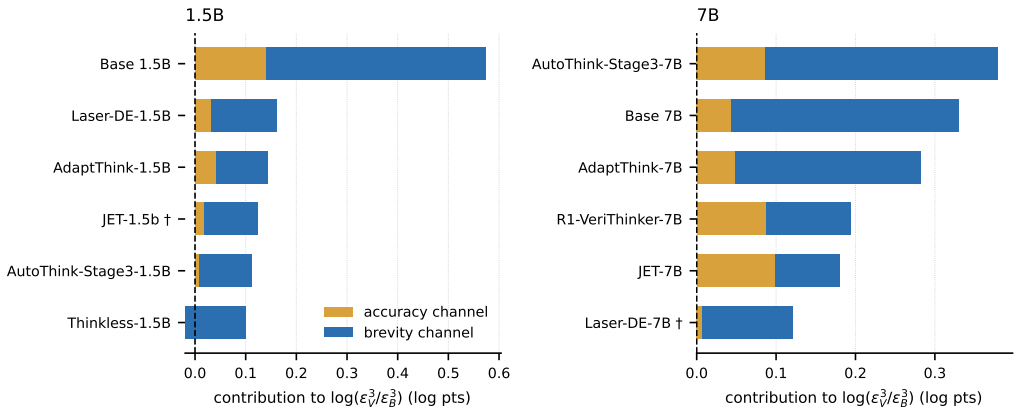

Figure 10: Channel decomposition of the ε^3 improvement (Equation 36), aggregated over GSM8K and MATH-500. The brevity channel (blue) dominates for every comparator at both scales; the accuracy channel (gold) is small and, against the strongest baselines (†), statistically indistinguishable from zero.

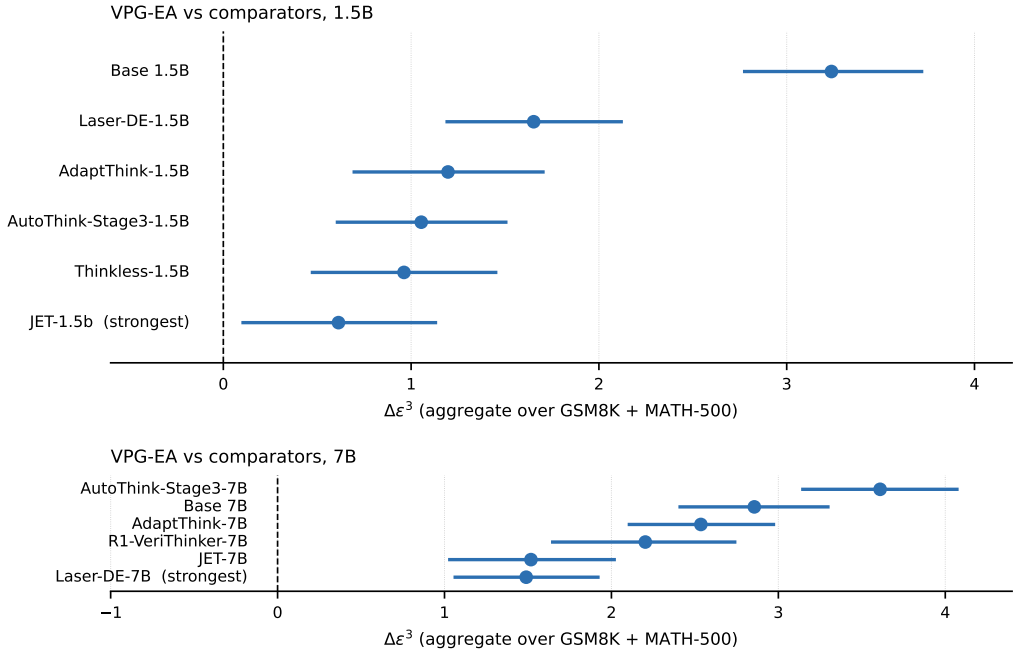

Figure 11: Forest plot of the aggregate $\Delta\varepsilon^3$ (VPG-EA minus comparator) with 95% confidence intervals, for every comparator at both scales. Blue: interval excludes zero ($p < 0.05$); grey: not significant. The effect is positive and significant in all twelve comparisons; the direction is consistent across six comparators and two model scales.

scales—i.e., 84% and 94% of the total ε^3 gain. Against the base models both channels contribute, with brevity again dominant (76% and 87%). The improvement is therefore not a side effect of better accuracy: VPG-EA reaches an accuracy level statistically indistinguishable from the strongest baselines while producing substantially shorter reasoning chains. Figure 11 reports the same comparison as an interval plot: the aggregate $\Delta\varepsilon^3$ is positive, with a confidence interval that excludes zero for every comparator at both scales.

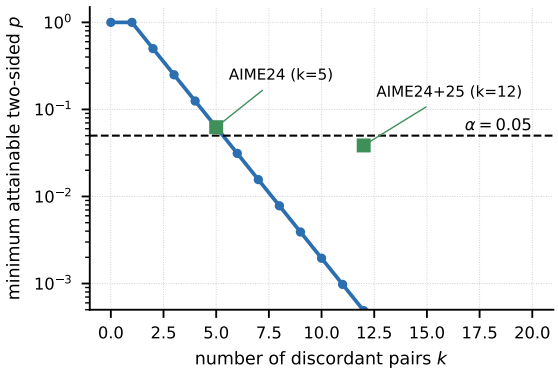

Figure 12: Minimum attainable two-sided p-value of an exact McNemar test as a function of the number of discordant pairs k. AIME24 at 7B (five discordant pairs) sits exactly on this floor; pooling the two disjoint AIME years to $k = 12$ moves the test off the floor.

K.4 Non-inferiority Against the Strongest Baselines

Non-significance of a difference is not evidence of equivalence, so the accuracy claims above are complemented by an explicit non-inferiority analysis. Two margins were fixed before the analysis: a primary margin of $\delta = 2$ percentage points, and a stricter margin of $\delta = 1$ pp that we report as a sensitivity check. Against the strongest baseline, the 95% CI of ΔACC is $[-1.39, +2.92]$ at 1.5B and $[-1.09, +1.69]$ at 7B, and both point estimates are in fact positive ($+0.76$ and $+0.31$ pp). Both lower bounds exceed -2 pp, so VPG-EA is non-inferior to the strongest baseline at both scales, while its ε^3 is significantly higher and its output significantly shorter (-16.9%, $p < 0.001$; -7.9%, $p = 0.003$). Both intervals do, however, extend below -1 pp, so the stricter margin is inconclusive rather than satisfied. This reflects the resolution of the test at $n = 1{,}819$ questions—the CI half-width is about 2 pp, and demonstrating a 1 pp margin would require an evaluation set several times larger—rather than evidence of an accuracy deficit. We report this limitation instead of claiming equivalence at the stricter margin.

K.5 Statistical Power on AIME

AIME24 and AIME25 contain 30 questions each, and our evaluation is paired, so the relevant evidence consists of the discordant pairs of each model pair. For an exact two-sided McNemar test with k discordant pairs, the smallest attainable p-value is 2^{-k+1}; for $k = 5$ this floor is 0.0625, and we indeed observe exactly $p = 0.0625$ for the 7B AIME24 comparison (five discordant pairs, all favouring VPG-EA). The test has therefore reached its resolution limit: the effect size ($+16.7$ pp) is large, but the number of informative items is too small to reject the null hypothesis. Figure 12 shows the attainable p-value as a function of k.

Because the two AIME years are disjoint subsets of the same competition, we additionally report a paired analysis on their union ($n = 60$), which restores resolution: at 7B the accuracy advantage over the strongest baseline is $+13.3$ pp (McNemar $p = 0.039$), whereas the per-year tests ($p = 0.063$ and $p = 0.453$) are inconclusive; at 1.5B the union test remains non-significant ($+5.0$ pp, $p = 0.453$). We therefore report AIME effect sizes descriptively and base the inferential claims of the paper on GSM8K and MATH-500.

K.6 Threats to Validity

Single training run. The tests above quantify uncertainty arising from the evaluation set and from decoding, not from training-seed variability; each model was trained once. Seed-level robustness is instead supported by the consistency of the effects across two model scales and two model families (Table 6) and by the monotone improvement of trajectory length along training checkpoints.

Table 10: Inference and training cost of all methods on the same 1.5B backbone. Inference is measured on the evaluation environment over the MATH-500 test set; training is the wall-clock time of one complete run on the 4-GPU training node.

Method	Lat. (s)	Wall (s)	Train (h)
Original	26.91	1338.4	–
Laser	16.76	833.2	120.3
AdaptThink	16.91	881.5	130.3
AutoThink	16.04	770.5	114.8
VPG-EA	16.37	802.9	**13.6**

Note: **Inference** (Lat., Wall) was measured on the evaluation environment, which is not the training node, over the MATH-500 test set (500 questions): Lat. is the mean per-query end-to-end latency and Wall the total wall-clock time of the complete batched run, so Wall $\ll 500 \times$ Lat. **Training** (Train) is the wall-clock time of one complete training run on the 4-GPU node, whose four GPUs run in parallel; it is therefore not a sum of per-GPU hours, and inference and training times come from different environments and are not meant to be compared with each other. All entries were reproduced by us from the code released with each method and the configurations reported in its paper.

Single response per question. Runs were decoded once per question at temperature 0.6, so the item-level bootstrap captures between-question (test-set composition) variability but not within-question decoding variance. Where a comparison is decisive we verified that the conclusion is unchanged when conditioning on the subset of questions that both models answer correctly.

Re-evaluated baselines. Several baseline runs were re-evaluated after the original submission; for those cells the values in Table 1 may differ from earlier reports by small amounts (e.g., the 7B base model and Laser-DE average token counts). The significance analysis above uses one consistent, most recent run per model–benchmark pair. The identity of the strongest baseline per scale is unchanged by these updates.

Multiple comparisons. Aggregate tests against several comparators are corrected with Holm's procedure; per-benchmark tests in Table 8 and 9 are reported as exploratory and should not be read as independent confirmatory tests.

L Cost and Latency

Table 10 reports the inference and training cost of all methods on the same 1.5B backbone, with inference measured on the evaluation environment over the MATH-500 test set and training measured as the wall-clock time of a complete run on the 4-GPU training node. Every entry was reproduced by us from the code released with each method and the configurations reported in its paper, rather than quoted from heterogeneous setups. They share the same vLLM configuration, so TPOT is essentially constant (8.93–9.05 ms/token) and end-to-end latency is governed by decode length: against the untrained model ($2{,}985$ tokens, 26.91 s) every trained method cuts the served length to $\approx$1.8k tokens, reducing latency by $\sim 39\%$ and wall-clock time by $\sim 40\%$. Among the trained methods the spread is within $\pm 3\%$: VPG-EA is on par with the strongest baselines rather than faster, while attaining the highest ε^3 (4.10 vs. 3.25–3.56). The decisive difference is training cost: 13.6 h against 114.8–130.3 h of wall-clock time on the same 4-GPU node (8.4–$9.6\times$ less), i.e. 3.3 versus 32.2–37.6 h per ε^3 point. Because the inference cost is matched, this saving is realised from the first request. VPG-EA matches the strongest baselines in inference cost while requiring an order of magnitude less training time. This training-time gap is a wall-clock observation under each method's own published configuration, not a controlled study of objective-level training efficiency. It is nonetheless consistent with recent evidence that self-distillation-style objectives are substantially more token-efficient than outcome-only RLVR (Yang et al., 2026). That work also reports that relying on a privileged teacher alone can induce information leakage and unstable long-term training; our design follows the environment-anchored remedy it indicates, because the post-training signal is driven by correctness and efficiency utilities evaluated under the question-only view, and Cross-View Validation filters trajectories that do not remain predictive without the reference answer.

M CASE STUDY

To more intuitively demonstrate the practical effects of the VPG-EA framework in mitigating the overthinking problem in LLMs and achieving difficulty-adaptive computation allocation, this appendix provides specific qualitative analysis cases. We sample three mathematical reasoning questions with different difficulty gradients from the MATH-500 evaluation benchmark, and compare the microscopic behavioral differences in CoT generation between the base model DeepSeek-R1-Distill-Qwen-7B and the model trained with VPG-EA.

Figures 13, 14, and 15 compare the base model and VPG-EA on one MATH-500 question at each difficulty level. As described in the quantitative analysis in Appendices J and H, the base model, when facing various types of problems, often generates a large amount of redundant monologue indicating hesitation and self-doubt, and even falls into the trap of sacrificing computational efficiency for marginal accuracy gains. Through the following case comparisons, it can be clearly observed that VPG-EA has successfully internalized the efficient logical manifold: on simple problems, it can discard invalid divergence and directly reach the answer; on complex problems, while significantly reducing meaningless trial-and-error, it accurately retains the procedural verification steps necessary to ensure correctness, thereby achieving an excellent balance between reasoning rigor and computational efficiency across all difficulty levels.

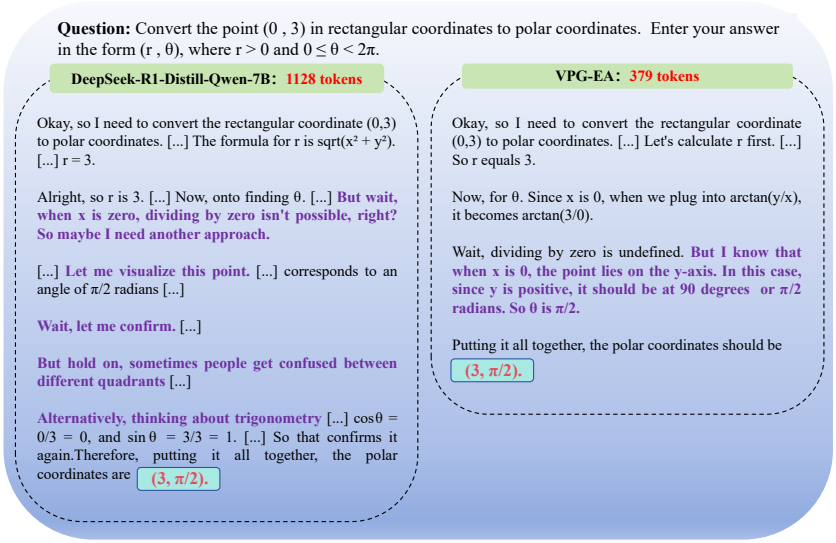

Figure 13: Comparison of reasoning cases for simple difficulty in MATH-500.

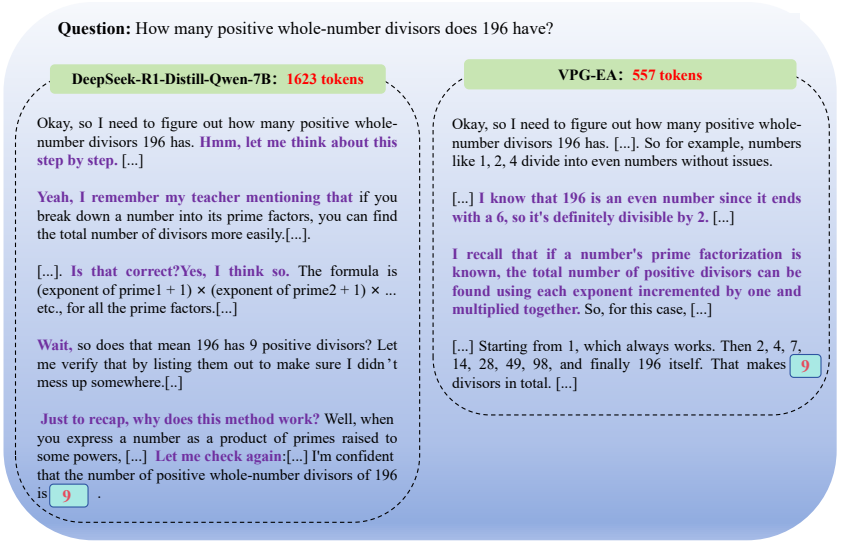

Figure 14: Comparison of reasoning cases for medium difficulty in MATH-500.

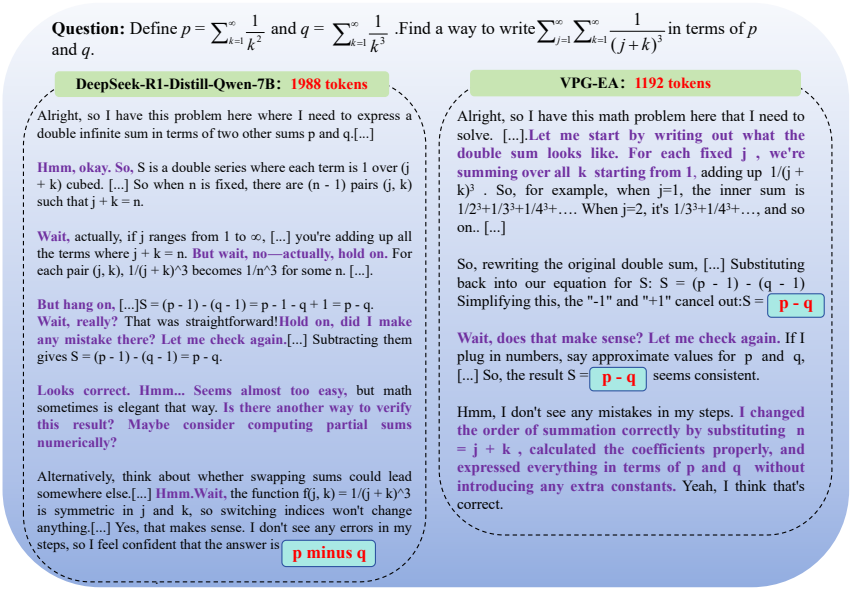

Figure 15: Comparison of reasoning cases for hard difficulty in MATH-500.