Variational Co-Evolution via Reinforcement Learning

Qiahao Zheng Hancong Jin Junbo Niu Yinjie Wang Wentao Zhang Ruixun Zhang

Peking University · University of Chicago · UESTC

COLM 2026

Motivation and contributions

Reliable reasoning depends on both **how a model solves a problem** and **how it constructs useful guidance**. Optimizing only one component can leave the other as a bottleneck.

The missing loop. Prompt search often assumes a fixed solver; solver training often assumes fixed prompts. As the model evolves, the two can become misaligned.

VCE closes the loop with three ingredients:

1. **A variational objective** that links latent prompt decisions to downstream solution quality.
2. **Outcome-driven self-reflection** that refines guidance using failed reasoning attempts.
3. **Coupled RL updates** that improve prompting and solving with one shared language model.

Variational foundation

For question q and solution observation D, treat prompt p as a latent decision. The shared policy factorizes as

$$\pi_\theta(D, p \mid q) = \pi_\theta(p \mid q)\, \pi_\theta(D \mid p, q).$$

Introduce a prompt distribution ω. Nonnegative KL divergence gives a lower bound on solution likelihood:

$$\log \pi_\theta(D \mid q) \geq \mathcal{L}(\theta, \omega)$$
$$\mathcal{L} = \mathbb{E}_{p\sim\omega}[\log \pi_\theta(D, p \mid q) - \log \omega(p \mid q)]\,.$$

$$\omega^*(p \mid q) = \pi_{\theta_{\text{old}}}(p \mid D, q)$$

Refine the prompt distribution. With the policy fixed, the optimal distribution conditions on observed solutions. Self-reflection implements this posterior-guided refinement.

Improve the shared policy. With the refined distribution fixed, increase the expected log-probability of prompts and correct solutions through reinforcement learning.

Coupled reinforcement learning

Prompt reward: prefer guidance that produces a higher fraction of correct solutions.

$$R_{p_j}^{\text{prompt}} = \frac{1}{m}\sum_{k=1}^{m} b_{j,k}$$

Each $b_{j,k} \in \{0, 1\}$ comes from the ground-truth test. Prompt quality is measured by downstream outcomes, rather than a task-agnostic heuristic.

Solution reward: calibrate correctness by the empirical success rate of its associated prompt.

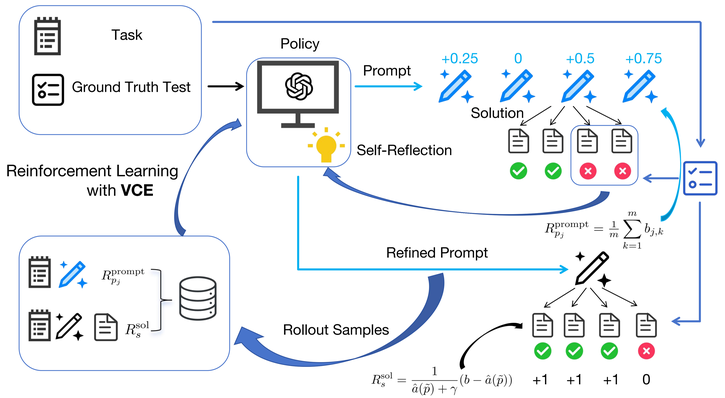
$$R_s^{\text{sol}} = \frac{b - \hat{a}(\tilde{p})}{\hat{a}(\tilde{p}) + \gamma}$$

Here $\hat{a}(\tilde{p})$ is the mean correctness under prompt $\tilde{p}$; γ stabilizes the denominator. The centered reward uses both successful and unsuccessful solutions.

Stable updates. Normalize rewards into advantages and iteratively optimize both policies with clipped policy objectives and KL regularization.

Method: Variational Co-Evolution

One LLM generates guidance, solves the task, reflects on failures, and learns from verified outcomes.



1. Generate and solve. Sample n prompts and m solutions under each prompt.

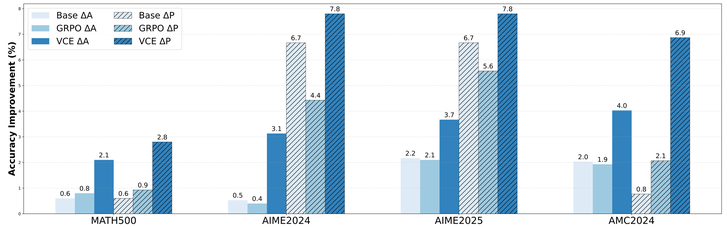
2. Verify and reflect. Test solutions; use failures to produce a refined prompt.

3. Resample solutions. Generate l new solutions for each refined prompt and verify them.

4. Co-evolve. Collect rollouts and iteratively update prompting and solving.

Prompt generation capability

Apply generated guidance to the **unchanged base solver** to isolate improvements in prompt quality.



ΔA: average accuracy gain; ΔP: Pass@16 gain. Results average three model sizes; VCE yields the largest gains on all four benchmarks.

Experimental setup

Data and models: ∼8K MATH problems (levels 3–5); Qwen3-1.7B, 4B, and 8B. Evaluation: MATH500, AIME2024, AIME2025, AMC2024.

Sampling: $n = 8$, $m = 4$, $l = 4$; temperature 1.0, top-p 0.95, top-k 20.

Optimization: learning rate 1.5×10^{-6}, KL coefficient 0.01; 8 H200 GPUs. VCE and GRPO use matched rollout-token budgets.

Main results

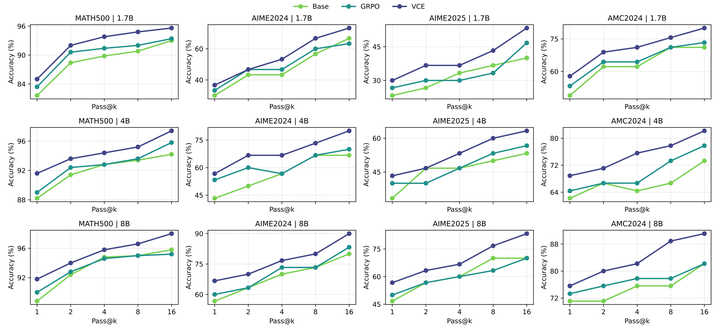
Avg / P@16: direct answering. Avg* / P@16*: self-generated guidance. Avg is averaged over 16 test runs; all values are percentages.

Model	MATH500				AIME2024				AIME2025				AMC2024			
	Avg	Avg*	P@16	P@16*	Avg	Avg*	P@16	P@16*	Avg	Avg*	P@16	P@16*	Avg	Avg*	P@16	P@16*
Qwen3-8B	91.0	91.2	95.8	95.8	59.8	60.0	80.0	83.3	46.0	47.3	70.0	73.3	68.6	69.0	82.2	80.0
Qwen3-8B + GRPO	92.0	92.0	95.2	95.4	62.3	62.9	83.3	83.3	51.7	50.8	76.7	76.7	71.3	71.9	82.2	84.4
Qwen3-8B + VCE	**93.3**	**94.0**	**98.0**	**98.8**	**64.8**	**66.0**	**90.0**	**96.7**	**54.4**	**55.2**	**83.3**	**86.7**	**74.2**	**74.9**	**91.1**	**93.3**
Qwen3-4B	88.4	88.7	94.2	94.6	48.8	47.3	66.7	76.7	32.3	35.4	53.3	60.0	58.3	60.6	73.3	75.6
Qwen3-4B + GRPO	90.1	90.0	95.8	94.6	49.2	50.6	70.0	73.3	38.5	39.4	56.7	56.7	60.7	63.5	75.6	77.8
Qwen3-4B + VCE	**91.7**	**92.6**	**97.4**	**98.2**	**51.9**	**54.0**	**80.0**	**86.7**	**40.8**	**43.8**	**63.3**	**66.7**	**63.2**	**66.1**	**82.2**	**84.4**
Qwen3-1.7B	82.9	84.2	93.0	94.4	29.2	32.1	63.3	70.0	25.4	27.5	40.0	50.0	53.8	57.2	71.1	73.3
Qwen3-1.7B + GRPO	83.6	84.8	93.4	93.8	35.2	35.6	66.7	70.0	27.1	26.7	46.7	53.3	55.6	57.8	77.8	77.8
Qwen3-1.7B + VCE	**85.8**	**87.5**	**95.6**	**97.2**	**37.5**	**39.4**	**73.3**	**80.0**	**30.6**	**32.7**	**53.3**	**63.3**	**57.9**	**59.2**	**80.0**	**82.2**

AIME2024, 8B: direct Avg rises from 59.8 to 64.8 and Pass@16 from 80.0 to 90.0 (base → VCE).

With self-generated guidance, Pass@16 reaches **96.7%**, a **13.4-point** gain over GRPO on the same setting.

Pass@k scaling



VCE's advantage persists from Pass@1 to Pass@16 across all three model scales and four benchmarks.

Ablation studies

Model	Method	MATH500				AIME2024				AIME2025				AMC2024			
		Avg	Avg*	P@16	P@16*	Avg	Avg*	P@16	P@16*	Avg	Avg*	P@16	P@16*	Avg	Avg*	P@16	P@16*
Qwen3-8B	Prompt Only	91.1	91.8	96.0	96.6	59.6	61.3	80.0	86.7	45.8	48.3	70.0	76.7	68.8	70.1	82.2	82.2
	Solution Only	91.8	92.4	96.4	97.2	62.5	63.1	83.3	86.7	51.5	52.1	76.7	80.0	71.7	72.4	84.4	88.9
	SFT	92.1	92.6	96.2	97.5	62.1	63.8	86.7	90.0	51.9	51.7	80.0	83.3	71.3	72.8	86.7	91.1
	VCE	**93.3**	**94.0**	**98.0**	**98.8**	**64.8**	**66.0**	**90.0**	**96.7**	**54.4**	**55.2**	**83.3**	**86.7**	**74.2**	**74.9**	**91.1**	**93.3**
Qwen3-4B	Prompt Only	88.5	89.4	94.4	95.2	49.0	48.8	66.7	80.0	32.5	36.9	53.3	63.3	58.1	61.7	73.3	77.8
	Solution Only	90.4	90.1	95.2	95.6	49.2	51.3	70.0	76.7	38.8	39.4	56.7	60.0	61.1	63.8	77.8	80.0
	SFT	90.2	91.1	95.8	96.5	50.2	51.0	73.3	80.0	38.5	40.4	60.0	63.3	61.7	64.4	75.6	82.2
	VCE	**91.7**	**92.6**	**97.4**	**98.2**	**51.9**	**54.0**	**80.0**	**86.7**	**40.8**	**43.8**	**63.3**	**66.7**	**63.2**	**66.1**	**82.2**	**84.4**
Qwen3-1.7B	Prompt Only	82.9	84.9	93.2	95.0	29.6	33.3	63.3	73.3	25.2	28.5	40.0	53.3	54.0	58.5	71.1	75.6
	Solution Only	83.8	85.0	93.4	94.6	35.8	35.8	66.7	70.0	27.1	28.1	46.7	53.3	56.4	57.8	75.6	77.8
	SFT	84.3	85.6	93.8	95.0	35.4	36.7	70.0	73.3	27.7	27.3	50.0	56.7	55.8	58.3	77.8	80.0
	VCE	**85.8**	**87.5**	**95.6**	**97.2**	**37.5**	**39.4**	**73.3**	**80.0**	**30.6**	**32.7**	**53.3**	**63.3**	**57.9**	**59.2**	**80.0**	**82.2**

Joint training matters. VCE outperforms prompt-only, solution-only, and SFT training. SFT uses positive rollouts; VCE also learns from negative outcomes.

Takeaway: Better guidance and stronger solvers improve together under a unified variational RL objective.

openreview.net/forum?id=x2pleWjnM9