Beyond Pass@1: Self-Play with Variational Problem Synthesis Sustains RLVR (SvS)

ICLR 2026

Arxiv: https://arxiv.org/abs/2508.14029

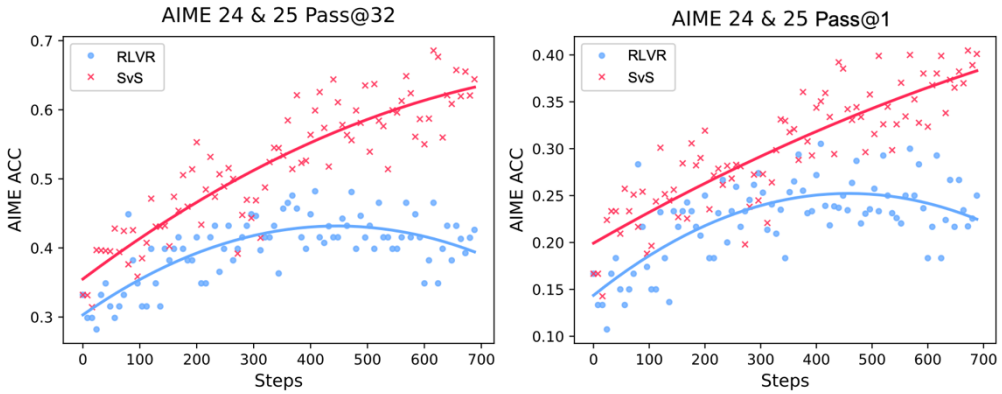
Github: https://github.com/MasterVito/SvS

[3] Liang, Xiao, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu Chen. "Beyond pass@1: Self-play with variational problem synthesis sustains rlvr." arXiv preprint arXiv:2508.14029 (2025).

Motivation for SvS

- Standard RLVR fails to improve **Pass@k** due to limited exploration, often caused by **entropy collapse**.
- Preserving **data diversity** is the most straightforward solution.
- SvS: Leveraging **online self-play** data augmentation to enhance model generalizability in RLVR.

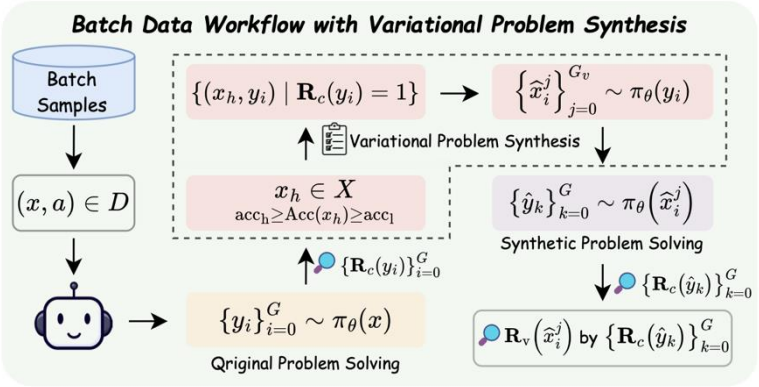
Qwen2.5-32B-IT on DAPO-Math-17k:

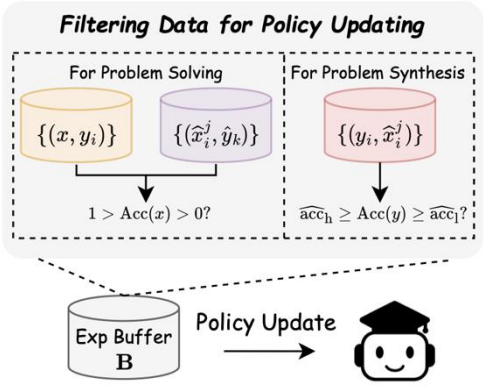


SvS - Method

Overview

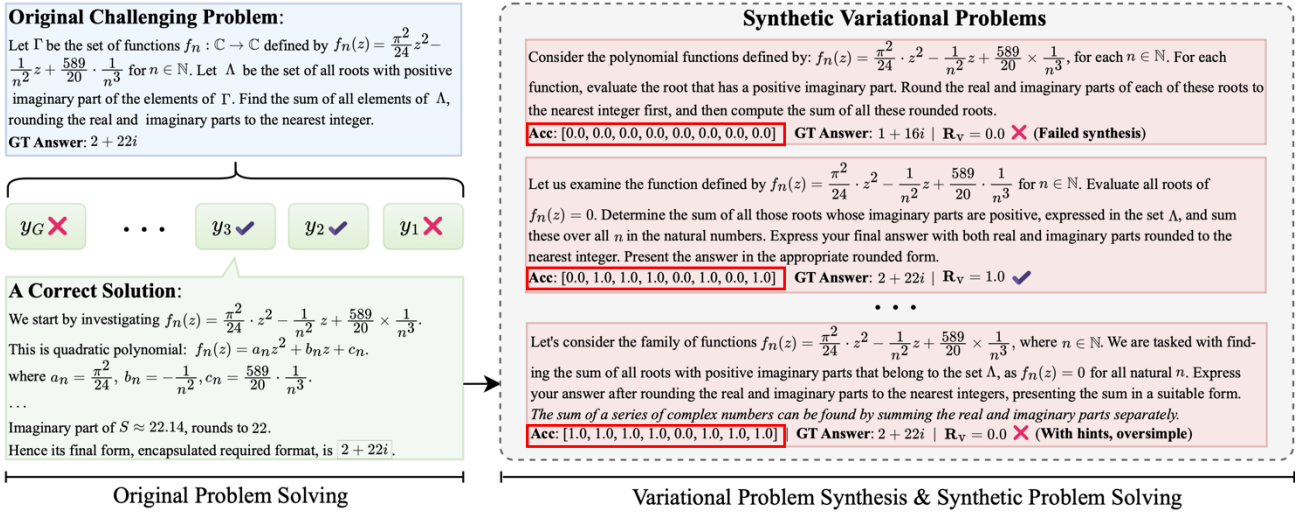
- **Self-play with Variational problem Synthesis** (SvS) strategy for RLVR.
- Variational problems are synthesized **from correct solutions to challenging tasks.**
- The variational problems are constrained to **share the same reference answers** as the original ones.
- Each step include: (1) Original Problem Solving, (2) Variational Problem Synthesis (3) Synthetic Problem Solving.





Example of SvS Augmentation

- The challenging problems are identified with **group accuracy between $\mathbf{acc}_l$ (0.125) and $\mathbf{acc}_h$ (0.5).**
- We filter verified **correct solutions** to generate multiple (n=8) synthetic problems.
- The policy generates a group of solutions to each synthetic problems.



☐ indicates policy accuracy on the synthetic problems, verified using the original problem's reference answer.

Reward Shaping for Problem Synthesis

- Correctness: The synthetic problems should **preserve the original reference answer**.
  - In our implementation, we guide the policy to solve the synthetic problems and verify whether it produces the original reference answer as a proxy. **At least one** such solution is required to validate each synthetic problem.
- Helpfulness: The synthetic problems should **remain challenging** for the policy at the current iteration.
- Final reward design for problem synthesis:

$$\mathbf{R}_{\mathrm{v}}(\hat{x}_i^j) = \mathbb{I}\left(\hat{\mathrm{acc}}_{\mathrm{l}} \leqslant \mathrm{Acc}(\hat{x}_i^j, a) \leqslant \hat{\mathrm{acc}}_{\mathrm{h}}\right)$$

1/8, maintain correctness

5/8, maintain helpfulness

SvS - Experiments

Experimental Setup

- Training Models: Qwen2.5-Instruct (3B, 32B), LLaMA-3.1-Instruct-8B.
- Datasets:
  - For all models: MATH-12k.
  - For the 32B models, we additionally run the experiments on DAPO-Math-17k.
- RL Algorithm: **GRPO + Dynamic Sampling + Clip-Higher + Token-level Reward**.
- Benchmarks: GSM8k, MATH-500, Minerva-Math, Olympiad-Bench, Gaokao23, AMC23, AIME24 & 25, Beyond-AIME, Math24o, OlymMath.

SvS - Experiments

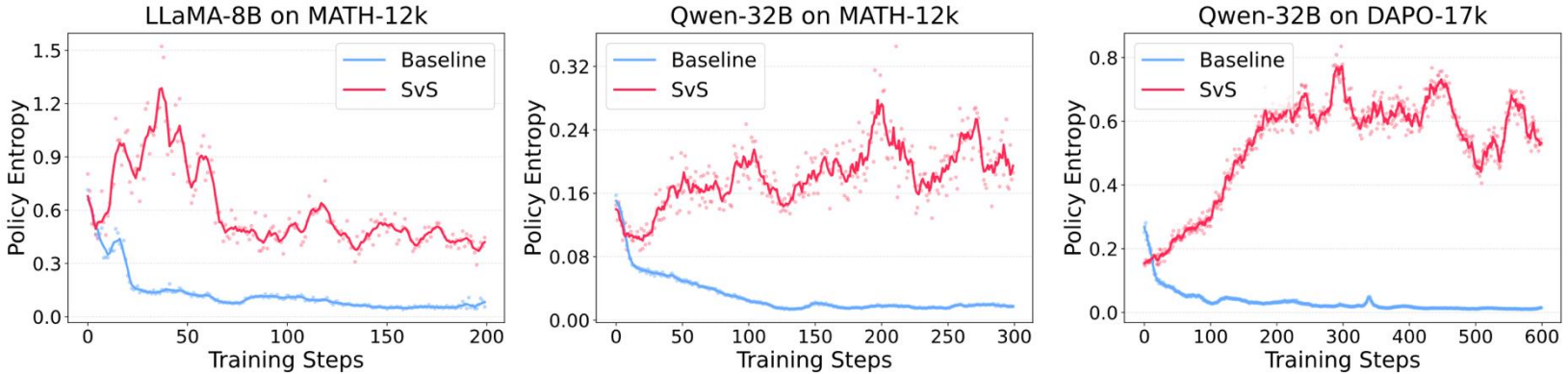
Main Results

- The SvS training demonstrate **much better performance** in comparison with standard RLVR training.
  - Pass@32: AIME 24 +18.3%; AIME 25 + 22.3%.
- Performance drop on OlymMath arises from answer formats that are o.o.d to the training data.

Model	Pass@1							Pass@32						
	AIME24	AIME25	BAIME	Math24o	OlymE	OlymH	Avg.	AIME24	AIME25	BAIME	Math24o	OlymE	OlymH	Avg.
Open-Source Models														
Qwen2.5-32B	4.3	1.2	2.4	8.0	3.7	1.6	3.5	38.9	15.6	18.7	34.0	24.6	15.2	24.5
Qwen2.5-32B-IT	10.0	13.0	7.4	26.0	8.6	2.0	11.2	40.2	34.6	24.0	67.8	35.2	9.5	35.2
SimpleRL-32B	22.1	13.9	8.3	25.5	9.4	3.7	13.8	62.0	38.5	27.4	69.9	42.5	19.4	43.3
ORZ-32B	24.2	26.3	10.9	16.1	12.2	1.1	15.1	55.7	47.0	29.4	58.0	45.9	12.3	41.4
MATH-12k														
→ RLVR	22.2	15.8	11.5	34.5	11.7	**4.1**	16.6	47.4	36.4	29.2	66.0	36.2	16.4	38.6
→ SvS	**30.3**	**21.7**	**13.8**	**42.7**	**20.1**	3.3	**22.0**	**63.6**	**55.1**	**41.5**	**79.2**	**63.6**	**24.8**	**54.6**
Δ	+8.1	+5.9	+2.3	+8.2	+8.4	-0.8	+5.4	+16.2	+18.7	+12.3	+13.2	+27.4	+8.4	+16.0
DAPO-17k														
→ RLVR	28.8	30.0	14.0	39.6	17.9	**4.8**	22.5	52.5	42.4	35.9	71.2	**47.1**	**18.3**	44.6
→ SvS	**39.3**	**40.5**	**19.2**	**44.1**	**21.8**	2.7	**27.9**	**70.8**	**65.2**	**45.9**	**76.5**	43.4	16.7	**53.1**
Δ	+10.5	+10.5	+5.2	+4.5	+3.9	-2.1	+5.4	**+18.3**	**+22.8**	+10.0	+5.3	-3.7	-1.6	+8.5

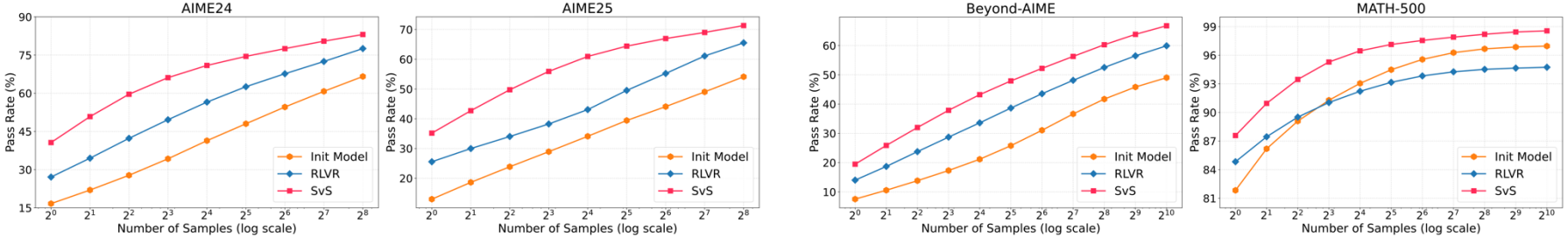
SvS Stably Maintains Policy Entropy in Training

- Training entropy and generation diversity are preserved because the **policy is forced to solve new problems at each RL step.**



SvS Pushes the Reasoning Boundary of the Policy

- SvS training augmentation consistently yields substantially stronger Pass@k performance than standard RLVR and the initial model, even as k increases to 1,024.
- On MATH-500, RLVR fails to outperform the initial policy at Pass@1024, **whereas SvS achiecves so.**



SvS - Experiments

SvS generalizes effectively to code generation

- We run Qwen2.5-7B-Instruct on code generation tasks **without tuning any hyperparameters**.
- Correct code programs are used to synthesize the corresponding coding task descriptions.
- Even show more promising results than on the mathematical reasoning tasks.