# ReST-RL: Reinforcing LLM Reasoning through Unified Self-Training and Value-Guided Search

**Sining Zhoubian**[*], **Dan Zhang**[†], **Jie Tang**[†]

Department of Computer Science and Technology, Tsinghua University

[*] Work done when interned at Z.ai. [†]Corresponding authors.

NEURAL INFORMATION PROCESSING SYSTEMS

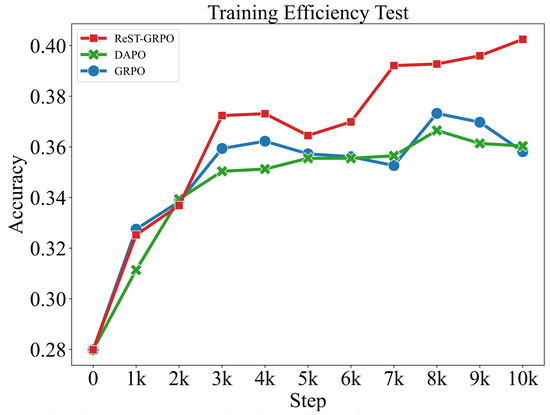
2026

## Restart from informative states. Search with learned values.

ReST-GRPO improves the policy's trajectories; a matched value model turns them into inference-time guidance.

### 01 Learn where to restart

When sampled answers receive similar rewards, GRPO gets little relative signal. **ReST-GRPO changes where online RL begins.**




Llama-3-8B; average benchmark score over 10k updates. [Fig. 3a]

#### ReST-GRPO: restart, then explore

**1. Filter** prompts by reward variation; select the highest-reward trace $A$ as the anchor.

**2. Add partial starts.** Retain original prompts; sample $\beta|A|$ prefixes, biased toward earlier states.

**Finite exponential sampling**

$$p(j) = \frac{1-\alpha}{1-\alpha^{|A|}}\alpha^{j-1}, \quad j = 1, \ldots, |A|, \quad 0 < \alpha < 1$$

$j$: prefix length in lines; $\alpha = 0.95$, $\beta = 0.5$ by default. [Eq. 2; Table 21]

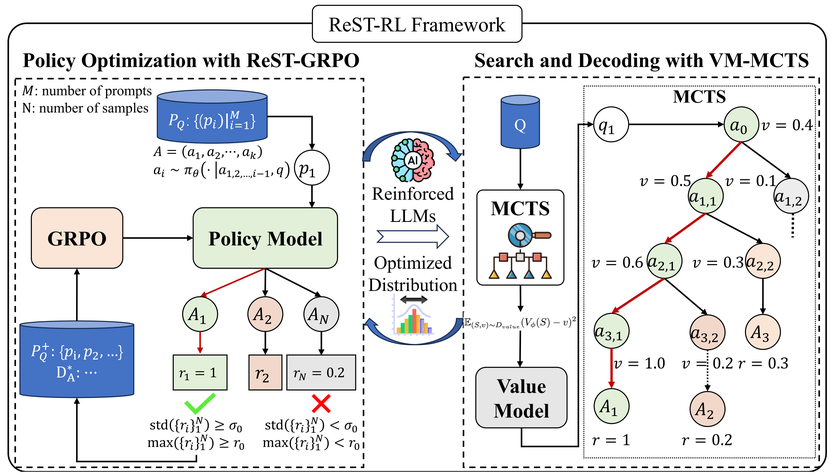
**3. Optimize fresh suffixes** with GRPO. Prefixes are context, never imitation targets.

#### Policy-only gain over DAPO: +5.8 pp

| ReST-DPO | GRPO | DAPO | **ReST-GRPO** |
|---|---|---|---|
| 52.6 | 59.0 | 59.7 | **65.5** |

Qwen3-8B, two training iterations; four-family average score (%). [Table 1; averaging rule at right]

### 02 Unified policy + value




ReST-GRPO improves trajectories for downstream value learning. [Fig. 2]

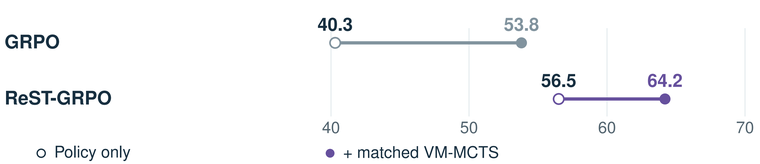
$$V^{\pi}(s) = \mathbb{E}_{\pi}[R(S_{\text{end}}) \mid s]$$

**Freeze the improved policy.** Learn state values from self-collected Monte Carlo returns, without extra human step annotations.

#### One VM for tree allocation + final selection.

Sequential training; $s = (q, a_{1:j})$, one action per line. Each VM is paired with its generating policy. [Sec. 2.3]

#### The policy gain survives matched VM learning



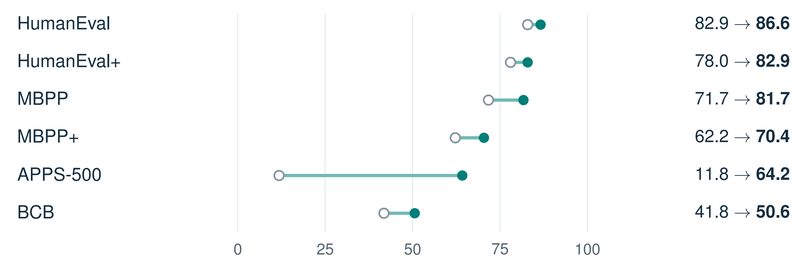

APPS-500 test-case pass rate (%). Same VM initialization, 1.1M targets and search settings. [Table 3b]

### 03 Stronger code reasoning

# 50.3 → 68.9

**Qwen3-8B: base → full ReST-RL**
Four-family average score (%); **+18.6 pp**.

Full ReST-RL uses 100 candidates at inference.




APPS: test-case pass rate; others: pass@1. [Table 17]
Avg.: average each + pair, then equally weight HumanEval, MBPP, APPS and BCB.

#### Gains across model families

| Model | Base | **ReST-RL** |
|---|---|---|
| Qwen2.5-Coder-7B | 56.3 | **67.3** |
| DeepSeek-Coder-6.7B | 49.3 | **58.4** |
| OpenCI-DS-6.7B | 48.6 | **58.3** |

Four-family average score (%); 100 candidates. [Table 2]

#### Code training transfers beyond code

Qwen3-8B; no target-domain tuning. [Table 5]

| | | |
|---|---|---|
| MATH | 78.0 → **87.2** | **+9.2 pp** |
| Omni-MATH | 23.4 → **25.6** | **+2.2 pp** |
| GPQA-Diamond | 44.9 → **49.0** | **+4.1 pp** |

Preliminary transfer evidence; coding is the primary evaluation domain.

**2.9× faster to a +9-point gain**

71 vs. 207 hours, including pre-sampling.
Llama-3-8B; ReST-GRPO vs. GRPO. [Table 4]

**64.2 vs. 61.9 at ≈41.5 s / prompt**

APPS-500; VM-MCTS vs. VM-BoN ($N = 260$).
Same policy + VM; approximately matched latency. [Table 3c]

**Code**
THUDM/
ReST-RL

**Sining Zhoubian**
zbsn21@mails.tsinghua.edu.cn
Email / WeChat →