CARVE: COORDINATING IMITATION AND EXPLORATION THROUGH REWARD VARIANCE FOR LLM REASONING POST-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) commonly follows supervised fine-tuning (SFT) in the post-training of reasoning models. However, the two stages often reuse the same problems even though group-relative RL depends on reward variation among rollouts. SFT can raise the solve rates of these problems and leave RL with many all-correct groups that provide no effective gradient, a failure mode we term *variance starvation*. Binary outcome rewards also cannot distinguish among correct derivations, while process supervision becomes less informative when trained on similarly saturated problems. To address these challenges, we propose CARVE, a variance-aware post-training framework with two coordinated designs. First, it separates SFT and RL problems and uses the fine-tuned parent to retain RL prompts with informative outcome variation. Second, it trains a process reward model on a subset of the screened problems, validates its selection ability before use, and restricts it to lowering the reward of weaker correct rollouts. The screened problems therefore provide both useful reward variation for policy optimization and informative supervision for process learning. Experiments across ten benchmarks and two backbone show that CARVE consistently outperforms shared-pool post-training baselines on mathematical reasoning while maintaining comparable performance on code and general reasoning. It achieves these gains with half the SFT data and substantially fewer RL updates.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a standard stage in the post-training of reasoning models (Jaech et al., 2024; Lambert et al., 2024; Guo et al., 2025). A common recipe first fine-tunes a base model on verified chain-of-thought (CoT) demonstrations (Wei et al., 2022) and then optimizes the resulting policy with a group-relative objective such as GRPO (Shao et al., 2024). GRPO estimates advantages across rollouts of one prompt, so identical rewards yield no policy gradient. Under binary outcome rewards, all-correct and all-wrong groups are therefore uninformative. The learning signal depends on the rollout policy shaped by preceding SFT. SFT and RL are thus coupled because fine-tuning changes the solve rates of problems later used for exploration. Yet standard post-training reuses problems across stages despite this coupling.

The first challenge arises from reusing the same problems across SFT and RL. Fine-tuning increases solve rates, reducing rollout disagreement during RL. A problem solved with 95% probability, for example, produces an all-correct group in 81% of four-rollout draws. We refer to this loss of within-group reward variation as *variance starvation*, which leaves an increasing fraction of RL rollouts without effective policy gradients. Existing approaches mainly improve data efficiency after the RL pool has been formed. Dynamic sampling and selective rollout methods filter uninformative groups (Yu et al., 2026; Zheng et al., 2026), data selection prioritizes prompts by learning utility or intermediate pass rate (Li et al., 2025; Bae et al., 2026), and mixed-policy methods introduce demonstrations during RL (Yan et al., 2026; Ma et al., 2026). These methods do not account for reward variance already reduced by the preceding SFT stage. Existing stage-wise splits use reasoning difficulty or base-model pass rates (Hu et al., 2026; He et al., 2026), missing the post-SFT policy.

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.

The second challenge is that binary outcome rewards cannot distinguish among correct rollouts. Correct solutions can include sound derivations and unreliable shortcuts that receive the same outcome reward. Process reward models (PRMs) provide step-level supervision (Lightman et al., 2024; Wang et al., 2024), but their value depends on labeling data and their use in RL. Monte-Carlo process labels become weakly discriminative when the parent already solves most problems in the labeling pool, since successful continuations dominate the resulting supervision. Variance starvation thus also weakens process supervision. Dense process rewards introduce a second risk because the policy may increase the process score without improving answer accuracy (Gao et al., 2023). Process supervision thus needs informative labels and bounded rewards.

To address these challenges, we propose CARVE (Complementary Allocation of Reward Variance for Exploration), a post-training framework that coordinates data allocation and process supervision around the policy entering RL. CARVE first separates the corpus into problem-disjoint SFT and RL pools, then uses the fine-tuned parent to retain RL problems with intermediate solve rates and thus informative outcome variation. A subset of the screened problems supplies PRM training rollouts from the parent, providing more discriminative process supervision. To limit reward exploitation, the PRM is incorporated only after demonstrating selection accuracy beyond majority voting and is restricted to down-weighting correct rollouts, with its strongest effect reserved for groups lacking outcome variation. This design couples the two components through a common source of informative problems, which provides reward variation for policy optimization and useful negative evidence for process learning. Extensive experiments across ten benchmarks and multiple backbone scales validate the effectiveness of both components and their behavior under different data and optimization budgets. Our contributions can be summarized as four-fold below:

- We identify *variance starvation*, a cross-stage failure mode in which supervised fine-tuning reduces the within-group reward variation required by subsequent group-relative reinforcement learning, progressively weakening its effective learning signal.
- We introduce variance-aware data allocation, which separates SFT and RL problems and uses the fine-tuned parent to retain prompts with informative outcome variation, concentrating RL updates on problems that remain learnable.
- We develop a gated dock-only process reward that trains a PRM on the screened RL problems, validates its selection ability before integration, and restricts its influence to distinguishing correct rollouts without increasing rewards for incorrect ones.
- Extensive experiments on ten benchmarks shows CARVE outperforms shared-pool post-training baselines on math while maintaining comparable code and general reasoning performance, using half the SFT data and fewer RL updates.

2 RELATED WORK

RL With Verifiable Rewards. RLVR is standard in reasoning-model post-training (Guo et al., 2025; Lambert et al., 2024; Team et al., 2025). GRPO replaces the learned critic in PPO (Schulman et al., 2017) with group-relative rollout advantages (Shao et al., 2024), which resemble the leave-one-out baseline in RLOO (Ahmadian et al., 2024). Nonzero advantages therefore depend on reward variation within each group. DAPO resamples batches to remove all-correct and all-wrong groups (Yu et al., 2026), GRESO filters such prompts before rollout (Zheng et al., 2026), and Dr. GRPO corrects biases from group normalization (Liu et al., 2025). Data selection prioritizes estimated learning impact (Li et al., 2025), intermediate pass rates associated with larger policy-improvement bounds (Bae et al., 2026), or persistent success and failure patterns (He et al., 2025). Other studies examine the interaction between SFT and subsequent RL through generalization (Chu et al., 2025), changes in reasoning capability (Chen et al., 2026), and the weak correspondence between SFT performance and RL gains (Kang et al., 2026). Mixed-policy approaches combine imitation and RL signals within a single stage (Yan et al., 2026; Ma et al., 2026; Fu et al., 2026), while carefully configured sequential post-training remains competitive (Limozin et al., 2026). Recent work allocates SFT and RL data using judged reasoning intensity (Hu et al., 2026) or base-model pass rates (He et al., 2026). These splits precede SFT and cannot reflect the resulting policy's reward variation.

Process Reward Models. PRMs supervise reasoning steps using human annotations (Uesato et al., 2022; Lightman et al., 2024) or Monte-Carlo continuation estimates (Wang et al., 2024; Luo

et al., 2024). Monte-Carlo labels can mark flawed prefixes as correct when their continuations reach the answer, motivating LLM-judge consensus filtering (Zhang et al., 2025) and step-error evaluation with ProcessBench (Zheng et al., 2025a). At inference, PRMs rank sampled solutions (Cobbe et al., 2021; Li et al., 2023), often with self-consistency (Wang et al., 2022). Dense PRM rewards risk process-level reward hacking under repeated optimization (Gao et al., 2023; Cheng et al., 2026). Recent work therefore explores implicit or progress-based rewards (Cui et al., 2025; Setlur et al., 2025) and process–outcome agreement for rollout filtering rather than direct reward shaping (Ye et al., 2025). Self-improvement methods further reuse policy-generated trajectories as fine-tuning data (Zelikman et al., 2022; Gulcehre et al., 2023; Singh et al., 2023) or to co-evolve a process model (Guan et al., 2025). Our proposed CARVE trains a self-labeled PRM on screened post-SFT problems, validates it as a selector, and only down-weights correct rollouts.

3 METHODOLOGY

Problem Formulation. Let $\mathcal{D} = \{(x_k, a_k^\star, \zeta_k)\}_{k=1}^N$ be a corpus of problems x with gold answers $a^\star$ and verified CoT solutions ζ, and let $v(y, a^\star) \in \{0, 1\}$ be an answer verifier. A two-stage post-training procedure fine-tunes a base model on a subset $\mathcal{D}_{\text{SFT}} \subseteq \mathcal{D}$ to obtain a parent π_0, and then optimizes π_0 by RL on prompts from a subset $\mathcal{D}_{\text{RL}} \subseteq \mathcal{D}$ under a reward r, yielding π_θ. For a prompt x, GRPO independently samples G rollouts $\{y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and assigns each the group-relative advantage

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G) + \epsilon}. \tag{1}$$

Under a binary reward $r_i = v(y_i, a^\star)$ and solve rate $p(x)$ under the rollout policy $\pi_{\theta_{\text{old}}}$, a group satisfies $\hat{A}_i \equiv 0$ with probability $p(x)^G + (1 - p(x))^G$, which approaches one as $p(x)$ approaches either extreme. A stage that drives $p(x)$ toward one on the prompts of the next stage therefore causes variance starvation. The goal of post-training is the expected verified accuracy of the resulting policy,

$$\mathcal{P}^\star = \arg\max_{\mathcal{P}} \; \mathbb{E}_{(x,a^\star)\sim P_{\text{test}}} \, \mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \big[v(y, a^\star)\big] \quad \text{s.t.} \quad T_{\text{RL}} \le B, \tag{2}$$

where π_θ results from procedure $\mathcal{P}$, T_{RL} is the number of RL steps, and B is a fixed budget. A procedure $\mathcal{P} = (\mathcal{D}_{\text{SFT}}, \mathcal{D}_{\text{RL}}, r)$ is thus specified by the problems each stage consumes and the reward the RL stage optimizes. Under a fixed budget, a group with $\hat{A}_i \equiv 0$ contributes no reward-driven policy gradient. Thus pool choice shapes update efficiency.

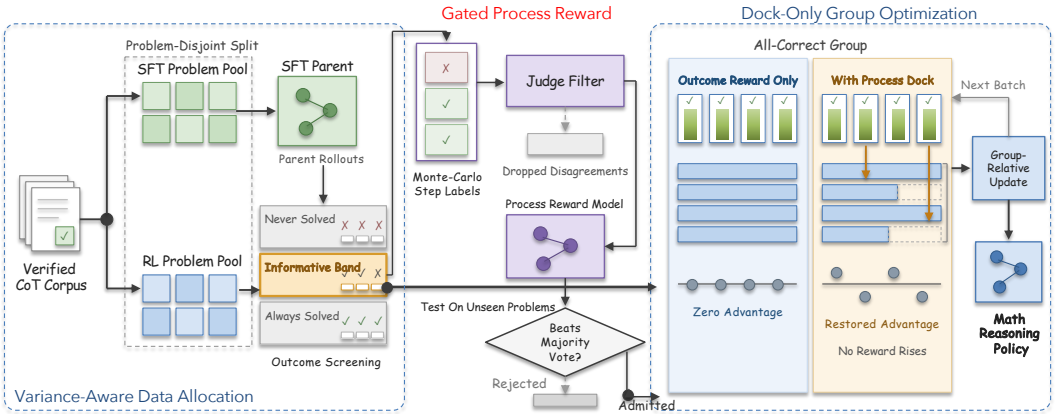

Figure 1: Overview of CARVE. **Variance-aware data allocation** (left) splits the corpus into problem-disjoint SFT and RL pools and keeps the informative band: the problems the parent solves sometimes but not always. **Gated process reward** (middle) trains a PRM on the parent's step labels and admits it only if it beats majority voting. **Dock-only group optimization** (right) docks weaker derivations inside an all-correct group, creating advantage binary correctness cannot.

Overview. CARVE follows the three-stage pipeline in Figure 1. *Variance-aware data allocation* partitions $\mathcal{D}$ into problem-disjoint SFT and RL pools, trains the parent π_0 on $\mathcal{D}_{\text{SFT}}$, and uses its rollout outcomes to retain an informative prompt set $\mathcal{X}_{\text{band}}$ from $\mathcal{D}_{\text{RL}}$. A subset of the screened problems

then supports *gated process reward*, which trains a PRM from the parent's process supervision and admits it only after validating its selection quality on held-out problems. Finally, *dock-only group optimization* applies group-relative RL on $\mathcal{X}_{\text{band}}$, where the admitted PRM down-weights correct rollouts and thereby restores relative learning signals without increasing rewards for incorrect ones.

Data Allocation And Outcome Screening. We partition $\mathcal{D}$ into problem-disjoint SFT and RL pools and train the parent π_0 only on $\mathcal{D}_{\text{SFT}}$. Reserving $\mathcal{D}_{\text{RL}}$ for policy optimization keeps its problems unseen during imitation. Let $\mathcal{X}_{\text{RL}} = \{x : (x, a^\star, \zeta) \in \mathcal{D}_{\text{RL}}\}$ be its prompts. For each $x \in \mathcal{X}_{\text{RL}}$, π_0 draws n solutions at the RL temperature, and $h(x)$ counts the correct ones. The RL stage keeps the informative band, which can be formulated as

$$\mathcal{X}_{\text{band}} = \{x \in \mathcal{X}_{\text{RL}} : \ 1 \le h(x) \le n-1\}, \tag{3}$$

which discards never- and always-solved problems before any RL step, as in the screening rows of Figure 1. Pass-rate filtering is established (He et al., 2025; Bae et al., 2026). Here it runs once, with the parent, on problems it has not imitated, and the band doubles as the PRM labeling pool. The source-stratified assignment and duplicate checks are specified in Appendix B.2.

Monte-Carlo Step Labels. A response y is segmented into steps $y = (s_1, \ldots, s_K)$ at blank lines, explicit step headers, and sentences opening with a derivation connective, without splitting inside LaTeX environments. Near-duplicate steps are removed and the number of steps is capped. The label of step k is estimated from M continuations of the prefix $s_{\le k}$ under π_0 (Wang et al., 2024),

$$\ell_k = \mathbf{1}\Big[\tfrac{1}{M}\textstyle\sum_{m=1}^{M} v\big(s_{\le k} \oplus \tilde{y}^{(m)}, a^\star\big) \ge \tau\Big], \qquad \tilde{y}^{(m)} \sim \pi_0(\cdot \mid x, s_{\le k}), \tag{4}$$

where the threshold τ is fixed in advance. Correct solutions are labeled positive throughout, and incorrect solutions are labeled by binary search for their first unrecoverable step (Luo et al., 2024). A judge independently locates the first erroneous step, and disagreements are adjudicated before PRM training (Zhang et al., 2025). The PRM ϕ shares the backbone of π_0 with a token-level head read at the end of each step, producing a step score $\sigma(z_k)$ from logit z_k, and is trained with

$$\mathcal{L}_{\text{PRM}}(\phi) = \mathcal{L}_{\text{BCE}}(z, \ell) + \lambda\, \mathbb{E}_{(j,k):\, \ell_j - \ell_k \ge \delta_\ell}\big[\log\big(1 + e^{-(z_j - z_k)}\big)\big], \tag{5}$$

where the class-balanced BCE term fits step correctness and the pairwise term, over within-solution pairs with label-gap threshold δ_ℓ and weight λ, enforces the ordering used by docking. The pairwise term is zero when no pair qualifies. The PRM enters RL only if its positive-label rate falls within an admissible interval and its held-out solution selection accuracy exceeds majority voting. Otherwise, RL uses the outcome reward alone. This gate tests the within-problem ranking needed for docking, with selector validation detailed in Appendix B.6.

Dock-Only Reward. For rollout y_i with K_i steps, the admitted PRM yields a mean score $K_i^{-1}\sum_{k=1}^{K_i}\sigma(z_{i,k})$. Its quality $q_i \in [0,1]$ is reduced when none of the closing steps reaches $a^\star$ and capped when any step falls below a break threshold (break flag $b_i = 1$). The depth $d_i = \min(1, K_i/K_0)$ discounts derivations shorter than reference length K_0. Within group g with correct rollouts $\mathcal{C}_g$, quality is centered as $\tilde{q}_i = \text{clip}\left(\frac{1}{2} + q_i - \text{mean}_{j\in\mathcal{C}_g} q_j, 0, 1\right)$, with $\tilde{q}_i = 1$ when $|\mathcal{C}_g| = 1$, so that the score reflects relative derivation quality rather than problem difficulty. Writing R for the correct-answer reward ceiling, a correct rollout receives

$$r_i = \min\Big(R,\ \max\big(c,\ R - \beta\,\gamma_g\,\big(1 - \tilde{q}_i\,((1-\alpha) + \alpha\, d_i)\big) - w_b\, b_i\big) - \kappa_i\Big), \quad \gamma_g = \begin{cases} 1, & |\mathcal{C}_g| = G, \\ \gamma_{\text{mix}}, & \text{otherwise}, \end{cases} \tag{6}$$

while an incorrect rollout receives the outcome reward alone, capped at c. Here β is the maximum dock, α weights the depth term, w_b is the break penalty, c the ceiling of incorrect rollouts, and $\kappa_i \ge 0$ collects format penalties applied to rollout i. Provided that $c \le R - \beta - w_b$, the form of Eq. 6 guarantees three properties: a correct rollout never exceeds the outcome ceiling R; every correct rollout stays at or above every incorrect one before format penalties; and incorrect rollouts receive no process credit. The schedule γ_g applies the full dock in an all-correct group, where binary correctness gives $\hat{A}_i \equiv 0$, and a shallow dock $\gamma_{\text{mix}} < 1$ in mixed groups, where correctness already separates rollouts. The right stage of Figure 1 contrasts the two states: equal rewards and a flat advantage axis under binary correctness, and unequal rewards with advantage on both sides of the baseline once the dock applies.

Optimization Objective. With advantages $\hat{A}_i$ computed from Eq. 6 by Eq. 1, the policy maximizes the token-averaged clipped objective over the informative band,

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{X}_{\text{band}},\,\{y_i\}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{\sum_i|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\Big(\rho_{i,t}\hat{A}_i,\ \text{clip}\big(\rho_{i,t},1-\varepsilon_{\text{low}},1+\varepsilon_{\text{high}}\big)\hat{A}_i\Big)\right] - \beta_{\text{KL}}\,\mathbb{D}_{\text{KL}}\big(\pi_\theta\,\|\,\pi_0\big), \tag{7}$$

where $\rho_{i,t} = \pi_\theta(y_{i,t} \mid x, y_{i,<t})/\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})$, the bounds ε_{low} and $\varepsilon_{\text{high}}$ clip the importance ratio asymmetrically (Yu et al., 2026), and $\mathbb{D}_{\text{KL}}$ is the token-averaged KL on the sampled prompts, estimated with the low-variance estimator (Schulman, 2020) with coefficient β_{KL} toward π_0. The PRM is trained once with Eq. 5 and stays frozen while the policy is optimized over batches from the informative band. The allocation thus determines where a gradient exists, and the reward determines what separates rollouts once it does.

Theorem 1 (Complementary Signal From Screening And Docking)**.** *Let $p \sim P$ be the fixed-parent solve rate across RL problems, $S_k(p) = 1 - p^k - (1-p)^k$, and $Z = \mathbb{E}_P[S_n(p)] > 0$. Under the assumptions in Appendix A, the probability I that a screened problem produces nonconstant rewards in G rollouts is*

$$I = \mathbb{E}_P[S_G(p)] + \frac{\text{Cov}_P(S_n, S_G)}{Z} + \frac{\mathbb{E}_P[S_n(p)p^G\eta(p)]}{Z}, \tag{8}$$

where $\eta(p)$ is the conditional chance of nonconstant centered PRM scores in an all-correct group. Both added terms are nonnegative. The covariance is positive whenever $p(1-p)$ is nonconstant.

Theorem 2 (Correct-Answer Advantage Under Docking)**.** *Consider a mixed group with $1 \le k = |\mathcal{C}_g| < G$, a common format penalty $\kappa \ge 0$, incorrect rewards at most c, and $D_g = \beta\gamma_{\text{mix}} + w_b$. Under Eq. 6, each correct rollout has advantage*

$$\hat{A}_i \ge \frac{(G-k)(R-c-\kappa) - (G-1)D_g}{G(s_g+\epsilon)}, \qquad i \in \mathcal{C}_g, \tag{9}$$

where s_g is the group's reward standard deviation and $\epsilon > 0$. Thus every correct rollout retains a positive advantage whenever $D_g < \frac{G-k}{G-1}(R-c-\kappa)$.

A shallow mixed-group dock can therefore preserve correct-answer advantages while the full dock distinguishes all-correct rollouts. Full proofs are provided in Appendix A.

4 Experiments

Benchmarks & Metrics. We evaluate mathematical reasoning on GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), MATH-500 (Lightman et al., 2024), and a disjoint MATH test set, with four-shot Minerva-style prompting (Lewkowycz et al., 2022) used for MATH. We further evaluate numerical robustness on GSM-Symbolic (Mirzadeh et al., 2025) and more challenging competition-level reasoning on Omni-MATH (Gao et al., 2025). HumanEval (Chen et al., 2021) evaluates code generation, and MMLU (Hendrycks et al., 2020), C-Eval (Huang et al., 2023), ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), and WinoGrande (Sakaguchi et al., 2021) cover knowledge and general reasoning. The evaluated problems in the main results have no problem-level overlap with the post-training corpus. We report accuracy for mathematical and multiple-choice tasks and pass@1 for HumanEval. The math average is the unweighted mean over GSM8K, Minerva-MATH, MATH-500, and the disjoint MATH set. More details in Appendix C.

Baselines. We compare CARVE with baselines that isolate the effects of supervised fine-tuning and reinforcement learning. These include full-corpus SFT, GRPO applied directly to the original backbone, and conventional shared-pool post-training with SFT followed by GRPO. We further include DAPO (Yu et al., 2026), which combines dynamic sampling with decoupled clipping, and GRESO (Zheng et al., 2026), which filters uninformative prompts before rollout. For each backbone, the post-SFT RL baselines share the SFT checkpoint and problem pool, with matched optimizer-update counts. Baseline configurations are detailed in Appendix B.4.

Implementation Details. We train Qwen2.5-Instruct backbones (Qwen et al., 2025). For CARVE, a source-stratified, problem-disjoint split assigns 4,382 records to SFT and 4,379 to RL. At 3B, eight-rollout outcome screening retains 2,166 RL problems. Full-parameter SFT runs for five epochs at learning rate 10^{-5} with a cosine decay schedule in LLaMA-Factory (Zheng et al., 2024). GRPO uses EasyR1 (Zheng et al., 2025b) on verl (Sheng et al., 2025), with four rollouts per prompt, 30 prompts per update, temperature 0.7, learning rate 10^{-6}, KL coefficient 0.04, and a 3,072-token response limit. The process model is trained on first-error Monte-Carlo step labels and admitted only when its held-out selector exceeds majority voting. Its reward docks correct rollouts beneath the correct-answer ceiling. The same allocation, optimization, and evaluation rules are applied across backbone sizes. More details are in Appendix B.

Training Corpus. We construct a CoT corpus from MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), BBH (Suzgun et al., 2023), DROP (Dua et al., 2019), AGIEval (Zhong et al., 2024), ARC-Challenge (Clark et al., 2018), MMLU-STEM (Hendrycks et al., 2020), and AIME 2024–2026 (Zhang & Math-AI, 2026). A solver and adversary generate and critique candidate derivations (Du et al., 2023). A three-vote judge supplies feedback before source-specific quality checks, exact gold-answer matching, and three-vote verification. The resulting 8,761 solution records cover 8,746 problems. Figure 2 summarizes source composition and derivation provenance. More details are in Appendix B.1.

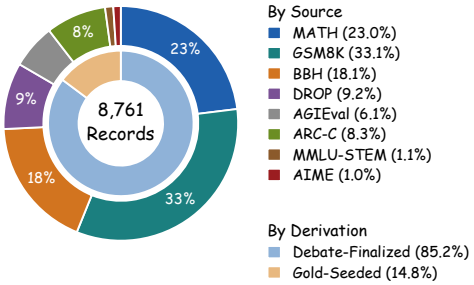

Figure 2: Source and derivation composition of the verified CoT corpus.

4.1 MAIN RESULTS

Table 1: Results across benchmarks at Qwen2.5-3B-Instruct and 7B. **Bold** and Underline denote the best and second best.

Method	RL Steps	Math Reasoning					Code	General Reasoning					
		GSM8K	Minerva	MATH-500	MATH	Avg.	Human Eval	MMLU	C-Eval	ARC-C	Hella Swag	Wino Grande	Avg.
Qwen2.5-3B-Instruct													
GRPO Without SFT	6.5k	33.1	51.8	59.6	58.4	50.7	52.4	65.0	72.9	49.3	74.9	69.7	66.4
SFT	–	60.5	29.8	42.8	41.1	43.5	**57.3**	62.1	68.6	**48.5**	71.8	64.4	63.1
SFT→GRPO	6.5k	45.5	27.9	41.0	40.5	38.7	54.9	62.2	69.0	48.3	72.0	65.8	63.4
SFT→DAPO	6.5k	60.4	39.3	46.4	44.3	47.6	54.9	62.8	69.5	47.1	72.8	66.7	63.8
SFT→GRESO	6.5k	60.5	37.7	44.8	44.4	46.9	54.3	62.6	69.3	46.9	72.5	66.5	63.6
CARVE (Ours)	1.4k	**64.0**	**40.8**	**47.8**	**46.6**	**49.8**	54.9	**63.3**	**69.9**	46.5	**73.5**	**67.5**	**64.1**
vs. SFT→GRPO	4.5× fewer	+18.5	+12.9	+6.8	+6.1	+11.1	0.0	+1.1	+0.9	−1.8	+1.5	+1.7	+0.7
Qwen2.5-7B-Instruct													
GRPO Without SFT	6.5k	79.2	68.5	73.2	70.4	72.8	70.7	73.8	78.0	54.4	78.1	71.8	71.2
SFT	–	81.0	66.0	71.2	69.7	72.0	**72.0**	73.2	77.3	**54.8**	77.8	71.3	70.9
SFT→GRPO	6.5k	77.3	63.5	70.0	67.4	69.5	69.5	72.8	76.9	53.4	77.6	71.2	70.4
SFT→DAPO	6.5k	81.7	67.8	71.8	70.1	72.8	70.7	73.1	77.5	54.1	77.9	71.4	70.8
SFT→GRESO	6.5k	81.4	67.3	71.4	69.5	72.4	70.1	73.0	77.4	53.9	77.7	71.5	70.7
CARVE (Ours)	1.4k	**83.2**	**69.2**	**73.6**	**71.6**	**74.4**	71.3	**74.0**	**78.1**	54.6	**78.2**	**71.9**	**71.4**
vs. SFT→GRPO	4.5× fewer	+5.9	+5.7	+3.6	+4.2	+4.9	+1.8	+1.2	+1.2	+1.2	+0.6	+0.7	+1.0

Table 1 exposes a consistent weakness of shared-pool post-training. Applying GRPO after SFT reduces the math average on both backbones, while DAPO and GRESO improve on the shared-pool GRPO result. In contrast, CARVE achieves the strongest math performance among SFT-based methods at both 3B and 7B, with improvements across all four math benchmarks. These gains require only 1.4k RL updates compared with 6.5k for the RL baselines, showing that better stage allocation improves both optimization efficiency and final performance. Average general reasoning is preserved or slightly improved, while code performance remains comparable to the other RL methods. The consistency across model scales supports separating imitation and exploration according to the post-SFT policy rather than repeatedly optimizing a shared problem pool.

4.2 Framework Analysis

Ablation Studies. Table 2 shows that data allocation provides the main initial gain, with both disjoint and difficulty-based splits clearly outperforming shared-pool training. Outcome screening then focuses RL on informative problems, reducing updates from 3.2k to 1.4k without sacrificing performance. Learned process supervision adds a further 2.8 points, whereas heuristic rewards degrade the screened model, consistent with the need for reliable process feedback. Removing outcome screening, group centering, or adaptive dock depth consistently weakens performance, supporting both informative data and controlled reward integration.

Table 2: Allocation and reward ablations at 3B. Δ is relative to shared-pool SFT→GRPO.

	Components			Math Reasoning					
Method	Screening	PRM	RL Steps	GSM8K	Minerva	MATH-500	MATH	Avg.	Δ
SFT→GRPO	✗	✗	6.5k	45.5	27.9	41.0	40.5	38.7	–
Disjoint Split	✗	✗	3.2k	60.7	37.4	43.7	43.9	46.4	+7.7
Difficulty Split	✗	✗	3.4k	**66.7**	35.6	44.8	43.2	47.6	+8.9
Disjoint Split + Screening	✓	✗	1.4k	60.9	38.2	44.1	44.7	47.0	+8.3
+ Heuristic PRM	✓	Heuristic	1.4k	51.6	34.7	42.5	41.8	42.7	+4.0
+ Learned PRM (CARVE)	✓	Learned	1.4k	64.0	**40.8**	**47.8**	**46.6**	**49.8**	+11.1
CARVE *w/o.* Outcome Screening	✗	Learned	3.2k	61.8	38.7	45.6	45.4	47.9	+9.2
CARVE *w/o.* Group Centering	✓	Learned	1.4k	62.5	39.0	46.0	45.6	48.3	+9.6
CARVE *w/o.* Adaptive Dock Depth	✓	Learned	1.4k	62.6	39.3	46.6	46.0	48.6	+9.9

Sensitivity And Reward Variance. Figure 3 shows clear interior optima for KL regularization, mixed-group dock depth, and maximum dock strength, indicating that both regularization and process feedback require moderate strength. The zero-variance rate decreases progressively from 92% under shared-pool training to 62% after disjoint allocation, 58% after screening, and 34% with the full method. This progression reflects two complementary effects: allocation preserves informative outcome variation, while process docking restores relative signal in groups that remain indistinguishable under outcome rewards alone. More details are in Tables 12 and 13.

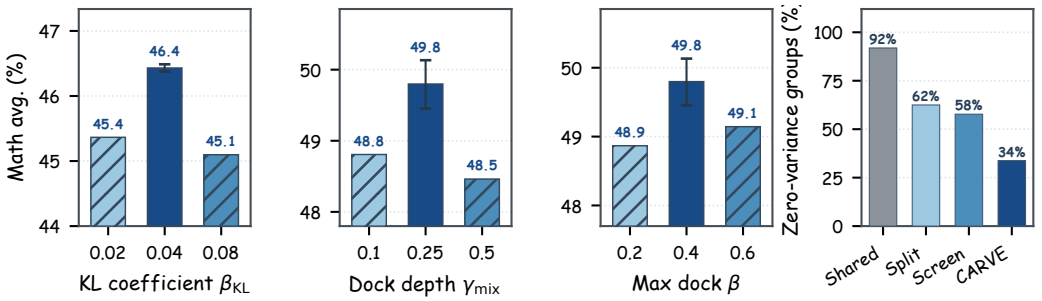

Figure 3: KL sensitivity for disjoint outcome-only RL and process-dock sensitivity for CARVE (first three panels), with zero-reward-variance groups (right). Hatching marks endpoint settings; error bars show repeat ranges.

Scaling Behavior. Figure 4 shows that CARVE consistently improves over its SFT parent from 3B to 14B, showing that the gain persists across model capacities rather than being confined to smaller models. Performance also increases steadily with additional RL updates for all three backbones, indicating that the screened data continues to support useful optimization as the training budget grows. In contrast, performance does not increase monotonically with the retained RL pool and instead peaks at an intermediate size. Together, these results distinguish optimization scaling from data scaling. Additional RL updates remain beneficial, whereas enlarging the training pool beyond policy-informative problems can dilute the learning signal. More details are provided in Table 11.

Group-Level Reward Variation. Table 3 shows that screening and dock replay recover reward variation through complementary mechanisms. Screening increases the share of mixed-correctness

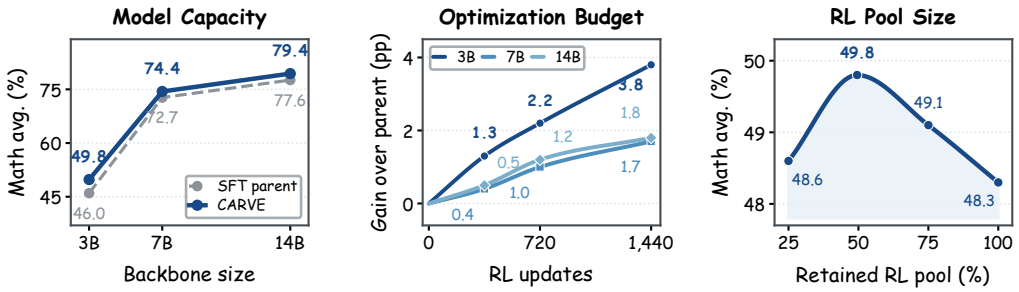

Figure 4: Scaling across backbone size, RL updates, and retained RL pool size.

Table 3: Fixed-parent group-type diagnostic at 3B. Dock replay uses identical screened rollouts.

			Correctness Pattern (%)			Zero Variance (%)	
Pool	Reward	Groups	All Wrong	Mixed	All Correct	Overall	Within All-Correct
Candidate RL pool	Outcome	4,373	53.9	19.4	26.7	63.1	84.6
Screened band	Outcome	2,166	42.3	27.6	30.1	56.7	89.0
Screened band	Dock replay	2,166	42.3	27.6	30.1	**37.2**	**24.1**

groups by 8.2 points, but leaves all-correct groups largely uninformative. On the same screened rollouts, dock replay reduces the all-correct zero-variance rate by 64.9 points without changing correctness patterns. All-wrong groups remain unchanged because docking never increases rewards for incorrect responses. Together, these results support the complementary signal pathways in Theorem 1: screening preserves outcome variation, while docking differentiates correct derivations.

Scoring Protocol Sensitivity. Table 4 shows that GSM8K is more sensitive to answer extraction than MATH. At 3B, strict matching favors shared-pool GRPO, whereas flexible extraction favors CARVE; at 7B, CARVE leads under both rules. In contrast, the MATH ordering remains unchanged under exact and symbolic scoring at both scales, with CARVE consistently strongest. The overall advantage therefore persists across scoring protocols, while the 3B GSM8K exception highlights the need to report extraction-sensitive results. More details are in Table 10.

Table 4: Answer-scoring sensitivity.

Model Scale	Method	GSM8K		MATH (4-shot)	
		Strict	Flexible	Exact	Symbolic
3B	GRPO	**49.7**	45.5	18.2	27.9
	DAPO	47.5	60.4	24.4	39.3
	CARVE	44.1	**64.0**	**25.9**	**40.8**
7B	GRPO	49.2	77.3	37.4	63.5
	DAPO	47.4	81.7	40.2	67.8
	CARVE	**50.6**	**83.2**	**41.4**	**69.2**

Table 5: Cross-benchmark accuracy.

	Accuracy (%)	
Method	GSM-Symbolic	Omni-MATH
Qwen2.5-3B-Instruct		
SFT→DAPO	45.1	11.8
CARVE	**48.2**	**13.2**
Qwen2.5-7B-Instruct		
SFT→DAPO	70.4	23.0
CARVE	**72.1**	**24.4**

Cross-Benchmark Generalization. Table 5 shows that CARVE consistently outperforms DAPO on both GSM-Symbolic and Omni-MATH at 3B and 7B. The gains extend to numerical perturbations as well as harder competition-level problems, suggesting that the improvement is not confined to the original evaluation suite. The consistent ordering across model scales further supports the transferability of the learned reasoning policy. More details are provided in Appendix D.1.

Selection Under Same Pool Size. Table 6 isolates the effect of pool selection by holding pool size, source composition, and optimizer budget fixed under outcome-only RL. Informative-band selection achieves the highest math average, exceeding random selection by 1.4 points. Highest- and lowest-solve-rate rules favor different benchmarks but remain weaker overall, showing that difficulty extremes do not provide a consistently useful RL curriculum. With process supervision

excluded, the gain supports selecting problems with residual outcome variation rather than simply easier or harder examples. More details are provided in Appendix D.2.

Table 6: Pool selection at 3B with 2,166 problems and outcome-only RL.

Selection Rule	GSM8K	Minerva	MATH-500	MATH	Avg.
Random Within Source	59.5	36.6	43.0	43.3	45.6
Highest Solve Rate	**61.1**	34.8	40.8	41.2	44.5
Lowest Solve Rate	58.1	**38.5**	43.8	42.9	45.8
Informative Band	60.9	38.2	**44.1**	**44.7**	**47.0**

Process Credit Assignment. Figure 5 shows a clear progression from outcome-only RL to additive process rewards, uniform docking, and adaptive docking. Additive PRM provides only a modest gain, while dock-only integration improves further by adjusting relative credit without broadly reshaping the outcome reward. Adaptive docking performs best, supporting group-dependent control of process feedback. A permutation control in Appendix D.3 reduces performance to near outcome-only RL, confirming that the gain depends on assigning deductions to the appropriate derivations. These results support the targeted credit mechanism in Theorem 2.

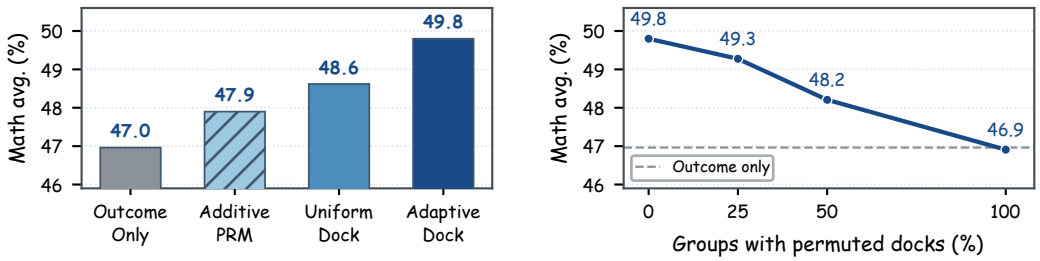

Figure 5: Reward integration (left) and permutation of process docks (right) at 3B.

Sampling Efficiency and Group Size. Figure 6 shows that CARVE remains stronger across sampling budgets and group sizes. Including screening and Monte-Carlo labeling, it exceeds both baselines at twice its 240-million-token budget. At fixed response count, larger groups initially help outcome-only RL, while CARVE peaks at a moderate group size. This pattern supports more effective response allocation and process credit. More details are in Appendix D.4.

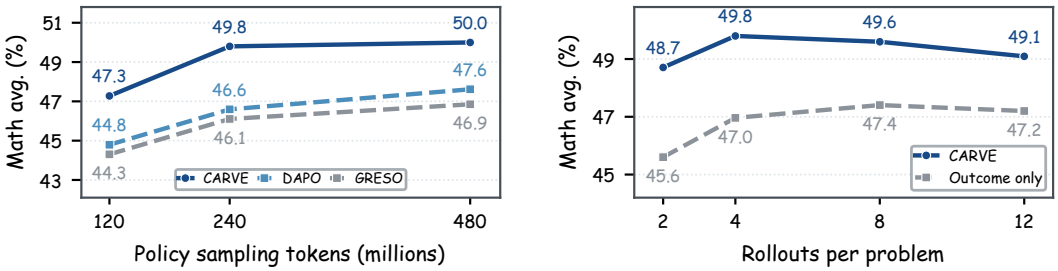

Figure 6: Sampling budget including preparation (left) and response allocation (right) at 3B.

5 CONCLUSION

We identify *variance starvation* that supervised fine-tuning can reduce the reward variation needed by group-relative reinforcement learning. CARVE addresses this coupling through variance-aware data allocation and a gated dock-only process reward that restores relative signal without rewarding incorrect responses. Extensive experiments across ten benchmarks and multiple backbone scales show that CARVE improves math accuracy over shared-pool post-training baselines, maintains comparable code and general-reasoning performance, and uses half the SFT data and fewer RL updates.

AI USE STATEMENT

In this work, Generative AI tools assisted with training-corpus construction and PRM step labeling, code revisions, manuscript language polishing, and mathematical notation checking in the proof. We carefully reviewed all AI-assisted content and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Section 3 specifies every component of CARVE, and Appendices B.3 and B.5 give the full reward definition and all hyperparameters. Appendices B.1 and C describe corpus construction and evaluation protocols.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, 2026.

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.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *Advances in Neural Information Processing Systems*, 38:131646–131671, 2026.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325*, 2023.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, 2019.

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 102430–102459, 2026.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Zhengyang Tang, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 100540–100569, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International conference on machine learning*, pp. 10835–10866. PMLR, 2023.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Chongyang He, Rui Zhang, Zixuan Wang, and Xin Li. Learning what to learn: Stage-specific data sets for sft-then-rl in small language model reasoning. *arXiv preprint arXiv:2606.04466*, 2026.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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.

Hanxu Hu, Yuxuan Wang, Maggie Huan, Jannis Vamvas, Yinya Huang, Zhijiang Guo, and Rico Sennrich. Dereason: A difficulty-aware curriculum improves decoupled sft-then-rl training for general reasoning. *arXiv preprint arXiv:2603.11193*, 2026.

Yuzhen Huang, Yuzhuo Bai, Zhihao Zhu, Junlei Zhang, Jinghan Zhang, Tangjun Su, Junteng Liu, Chuancheng Lv, Yikai Zhang, Yao Fu, et al. C-eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *Advances in neural information processing systems*, 36:62991–63010, 2023.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Feiyang Kang, Michael Kuchnik, Karthik Padthe, Marin Vlastelica, Ruoxi Jia, Carole-Jean Wu, and Newsha Ardalani. Quagmires in sft-rl post-training: When high sft scores mislead and what to use instead. In *International Conference on Learning Representations*, volume 2026, pp. 56876–56918, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pp. 5315–5333, 2023.

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 *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Alexis Limozin, Eduard Durech, Torsten Hoefler, Imanol Schlag, and Valentina Pyatkin. Sft-then-rl outperforms mixed-policy methods for llm reasoning. *arXiv preprint arXiv:2604.23747*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Lu Ma, Hao Liang, Meiyi Qiang, Lexiang Tang, Xiaochen Ma, Zhen Wong, Junbo Niu, Chengyu Shen, Runming He, Yanhao Li, et al. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *International Conference on Learning Representations*, volume 2026, pp. 80802–80821, 2026.

Iman Mirzadeh, Keivan Alizadeh-Vahid, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *International Conference on Learning Representations*, volume 2025, pp. 94743–94765, 2025.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

John Schulman. Approximating KL divergence. http://joschu.net/blog/kl-approx.html, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, 2023.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *Advances in Neural Information Processing Systems*, 38:117157–117186, 2026.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2026, 2026.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, 2025.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, 2025a.

Haizhong Zheng, Yang Zhou, Brian Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *Advances in Neural Information Processing Systems*, 38:124321–124346, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 3: system demonstrations)*, pp. 400–410, 2024.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, Yuwen Xiong, and Richong Zhang. Easyr1: An efficient, scalable, multi-modality rl training framework, 2025b.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. In *Findings of the association for computational linguistics: NAACL 2024*, pp. 2299–2314, 2024.

Roadmap. Appendix A presents the theoretical assumptions and proofs. Appendix B details corpus construction, data allocation, training and reward configurations, and PRM validation. Appendix C describes the evaluation benchmarks and metrics. Appendix D reports supplementary analyses of scoring sensitivity, group reward variation, RL pool size, cross-benchmark generalization, pool selection, process credit assignment, and sampling efficiency.

A Theoretical Analysis

A.1 Complementary Signal Enrichment

Assumptions. Let P be the distribution of fixed-parent solve rates $p \in [0,1]$ across RL problems. Screening draws $H \mid p \sim \mathrm{Bin}(n,p)$ with $n \geq 2$ and retains a problem when $1 \leq H \leq n-1$. A subsequent $G \geq 2$ rollout group, including its correctness labels and process scores, is independent of H given p; its correctness labels are conditionally independent Bernoulli draws with rate p. We require $Z = \mathbb{E}_P[S_n(p)] > 0$, where $S_k(p) = 1 - p^k - (1-p)^k$. In this binary-outcome abstraction, incorrect rewards are constant within an all-incorrect group and at most c in a mixed group. Format penalties equal $\kappa \geq 0$. The parameters satisfy $\beta > 0$, $w_b \geq 0$, $0 \leq \alpha, \gamma_{\mathrm{mix}} \leq 1$, and $c < R - \beta - w_b - \kappa$. The break indicator satisfies $b_i \in \{0,1\}$, depth satisfies $d_i \in [0,1]$, and centered quality satisfies $\tilde{q}_i \in [0,1]$. In an all-correct group, $b_i = 0$ and every rollout has the same depth d, with $a = 1 - \alpha + \alpha d > 0$. Define $\eta(p) = \Pr[\mathrm{Var}_g(\tilde{q}) > 0 \mid p, C = G]$, setting $\eta(0) = 0$ where the conditioning event has zero probability.

Proof Of Theorem 1. Fix p and let C count correct rollouts in the subsequent group. A mixed group occurs with probability $S_G(p)$, because the only excluded correctness patterns are $C = 0$ and $C = G$. For a correct rollout, put $a_i = 1 - \alpha + \alpha d_i$ and let its score before reward clipping and the common penalty be

$$u_i = R - \beta\gamma_g(1 - a_i\tilde{q}_i) - w_b b_i. \tag{10}$$

The assumptions give $0 \leq a_i, \gamma_g, \tilde{q}_i, b_i \leq 1$, hence $R - \beta - w_b \leq u_i \leq R$. Since $c < R - \beta - w_b - \kappa$, the lower clipping at c does not change u_i. Since $\kappa \geq 0$, the upper clipping at R does not change $u_i - \kappa$. Every correct reward is therefore $u_i - \kappa > c$, whereas each incorrect reward is at most c. Every mixed group consequently has positive reward variance.

When $C = G$, the common depth and absence of break penalties give $\gamma_g = 1$, $a_i = a > 0$, and

$$r_i = R - \beta - \kappa + \beta a\tilde{q}_i, \qquad \mathrm{Var}_g(r) = \beta^2 a^2 \,\mathrm{Var}_g(\tilde{q}). \tag{11}$$

Thus an all-correct group has positive reward variance with conditional probability $\eta(p)$. An all-incorrect group has zero variance by the constant-reward assumption. These three disjoint cases yield

$$\Pr[\mathrm{Var}_g(r) > 0 \mid p] = S_G(p) + p^G\eta(p). \tag{12}$$

Conditional independence of screening and the subsequent rollout group gives $\Pr(1 \leq H \leq n-1 \mid p) = S_n(p)$ and

$$I = \frac{\mathbb{E}_P[S_n(p)S_G(p)]}{Z} + \frac{\mathbb{E}_P[S_n(p)p^G\eta(p)]}{Z}. \tag{13}$$

By the definition of covariance, $\mathbb{E}_P[S_nS_G] = Z\mathbb{E}_P[S_G] + \mathrm{Cov}_P(S_n, S_G)$. Substituting gives Eq. 8.

It remains to establish the sign of the covariance. For every integer $k \geq 2$, $S_k(p) = S_k(1-p)$, and on $0 \leq p \leq \frac{1}{2}$ its derivative is

$$\frac{\mathrm{d}S_k}{\mathrm{d}p} = k\big((1-p)^{k-1} - p^{k-1}\big) \geq 0. \tag{14}$$

The derivative is positive for $p < \frac{1}{2}$. Since $u = p(1-p)$ is strictly increasing on $[0, \frac{1}{2}]$, symmetry makes S_k a strictly increasing function $f_k(u)$ on $u \in [0, \frac{1}{4}]$. Let $U = p(1-p)$ for $p \sim P$ and let U' be an independent copy. The covariance identity

$$2\,\mathrm{Cov}_P(S_n, S_G) = \mathbb{E}\big[(f_n(U) - f_n(U'))(f_G(U) - f_G(U'))\big] \tag{15}$$

has a nonnegative integrand. It is positive with positive probability if U is nonconstant, making the covariance strictly positive in that case. The final term of Eq. 8 is nonnegative because $S_n(p)$, p^G, and $\eta(p)$ are nonnegative. This proves both the decomposition and its sign claims. □

The reference term $\mathbb{E}_P[S_G(p)]$ is the informative-group probability for an unscreened binary-outcome reward. The covariance term measures enrichment by screening within the fixed-parent pool, and the final term counts all-correct groups distinguished by docking after screening. The decomposition evaluates the fixed-parent distribution under the stated reward assumptions. The quantity $\eta(p)$ measures within-group score variation separately from held-out selector accuracy.

When $C = G$ and $\mathrm{Var}_g(\tilde{q}) > 0$, Eq. 1 and Eq. 11 also imply

$$\hat{A}_i = \frac{\beta a(\tilde{q}_i - \bar{\tilde{q}})}{\beta a \,\mathrm{std}_g(\tilde{q}) + \epsilon}. \tag{16}$$

At $\epsilon = 0$, this standardized advantage is independent of β. With $\epsilon > 0$, β changes its magnitude.

A.2 Correct-Answer Advantage Under Docking

Assumptions And Notation. Fix a group of $G \geq 2$ rollouts containing k correct answers, with $1 \leq k < G$. Let $\mathcal{C}_g$ be the correct subset. Assume $R > c$, $\beta, w_b \geq 0$, $0 \leq \gamma_{\mathrm{mix}}, \alpha, d_i, \tilde{q}_i, b_i \leq 1$, and $D_g = \beta\gamma_{\mathrm{mix}} + w_b$. All rollouts receive the same nonnegative format penalty κ. Incorrect rewards are at most c after capping, and correct rewards follow Eq. 6. Write $\bar{r} = G^{-1}\sum_{j=1}^{G} r_j$ and $s_g = \mathrm{std}(\{r_j\}_{j=1}^{G})$. The normalization in Eq. 1 uses $\epsilon > 0$.

Proof Of Theorem 2. Put $a_i = 1 - \alpha + \alpha d_i$. The assumed bounds give $0 \leq a_i\tilde{q}_i \leq 1$. A correct rollout's reward before clipping and the common penalty is therefore

$$u_i = R - \beta\gamma_{\mathrm{mix}}(1 - a_i\tilde{q}_i) - w_b b_i \in [R - D_g, R].$$

Since $c < R$ and $u_i \leq R$, the inner clipping $\max(c, u_i)$ is at most R and at least u_i. Subtracting $\kappa \geq 0$ cannot activate the outer clipping at R. Thus every correct reward lies in $[R - D_g - \kappa, R - \kappa]$, while every incorrect reward is at most c.

Fix any $i \in \mathcal{C}_g$. The centered reward can be written exactly as

$$r_i - \bar{r} = \frac{(G-1)r_i - \sum_{j \neq i} r_j}{G}.$$

There are $k - 1$ other correct rollouts, each with reward at most $R - \kappa$, and $G - k$ incorrect rollouts, each with reward at most c. Applying these upper bounds and $r_i \geq R - D_g - \kappa$ gives

$$\begin{aligned} G(r_i - \bar{r}) &\geq (G-1)(R - D_g - \kappa) - (k-1)(R - \kappa) - (G-k)c \\ &= (G-k)(R - c - \kappa) - (G-1)D_g. \end{aligned}$$

Division by $G(s_g + \epsilon) > 0$ proves Eq. 9. Its numerator is strictly positive under the stated condition on D_g, proving the sign claim for every correct rollout. At equality, the bound certifies nonnegative advantages. The all-correct case is excluded because centered advantages sum to zero, so they cannot all be strictly positive. □

Implication For The Dock Schedule. With $G = 4$, $R = 2$, $c = 0.55$, and $\kappa = 0$, the most restrictive mixed group has $k = 3$ and requires $D_g < 1.45/3$. The default $\beta = 0.4$, $\gamma_{\mathrm{mix}} = 0.25$, and $w_b = 0.15$ give $D_g = 0.25$, satisfying this condition. Applying the full dock to mixed groups gives $D_g = 0.55$, outside the sufficient bound.

Unequal Format Penalties. If penalties vary within a group, let $\kappa_{\min} = \min_j \kappa_j$ and $\Delta_\kappa = \max_j \kappa_j - \kappa_{\min}$. The same reward bounds give

$$G(r_i - \bar{r}) \geq (G-k)(R-c) - (G-1)D_g - (G-1)\kappa_i + \sum_{j \in \mathcal{C}_g \setminus \{i\}} \kappa_j.$$

Since $\kappa_i \leq \kappa_{\min} + \Delta_\kappa$ and every other correct penalty is at least $\kappa_{\min}$, it follows that

$$\hat{A}_i \geq \frac{(G-k)(R - c - \kappa_{\min}) - (G-1)(D_g + \Delta_\kappa)}{G(s_g + \epsilon)}.$$

At $\Delta_\kappa = 0$, this recovers Eq. 9.

B Implementation Details

B.1 The Verified CoT Corpus

Scope. The corpus supplies the demonstrations consumed by the SFT stage and the prompts consumed by the RL stage. It holds 8,746 problems and 8,761 solution records from eight sources: MATH, GSM8K, BBH, DROP, AGIEval, ARC-Challenge, MMLU-STEM, and AIME 2024–2026. Each record carries the problem, the gold answer, one derivation, and its provenance: the number of adversarial rounds, the judge confidence, the verifier confidence, and whether the gold answer was supplied during generation. These fields make the difficulty split of Section 3 definable without extra annotation.

Construction. For each problem, a solver and an adversarial model generate independent solutions and exchange critiques through multi-agent debate (Du et al., 2023). A judge (Zheng et al., 2023) votes three times on the stronger solution and returns its verdict to both models as feedback. The solver's finalized derivation undergoes source-specific checks of length, explicit reasoning, boxed answers, and debate artifacts. The final pipeline requires an exact gold-answer match and unanimous correct and acceptable judgments from three independent verifier calls, with mean confidence at least 0.97 and no reported issues. After the retry limit, a fallback generator receives the gold answer and produces candidates subject to the same checks. These records are marked *gold-seeded*.

Generation Models And Prompt Templates. The standard roles use `claude-opus-4-8` for the solver and adversary ($T = 0.7$), `gpt-5.5` for the three-vote judge ($T = 0.3$), and `gemini-3.5-flash` for three verifier calls ($T = 0.1$). Finalization uses $T = 0.2$. The [DOMAIN] field names the source domain, and [ANSWER] is a boxed value or choice letter. Each verifier receives the problem, gold answer, and candidate. The compact templates retain the role-defining instructions and input fields.

> **Solver system:** "You are an expert problem-solver specializing in [DOMAIN]. Produce a correct, complete solution. State every nontrivial step, track constraints and boundary cases, and end in \boxed{ANSWER}. Keep the final writeup self-contained and free of debate artifacts."
>
> **Adversary system:** "Produce a plausible but incorrect solution for [DOMAIN], with one subtle core flaw and a wrong final answer. Do not disclose the flaw or create training labels. End in \boxed{ANSWER}."
>
> **Shared generation user message:** "Solve the following problem with detailed step-by-step reasoning. Show all work and end with the final answer in \boxed{ANSWER}. Problem: [PROBLEM]"
>
> **Critique user message:** "Problem: [PROBLEM]. Your solution: [OWN SOLUTION]. Opponent's solution: [OTHER SOLUTION]. Argue why your solution is superior and point out specific flaws in the opponent's reasoning. Be technical. Keep it concise."
>
> **Judge system:** "Compare solutions A and B for correctness and rigor. Trace both arguments, check assumptions and missing cases, and choose the more reliable explanation. Return BOTH_BAD or UNCLEAR when appropriate. Respond only with JSON fields selected, confidence, and reasoning."
>
> **Finalization user message:** "Rewrite the candidate as a clean supervised solution. Preserve every nontrivial step, remove debate artifacts and repetition, and end in \boxed{ANSWER}. Problem: [PROBLEM]. Candidate: [DRAFT]. Judge feedback: [FEEDBACK]. Return only the solution."
>
> **Verifier system:** "Accept only when the final answer matches the gold answer and the reasoning is sound and complete enough for training. Reject false inferences, unsupported jumps, and debate artifacts. Return JSON fields answer_correct, acceptable, confidence, final_answer, issues, and reasoning."
>
> **Gold-seeded fallback:** "The final answer is [GOLD]. Derive it rigorously, show every nontrivial step, track constraints, and end with \boxed{[GOLD]}." The candidate undergoes the same finalization and verification.

Composition. Table 7 gives source-level statistics, while Figure 2 shows the composition. Competition-style sources produce the longest derivations, with a median of 966 tokens on AIME

Table 7: Verified CoT corpus by source, with record counts, generation provenance, derivation lengths, and SFT/RL allocation.

	Size		Gold-Seeded	Provenance		CoT Length		Split	
Source	Problems	Records	(%)	Debate Rounds	Judge Margin	Median	P90	SFT	RL
MATH	2,019	2,019	15.2	1.28	0.762	401	773	1,010	1,009
GSM8K	2,898	2,898	9.9	1.43	0.808	194	303	1,449	1,449
BBH	1,586	1,586	17.8	1.31	0.735	250	412	793	793
DROP	793	804	24.3	1.24	0.686	167	291	402	402
AGIEval	537	537	21.0	1.28	0.706	370	659	268	269
ARC-C	728	728	15.1	1.38	0.775	271	386	364	364
MMLU-STEM	95	99	1.0	1.23	0.856	329	500	51	48
AIME	90	90	1.1	1.99	0.929	966	1387	45	45
All	8,746	8,761	14.8	1.35	0.766	250	525	4,382	4,379

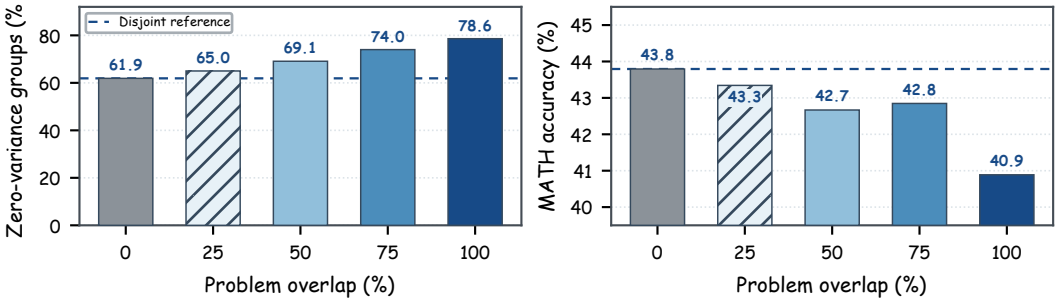

Figure 7: Fixed-parent SFT/RL problem-overlap sweep at 3B. The RL pool size and outcome-only optimization budget are held constant. Zero-variance rates average over logged batches.

against 194 on GSM8K, and require the most debate rounds. Gold-seeded records make up 14.8% of the corpus and concentrate in the sources with lower debate acceptance rates. The problem-level split assigns 4,382 records to SFT and 4,379 to RL with no shared problem, and outcome screening retains 2,166 RL problems. AIME 2024–2026 contributes to training and is excluded from held-out evaluation.

B.2 Data Allocation And Screening

Half of each source is assigned to SFT by a hash of the problem text, giving 4,382 SFT records over 4,373 problems and 4,379 RL records over 4,373 problems. Screening draws $n = 8$ solutions per RL problem at temperature 0.7 under the SFT parent and keeps the informative band $1 \leq h(x) \leq 7$, which retains 2,166 problems. The difficulty split of Table 2 replaces the hash with the verifier confidence of each record, sending the hardest half to SFT.

Controlled Pool Overlap. The overlap stress test keeps the 3B SFT parent trained on the 4,373-problem SFT partition fixed. It forms five RL pools with 0, 1,093, 2,187, 3,280, or 4,373 SFT-exposed problems among 4,373 unique RL problems, replacing unseen problems within source and pre-SFT difficulty strata. Every pool contains 4,379 records over 4,373 unique problems. Each configuration uses outcome-only GRPO for 3,240 optimizer updates, with no outcome screening or PRM docking. The zero-overlap endpoint uses the disjoint-split configuration of Table 2.

Overlap And Group Variance. Figure 7 shows that full SFT/RL overlap raises zero-variance groups by 16.7 points and lowers MATH accuracy by 2.9 points relative to a disjoint pool. This trend is consistent with imitation reducing the outcome variation available to RL.

B.3 Training Configuration

SFT is full-parameter and bf16 in LLaMA-Factory. GRPO runs in EasyR1 on verl with the token-averaged clipped objective of Eq. 7, asymmetric clipping bounds $\varepsilon_{\text{low}} = 0.2$ and $\varepsilon_{\text{high}} = 0.3$, and the

low-variance KL estimator toward the SFT parent. The controlled GRPO variants share the prompt template and default seed. Checkpoints use the update budgets in Tables 1 and 2. Algorithm-specific sampling and clipping settings are described in Appendix B.4. Table 8 lists the common settings.

Table 8: Training and reward settings shared across the reported experiments.

Stage	Setting	Value
All stages	hardware	8×A100 (80GB)
SFT	framework / type	LLaMA-Factory, full parameter, bf16
	epochs / learning rate / schedule	5 / 1×10^{-5} / cosine, 5% warm-up
	per-device batch / grad. accumulation	1 / 4
	weight decay / max grad norm / cutoff	0.01 / 1.0 / 8,192 tokens
	template / seed	none (plain continuation) / 42
GRPO	framework	EasyR1 (verl)
	rollouts per prompt / temperature	4 / 0.7
	prompts per step / actor global batch	30 / 30
	learning rate / maximum epochs over pool	1×10^{-6} / 30
	clipping ε_{low} / $\varepsilon_{\text{high}}$	0.2 / 0.3
	KL loss / coefficient β_{KL}	low-variance KL / 0.04
	max prompt / response length	1,024 / 3,072 tokens
Screening	samples n / band / temperature	8 / $[1, 7]$ / 0.7
PRM labels	strategy / MC samples M / threshold τ	first-error / 16 / 0.5
PRM training	loss / epochs / learning rate	BCE + ranking (δ_ℓ=0.25, λ=1) / 2 / 1×10^{-5}
	batch / grad. accumulation / class-balance cap	4 / 4 / 10
PRM reward	ceiling R / max dock β / break penalty w_b	2 / 0.40 / 0.15
	incorrect ceiling c / mixed-group γ_{mix}	0.55 / 0.25
	max steps / depth weight α / length K_0 / break threshold	20 / 0.4 / 5 / 0.45
Evaluation	context / Minerva gen. limit / AIME gen. limit	4,096 / 1,024 / 3,072 tokens

B.4 Baseline Configurations

The SFT control applies the full-corpus recipe and is evaluated before RL. GRPO without SFT starts from the original Qwen2.5-Instruct checkpoint and trains on the full corpus problem pool with the outcome reward. Shared-pool SFT→GRPO starts from the full-corpus SFT checkpoint and reuses the same pool for outcome-only RL. DAPO and GRESO retain that SFT initialization and candidate pool: DAPO changes group sampling and clipping, whereas GRESO filters prompts before rollout. All methods draw from the same CoT corpus. The distinct SFT parent of CARVE follows from its problem-disjoint allocation.

B.5 Full Reward Definition

Let $\mathbf{1}[\text{correct}]$ be answer correctness under symbolic normalization of the final answer. The outcome backbone, used for all rollouts in outcome-only runs and for incorrect rollouts in PRM runs, is

$$r^{\text{out}} = 2\,\mathbf{1}[\text{correct}] + 0.20\,u_{\text{prox}} + 0.12\,u_{\text{reason}} + 0.08\,u_{\text{fmt}} + 0.03\,u_{\text{len}} - \kappa, \tag{17}$$

where $u_{\text{prox}} \in [-1, 1]$ rewards closeness of a wrong answer, $u_{\text{reason}}, u_{\text{fmt}} \in [0, 1]$ score explicit reasoning and answer formatting, $u_{\text{len}} \in [-1, 1]$ is a soft length score whose target is 18% of the response limit, clipped to $[600, 2200]$ tokens, and κ sums penalties for a missing boxed answer (0.45), an unparseable answer (0.30), extra boxed answers (0.10), truncation (0.30), and debate artifacts (0.18). For math sources, an incorrect rollout is capped at c (0.40 for outcome-only runs, 0.55 for PRM runs) and at 0.20 if its answer is numerically far from the gold. All rewards are clipped to $[-1.5, 2]$. The process term of Eq. 6 applies only to math sources (MATH, GSM8K, AIME, AGIEval) with numeric or expression answers. Other sources always use r^{out}. The heuristic process signal in Table 2 computes q and d from the same segmentation with structural and numeric checks rather than the learned PRM.

B.6 Process Reward Model And Gate

Step Labels And Model Training. Step labels use $M = 16$ Monte-Carlo continuations at threshold $\tau = 0.5$, with at most 20 steps per solution and first-error binary search on incorrect solutions.

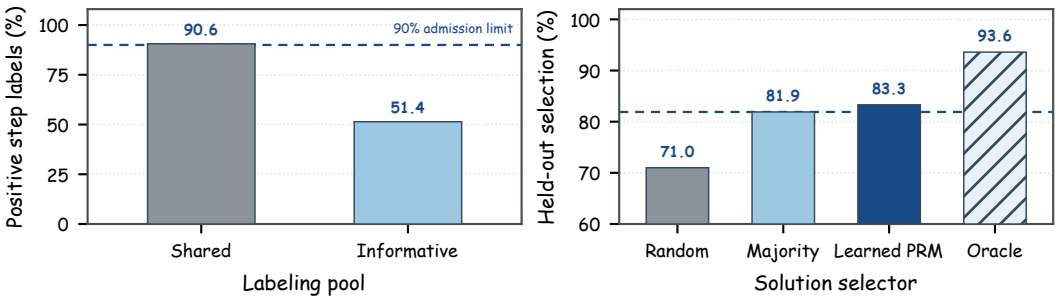

Figure 8: PRM admission at 3B: process-label balance by pool and solution selection on held-out problems. Dashed lines mark the label-rate limit (left) and majority-vote accuracy (right).

The informative band contributes 1,025 problems and 4,100 solutions for judge review, yielding 48,581 step labels. The judge is `gpt-5.5` at temperature zero. Its system instruction asks for the first mathematically or logically incorrect step, returning its one-based index or zero if every step is correct. The user message supplies the problem, gold answer, and numbered solution steps. When the judge and Monte-Carlo first-error indices disagree, the judged index determines the step labels; a judged all-correct solution with an incorrect final answer is discarded. Training uses the class-balanced BCE and pairwise terms of Eq. 5 with label-gap threshold $\delta_\ell = 0.25$ and weight $\lambda = 1$.

Held-Out Selector Validation. Figure 8 shows that informative-band labeling satisfies the label-balance gate and improves held-out solution selection. Its positive-label rate is 51.4% (61% on GSM8K and 44% on MATH), within the admissible interval $[15\%, 90\%]$, compared with 90.6% for shared-pool labeling. On unseen problems, mean, final-step, and minimum step scores are combined with best-of-N, score-weighted voting, or veto-then-majority selection. One half of the problem set selects the configuration and the other measures its accuracy. Final-step score-weighted voting reaches 83.3%, exceeding majority voting at 81.9% and satisfying the selector criterion.

C Evaluation Benchmarks

Table 9 lists the evaluated benchmarks, splits, sizes, prompts, and metrics.

Benchmark Descriptions. GSM8K (Cobbe et al., 2021) contains grade-school word problems with a numerical answer. Minerva-MATH denotes the four-shot evaluation of the 5,000-question Hendrycks MATH test across seven competition-math subjects with symbolic equivalence (Hendrycks et al., 2021; Lewkowycz et al., 2022). MATH-500 (Lightman et al., 2024) contains 500 test questions. The disjoint MATH set removes 495 exact problem-text matches to MATH-500 from the 4,934-question local MATH test collection, leaving 4,439 questions. GSM-Symbolic (Mirzadeh et al., 2025) supplies 50 numerical instantiations of 100 GSM8K test templates. Omni-MATH (Gao et al., 2025) consists of 4,428 olympiad-level questions from varied mathematical domains. AIME 2024–2026 contributes 30 problems per contest to the training corpus and is excluded from evaluation. HumanEval (Chen et al., 2021) tests Python synthesis by running unit tests for 164 functions. For broader capability, MMLU (Hendrycks et al., 2020) covers 57 knowledge subjects, C-Eval (Huang et al., 2023) covers 52 Chinese-language disciplines, ARC-Challenge (Clark et al., 2018) tests grade-school science, HellaSwag (Zellers et al., 2019) tests plausible event continuations, and WinoGrande (Sakaguchi et al., 2021) tests pronoun resolution.

Metrics And Evaluation Protocol. Let $\hat{a}_i$ be the extracted answer, $a_i^\star$ the reference answer, and $\simeq$ the benchmark's matching rule. For benchmark b with N_b evaluated items, answer accuracy and the unweighted four-task math average are

$$\mathrm{Acc}_b = \frac{100}{N_b} \sum_{i=1}^{N_b} \mathbf{1}[\hat{a}_i \simeq a_i^\star], \qquad \mathrm{MathAvg} = \frac{1}{4} \sum_{b \in \mathcal{B}_{\mathrm{math}}} \mathrm{Acc}_b. \tag{18}$$

Here $\mathcal{B}_{\text{math}}$ contains GSM8K, Minerva-MATH, MATH-500, and the disjoint MATH set. The Harness evaluation (v0.4.12) uses five-shot flexible extraction on GSM8K and four-shot symbolic verification on the full MATH test. The alternative scorers apply strict template extraction on GSM8K and exact matching on MATH to the same responses. MATH-500 and the disjoint MATH set use one greedy response per problem with symbolic answer matching. GSM-Symbolic uses the same five-shot numerical extractor as GSM8K, and Omni-MATH uses one greedy response with the benchmark's GPT-4o equivalence judge. Both are separate from the four-task math average. HumanEval uses test-based pass@1, and the multiple-choice tasks use zero-shot accuracy, length-normalized for ARC-Challenge and HellaSwag. Every evaluation uses a 4,096-token context. The four-task math average combines task and scoring variation because Minerva-MATH overlaps the two other MATH sets. Training-visible AIME problems are excluded from the main comparison. The alternative scorer results appear in Table 10.

Table 9: Evaluated datasets, split sizes, in-context examples, and scoring rules.

Benchmark	Domain	Split	Items	Shots	Metric
Math Reasoning					
GSM8K	Grade-School Math	Test	1,319	5	Exact Match (Flexible Extract)
Minerva-MATH	Competition Math	Test	5,000	4	Symbolic Equivalence
MATH-500	Competition Math	Test Subset	500	0	Greedy Pass@1
MATH	Competition Math	Local Test Without MATH-500	4,439	0	Greedy Pass@1
GSM-Symbolic	Numerical Variation	Main Test	5,000	5	Answer Accuracy
Omni-MATH	Olympiad Math	Full Set	4,428	0	Judged Pass@1
Code					
HumanEval	Python Synthesis	Test	164	0	Pass@1
General Reasoning					
MMLU	Multitask Knowledge	Test	14,042	0	Accuracy
C-Eval	Chinese Knowledge	Validation	1,346	0	Accuracy
ARC-Challenge	Grade-School Science	Test	1,172	0	Normalized Accuracy
HellaSwag	Commonsense Completion	Validation	10,042	0	Normalized Accuracy
WinoGrande	Pronoun Resolution	Validation	1,267	0	Accuracy

D Supplementary Results

Table 10 reports alternative answer scorers, and Table 13 gives the benchmark-level results behind the sensitivity curves in Figure 3.

Table 10: GSM8K accuracy under strict and flexible answer extraction, and four-shot MATH accuracy under exact and symbolic scoring.

	GSM8K		MATH (4-shot)	
Method	Strict	Flexible	Exact Match	Symbolic
Qwen2.5-3B-Instruct				
Base (w/o Training)	17.7	66.2	29.8	52.3
SFT	48.8	60.5	13.5	29.8
GRPO Without SFT	7.4	33.1	27.3	51.8
SFT→GRPO	49.7	45.5	18.2	27.9
SFT→DAPO	47.5	60.4	24.4	39.3
SFT→GRESO	48.3	60.5	23.0	37.7
CARVE (Ours) (mean)	44.1	64.0	25.9	40.8
Qwen2.5-7B-Instruct				
Base (w/o Training)	35.3	83.8	42.1	70.3
SFT	43.3	81.0	39.0	66.0
GRPO Without SFT	42.2	79.2	38.6	68.5
SFT→GRPO	49.2	77.3	37.4	63.5
SFT→DAPO	47.4	81.7	40.2	67.8
SFT→GRESO	48.1	81.4	39.8	67.3
CARVE (Ours)	50.6	83.2	41.4	69.2

Scoring Protocol Sensitivity. Table 10 shows that the GSM8K comparison depends on answer extraction. At 3B, CARVE leads shared-pool GRPO under flexible extraction but trails it under strict matching. At 7B, it leads under both. Reporting both extractors separates answer-format sensitivity from mathematical verification.

Fixed-Parent Group Replay. The diagnostic in Table 3 draws an independent four-rollout group from the fixed 3B SFT parent for each of the 4,373 candidate RL problems. It compares the full candidate pool with the 2,166 problems retained by the separate eight-rollout outcome screen. On retained problems, outcome-only and dock-only rewards score the same four responses with no policy update. Each problem contributes one group. The within-all-correct rate is normalized by the number of groups with four correct answers, whereas Table 12 averages rates over training batches.

RL Pool Size Sensitivity. Table 11 varies the number of RL problems admitted by their proximity to intermediate post-SFT solve rates. The SFT parent, admitted PRM, candidate partition, and 1,440 updates are fixed. The central band retains 2,166 problems. Wider bands admit progressively easier and harder candidates.

Table 11: Four-task math accuracy as the retained RL pool widens around intermediate post-SFT solve rates at 3B.

Retained	Problems	GSM8K	Minerva	MATH-500	MATH	Avg.
25.0%	1,093	62.2	39.7	47.2	45.3	48.6
49.5%	2,166	64.0	40.8	47.8	46.6	49.8
75.0%	3,280	62.9	40.4	47.0	46.1	49.1
100.0%	4,373	61.7	39.6	46.6	45.3	48.3

Logged Group Diagnostics. Table 12 reports training-batch diagnostics for each reward configuration. Uniform correctness is the mean within-batch share of groups whose sampled answers all have the same binary correctness label. Zero reward variance is the corresponding share computed from the scalar reward passed to optimization. Each rate uses the number of logged batches shown in its column. Format and answer-proximity terms can distinguish rewards among rollouts with identical correctness labels. The screened outcome-only control measures variation under the same allocation. The paired replay in Table 3 isolates the effect of docking on identical responses.

Maximum Dock Sensitivity. The coefficient β in Eq. 6 bounds the quality-dependent dock, with the break penalty $w_b b_i$ added separately. The sweep in Figure 3 varies β while keeping the 3B data allocation, screened RL pool, admitted PRM, mixed-group depth, and optimization budget fixed. The central setting matches the main method in Table 1.

D.1 CROSS-BENCHMARK EVALUATION

Table 5 compares SFT→DAPO and CARVE at each backbone using the released GSM-Symbolic main set and Omni-MATH set. GSM-Symbolic tests numerical variations of GSM8K templates. Omni-MATH tests transfer to harder competition problems. Both sets are excluded from post-training and model selection. GSM-Symbolic responses use the same five-shot prompt and numerical extractor as GSM8K, while Omni-MATH responses use the official GPT-4o equivalence-judgment protocol.

D.2 MATCHED POOL SELECTION

The comparison in Table 6 fixes the 3B half-corpus SFT parent and uses 2,166 unique RL problems for every rule. Each source contributes the same number of problems as in the informative band. Within a source, random selection orders candidates by a seeded hash, whereas the highest- and lowest-solve-rate rules order them by the parent's correct count among eight independent screening rollouts. A fixed hash breaks ties. The informative rule retains problems with both correct and incorrect screening responses. All four pools train with the outcome-only reward for 1,440 optimizer updates, four rollouts per prompt, and the common optimizer settings in Table 8. All arms

Table 12: Batch-mean proportions of groups with uniform answer correctness or zero reward variance during 3B RL.

	Uniform Correctness		Zero Reward Variance	
Training recipe	Rate (%)	Batches	Rate (%)	Batches
Shared-pool GRPO	96.2	5,275	91.9	5,259
Disjoint split	81.8	3,401	61.9	3,397
Split + screening	**73.6**	1,456	57.9	1,452
CARVE	76.3	1,443	**33.4**	1,442

Table 13: Benchmark-level math accuracy across KL regularization and process-dock settings at 3B.

		Math Reasoning				
Method	RL Steps	GSM8K	Minerva	MATH-500	MATH	Avg.
KL Coefficient (Disjoint Split, Outcome Reward)						
Disjoint Split, β_{KL}=0.02	3.2k	58.8	**38.4**	41.6	42.7	45.4
Disjoint Split, β_{KL}=0.04	3.2k	**60.7**	37.4	**43.7**	**43.9**	**46.4**
Disjoint Split, β_{KL}=0.08	3.2k	56.6	37.5	43.6	42.7	45.1
*Mixed-Group Dock Depth (*CARVE*)*						
CARVE, γ_{mix}=0.1	1.4k	62.9	39.7	46.8	45.8	48.8
CARVE, γ_{mix}=0.25	1.4k	**64.0**	**40.8**	**47.8**	**46.6**	**49.8**
CARVE, γ_{mix}=0.5	1.4k	62.2	39.4	46.4	45.8	48.5
*Maximum Dock Strength (*CARVE*)*						
CARVE, β=0.2	1.4k	62.6	40.4	46.6	45.8	48.9
CARVE, β=0.4	1.4k	**64.0**	**40.8**	**47.8**	**46.6**	**49.8**
CARVE, β=0.6	1.4k	63.4	40.4	47.2	45.6	49.1

use outcome-only RL to isolate the problem-selection rule. The informative-band row is the same outcome-only configuration summarized in Table 2.

D.3 PROCESS CREDIT ASSIGNMENT

Reward Integration. The comparisons in Figure 5 hold the 3B SFT parent, informative pool, admitted PRM, and 1,440 RL updates fixed. Additive integration uses $r_i^{\text{add}} = r_i^{\text{out}} + \beta(q_i - \bar{q}_g)$, where $\bar{q}_g$ averages the PRM quality over every rollout in the group and $\beta = 0.4$. The bonus applies to both correct and incorrect responses. The additive variant retains its process bonus above the outcome ceiling. Uniform docking uses Eq. 6 with $\gamma_g = 1$ for every group. Adaptive docking uses the default group-dependent schedule. The three process variants share the incorrect outcome ceiling $c = 0.55$ before their process adjustment. The outcome-only reference retains its original $c = 0.40$ recipe. Uniform and adaptive docking reuse their corresponding configurations in Table 2.

Permutation Control. For each group with at least two correct rollouts, compute the complete process deduction $\delta_i = R - \max\{c, R - \beta\gamma_g(1 - a_i\tilde{q}_i) - w_b b_i\}$ on each correct response, where $a_i = 1 - \alpha + \alpha d_i$. Independently choose the group with probability $\xi \in \{0, 0.25, 0.5, 1\}$. In a chosen group, draw a uniform permutation π of its correct rollouts and assign $R - \delta_{\pi(i)} - \kappa_i$ to correct rollout i. Incorrect rewards are unchanged. This intervention preserves the multiset of process deductions, including its mean and variance, while changing which derivation receives each deduction. Recipient-specific format penalties remain attached to their responses, and final reward variance also depends on their association with the permuted deductions. Groups with fewer than two correct responses are unchanged. The $\xi = 0$ endpoint is the main CARVE configuration, and all four settings use the same optimization budget and evaluation protocol.

D.4 Sampling Budget And Group Size

Policy-Sampling Budget. The left panel of Figure 6 counts tokens generated by each trainable policy and its fixed SFT parent. For CARVE, the counter includes outcome-screening responses, Monte-Carlo continuations used for process labels, and RL rollouts. For the baselines, it includes rejected rollouts from dynamic sampling and any policy-generated screening responses. Evaluation responses are excluded for every method. Each recipe uses its original parent, allocation, optimizer, and decoding settings, and generation stops at cumulative budgets of 120, 240, and 480 million tokens. The metric measures policy sampling rather than total computation.

Response Allocation Across Groups. The right panel fixes the 3B half-corpus SFT parent, informative pool, and 1,440 updates. Group sizes $G \in \{2, 4, 8, 12\}$ use respectively 60, 30, 15, and 10 prompts per update, giving 120 responses per update and 172,800 in total. Both reward recipes use these allocations. CARVE keeps the same frozen PRM and numerical reward parameters, including the mixed-group dock depth. Changing G therefore trades distinct prompts for responses per prompt without increasing the response count. Token lengths may differ across settings. The $G = 4$ endpoints reuse the main method and screened outcome-only configurations. The sufficient margin in Theorem 2 depends on group size and the number of correct responses.