GREEDY-PROMPT POLICY GRADIENT FLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) optimizes a policy over a prompt distribution. Inspired by recent findings that RLVR remains effective when only one or a few prompts are used for gradient updates, we study how each prompt gradient shapes policy-optimization dynamics. This perspective leads to a *greedy flow* that selects the prompt maximizing instantaneous improvement. Although effective, one can hypothesize that the advantage has a trivial explanation: the selected update may simply have a larger component along the population gradient. We therefore remove this parallel advantage through normalization and find that the normalized flow remains effective, showing that parallel rescaling alone does not explain the observed gap in LLM RLVR. Our diagnostics further document substantial orthogonal motion and nontrivial temporal geometry. Motivated by these observations, we study projection-steering regularity as one possible sufficient condition for global dominance over standard gradient flow. Empirically, we design algorithms approximating these flows and evaluate them on Qwen3-8B-Base with LoRA over DAPO-Math-17K. The results show consistent dominance throughout the observed training horizon, with gains transferring across multiple out-of-distribution benchmarks, including MATH-500, AMC 2023, AIME 2024, AIME 2025, Minerva Math, and OlympiadBench.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) [Uesato et al., 2022; Shao et al., 2024; Guo et al., 2025; Yu et al., 2025] optimizes the expected verifier reward over a distribution of prompts. Recent few-shot RLVR studies show that policy optimization using only one or a few prompts can nevertheless improve performance across a much broader prompt distribution [Wang et al., 2025; Koksal & Alatan, 2025; Xiao et al., 2026; Zhang et al., 2026; Pang et al., 2026]. This phenomenon begs the questions: (1) how can individual prompt's gradient lead to significant improvement in the population objective? and (2) does the understanding of the former question tell us how to better select the prompt to use in the updates?

In this work, we explore these questions theoretically in the framework of gradient flows [Zinkevich, 2003; Nutini et al., 2015; Lu et al., 2018; Wibisono et al., 2016; Muehlebach & Jordan, 2020]. Let $J_x(\theta)$ denote the expected return on prompt x, and let $J(\theta) = \mathbb{E}_{x\sim d_0}[J_x(\theta)]$ denote the population objective. To isolate the first-order effect of prompt choice, consider the *prompt-conditioned* policy-gradient flow, where the gradient induced by prompt x_t, ∇J_{x_t}, is used to update the parameter at time t (the x_t themselves will be chosen according to some prompt-selection rule):

$$\mathrm{d}\theta_t = \nabla J_{x_t}(\theta_t)\ \mathrm{d}t \quad \Rightarrow \quad \frac{\mathrm{d}}{\mathrm{d}t}J(\theta_t) = \langle \nabla J(\theta_t), \nabla J_{x_t}(\theta_t)\rangle\,. \tag{1}$$

As a simple observation, if we choose $x_t = \arg\max_{x\in \mathrm{Supp}(d_0)} \langle \nabla J(\theta_t), \nabla J_x(\theta_t)\rangle$, it holds that

$$\langle \nabla J(\theta_t), \nabla J_{x_t}(\theta_t)\rangle \geq \mathbb{E}_{x\sim d_0}\left[\langle \nabla J(\theta_t), \nabla J_x(\theta_t)\rangle\right] = \|\nabla J(\theta_t)\|_2^2\,. \tag{2}$$

The above observation essentially defines a greedy oracle: at each parameter θ, the oracle selects the prompt whose policy gradient produces the largest instantaneous improvement in the population objective. Since max is no less than average, such a greedy choice of x_t yields larger instantaneous improvement rate $\mathrm{d}J/\mathrm{d}t$ than that of a standard gradient flow $\mathrm{d}\theta_t = \nabla J(\theta_t)\,\mathrm{d}t$. Indeed, when the resulting *greedy flow* is approximated empirically, it yields strong gains over uniform prompt selection and batch-averaged policy gradients (Section 4).

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.

This inequality alone, however, does not provide a satisfactory explanation of the empirical gain. Most notably, the selected prompt's gradient, ∇J_{x_t}, has a component $c_t \nabla J$ that is parallel to the population gradient with $c_t = \langle \nabla J, \nabla J_{x_t} \rangle / \|\nabla J\|_2^2 \geq 1$. Hence, one can reasonably suspect that updating in the greedy prompt's direction is largely equivalent to taking a larger effective step along the standard gradient direction ∇J. We isolate this effect through a normalized greedy flow in Sec. 5, where the normalization rescales the selected update so that its instantaneous population improvement matches that of the standard population-gradient flow at the same state, while retaining its prompt-specific direction. Perhaps surprisingly, its practical approximation retains most of the raw greedy improvement and remains ahead of the standard baseline throughout the observed training horizon.

The above results suggest a trajectory effect beyond instantaneous improvement. Prompt-conditioned gradients in our experiments contain large components orthogonal to the population gradient, and greedy selection retains substantial orthogonal energy. This component does not contribute to the first-order improvement of J, but changes the parameter trajectory. Temporal diagnostics describe its alignment with subsequent population-gradient changes without establishing a favorable steering effect. The improvement retained after normalization therefore motivates studying orthogonal steering as a possible explanation, rather than attributing the entire greedy advantage to parallel rescaling. Motivated by these observations, we study one interpretable sufficient formalization in which prompt-gradient projections are coupled to future population speed. Under this condition, the normalized greedy flow globally dominates the population-gradient flow. The raw and normalized greedy flows follow the same controlled orbit at different speeds, which further yields raw greedy dominance over its normalization.

The population oracle is unavailable during training, since neither the exact population gradient nor the full prompt support can be enumerated. We therefore introduce a finite-batch greedy leave-one-out tournament. Prompts in the current batch form the candidate set, while the remaining prompts estimate the population direction used to score each candidate. A blockwise randomized projection [Johnson & Lindenstrauss, 1984; Ailon & Chazelle, 2006; Park et al., 2023] makes the required gradient inner products practical without storing one full-dimensional gradient per candidate. Our contributions are as follows:

- **Prompt-controlled policy-gradient flows.** In Sec. 3, we formulate RLVR updates as controls selected from prompt-conditioned policy gradients and introduce the greedy flow. Sec. 5 introduces its normalized counterpart, which removes the winner's excess projection onto the population gradient, separating prompt-selection geometry from trivial learning-rate scaling.
- **Orthogonal motion and conditional global dominance.** We show that selected prompt gradients retain substantial orthogonal energy and report descriptive temporal diagnostics of the resulting trajectory geometry. Motivated by these observations, we study projection–steering regularity as one sufficient structural condition. Under exact expectations and the stated assumptions, normalized greedy globally dominates vanilla population-gradient flow, while raw greedy inherits dominance through a time reparameterization.
- **Experimental validation.** Using Qwen3-8B-Base [Yang et al., 2025] with LoRA [Hu et al., 2021] on DAPO-Math-17K [Yu et al., 2025] in the `verl` framework [Sheng et al., 2025], we implement a practical leave-one-out tournament and compare greedy selection with uniform selection and batch policy gradients. Across the observed optimization trajectory, greedy selection remains ahead of the comparison methods and yields improvements that generalize beyond the training distribution to six held-out mathematical-reasoning benchmarks.

2 PRELIMINARIES

Preliminaries and Related Work. We adopt the standard contextual-bandit formulation of RLVR over prompts and responses with verifier rewards [Uesato et al., 2022; Le et al., 2022b; Shojaee et al., 2023; Liu et al., 2023; Lambert et al., 2025; Guo et al., 2025; Team et al., 2025; Hu et al., 2025; Cui et al., 2025], with standard notation such as policy π_θ, prompt objective $J_x(\theta)$, and population objective $J(\theta)$. See details in App. E along with a brief review of policy-optimization algorithms for RLVR [Ziegler et al., 2020; Stiennon et al., 2022; Ouyang et al., 2022; Shao et al., 2024; Liu et al., 2025; Yu et al., 2025; Yue et al., 2025; Zheng et al., 2025]. The appendix also reviews related work

on few-shot RLVR [Wang et al., 2025; Koksal & Alatan, 2025; Tang et al., 2026; Yi et al., 2026; Zhu et al., 2026; Wu et al., 2026] and gradient-based sample selection [Nutini et al., 2015; Lu et al., 2018; Zhao & Zhang, 2015; Perekrestenko et al., 2017; Katharopoulos & Fleuret, 2018; Loshchilov & Hutter, 2016; Chang et al., 2017; Kawaguchi & Lu, 2020; Shah et al., 2020; Joseph et al., 2019; Alistarh et al., 2018; Shi et al., 2019; Sahu et al., 2021].

3 Prompt-Conditioned Dynamics and the Greedy Oracle

Before proceeding with our detailed analyses, we first elaborate on the problem formulation, which was briefly introduced in Sec. 1 but merits a more thorough discussion here. Based on the consensus established in Sec. E, we can now introduce the convention that will be repeatedly invoked throughout the paper. Consider any $x \in \mathrm{Supp}(d_0) \subseteq \mathcal{X}$, where $\mathcal{X}$ is the prompt space and $\mathrm{Supp}(d_0)$ is the support induced by the prompt distribution d_0. For any parameter θ, we denote the single-prompt return as

$$J_x(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\left[R(x,y)\,|\,x\right], \tag{3}$$

which satisfies $\mathbb{E}_{x\sim d_0}[J_x(\theta)] = J(\theta)$ and $\mathbb{E}_{x\sim d_0}[\nabla J_x(\theta)] = \nabla J(\theta)$ by definition. Given a selection rule $x^*(\theta)$ that chooses the prompt based on the current parameter θ, we use the shorthand $\nabla J_{x^*}(\theta) = \nabla J_x(\theta)|_{x=x^*(\theta)}$ (that is, the gradient operator treats the prompt as fixed); the same convention applies to $\nabla J_{x^*,\|}$ and $\nabla J_{x^*,\perp}$, which we introduce shortly.

The Gradient Flows. Gradient flows can be viewed as the continuous-time limit of discrete-time gradient updates and are an important mathematical idealization for studying training dynamics [Zinkevich, 2003; Nutini et al., 2015; Lu et al., 2018; Wibisono et al., 2016; Muehlebach & Jordan, 2020]. In App. E.2, we prescribe how a common family of learning-rate schedules approaches its vanishing-step limit and define all compared flows in this paper through the synchronous limit under that family. Consequently, $\mathrm{d}J/\mathrm{d}t$ and every comparison between two flows use the same cumulative-learning-rate clock. We also discuss alternative definitions and explain why a common learning schedule is appropriate for the present comparison.

The Optimization Setup. Starting from $t = 0$ and the pretrained parameters θ_0, we study an arbitrary flow $\{\theta_t\}_{0\le t<T}$ induced by a gradient-style algorithm $\mathrm{d}\theta_t = g_t\,\mathrm{d}t$. We restrict the update vector g_t to the candidate set

$$\mathcal{C}(\theta_t) = \{\nabla J_x(\theta_t)\,|\,x \in \mathcal{X}\} \tag{4}$$

and its convex combinations. For instance, the vanilla population policy gradient corresponds to $g_t^{(e)} = \nabla J(\theta_t^{(e)}) = \mathbb{E}_{x\sim d_0}[\nabla J_x(\theta_t^{(e)})]$, which induces $\mathrm{d}\theta_t^{(e)} = \nabla J(\theta_t^{(e)})\,\mathrm{d}t$ and $\frac{\mathrm{d}}{\mathrm{d}t}J(\theta_t^{(e)}) = \|\nabla J(\theta_t^{(e)})\|_2^2$. Here the superscript (e) denotes "expectation", which refers to vanilla gradient updates, and (g) below denotes "greedy", which refers to our greedy prompt selection. Both identities should be understood in the Carathéodory sense defined in Sec. E: $\theta_t^{(e)}$ is absolutely continuous, the first identity holds almost everywhere, and the second follows from the chain rule for absolutely continuous trajectories.

The Ideal Greedy Oracle. As suggested in Eq. 1, if we select only one $x \in \mathcal{X}$ to construct $\nabla J_x(\theta) \in \mathcal{C}(\theta)$, the chain rule $\mathrm{d}J(\theta) = \langle \nabla J(\theta), \nabla J_x(\theta)\rangle\,\mathrm{d}t$ suggests the following oracle:

$$x^*(\theta) \in \arg\max_{x\in\mathcal{X}} \langle \nabla J(\theta), \nabla J_x(\theta)\rangle\,. \tag{5}$$

We assume that the maximizer correspondence is nonempty, fix a measurable selection including a measurable tie-breaking rule, and allow $x^*(\theta_t)$ to switch infinitely often along a trajectory. The oracle induces the greedy flow $\mathrm{d}\theta_t^{(g)} = \nabla J_{x^*}(\theta_t^{(g)})\,\mathrm{d}t$, which has no smaller instantaneous growth than the vanilla control when both are evaluated at the common parameter $\theta_t^{(g)}$.

3.1 A Practical Greedy Tournament

Practical Batch Algorithms. The oracle rule requires direct access to the population gradient $\nabla J(\theta)$ and scores each prompt by $\langle \nabla J_x(\theta), \nabla J(\theta)\rangle$. In practice, this population gradient is unavailable, and the learner can only observe a finite batch $\mathcal{B} = \{x_i\}_{i=1}^B$ at each optimization step. For each x_i, the gradient estimate $\widehat{\nabla J_{x_i}}(\theta)$ is constructed from G i.i.d. rollouts. We estimate $\nabla J(\theta)$

independently of the i-th candidate using the leave-one-out mean $\widehat{\nabla J}_i(\theta) = \frac{1}{B-1}\sum_{j\neq i}\widehat{\nabla J_{x_j}}(\theta)$. Consequently,

$$\mathbb{E}\Big[\Big\langle \widehat{\nabla J_{x_i}}(\theta), \widehat{\nabla J}_i(\theta)\Big\rangle \,\Big|\, x_i, \theta\Big] = \langle \nabla J_{x_i}(\theta), \nabla J(\theta)\rangle\,,$$

so each inner product is an unbiased estimate of the corresponding oracle score. Selecting the largest among these B empirical scores gives the batch greedy tournament in Alg. 1. The normalized variant introduced below uses the same selected candidate and rescales the effective update by $\frac{\langle \widehat{\nabla J}(\theta), \widehat{\nabla J}(\theta)\rangle}{\langle \widehat{\nabla J_{x_{i^*}}}(\theta), \widehat{\nabla J}(\theta)\rangle}$.

4 Testing the Greedy Oracle

4.1 Implementation and Reproducibility

Overview of the Workflow. We implement our method in `verl` [Sheng et al., 2025] following the four-stage decomposition in our formulation: candidate construction, leave-one-out estimation of the mean gradient, greedy selection, and update commitment. At each training step, we sample B distinct prompts and generate G responses per prompt with vLLM [Kwon et al., 2023]. The resulting rollouts yield a candidate update and an independent clean-gradient estimate for each prompt. Using these quantities, the tournament selects $\widehat{x^*}$ based on leave-one-out inner products and commits the corresponding candidate update to the optimizer. Our method therefore changes only the prompt-level selection rule while leaving the surrounding RLVR optimization procedure unchanged. Readers can also find the entire procedure in Fig. 2.

Blockwise Fast-JL Projection. Materializing and retaining B full-dimensional candidate updates is infeasible at LLM scale. We instead stream parameter tensors through an implicit Rademacher Johnson-Lindenstrauss projection [Johnson & Lindenstrauss, 1984; Ailon & Chazelle, 2006], implemented using the `fast_jl` CUDA kernels released with TRAK [Park et al., 2023]. Our blockwise construction concatenates 32 independently seeded $8{,}192$-dimensional sub-sketches, producing a $262{,}144$-dimensional sketch for each candidate update and clean-gradient estimate. A common, run-fixed projection is applied to all sketches, allowing the tournament to approximate the required inner products without materializing either the full-dimensional vectors or the projection matrices.

Probe Parallelization. Projection reduces memory use, but constructing B candidates still requires $O(B)$ autodiff passes. We therefore distribute the engineering and clean-gradient probes across isolated replica groups. Each probe starts from the same pre-update state and cannot alter the optimizer, scheduler, gradient scaler, buffers, or RNG state observed by subsequent probes. Each replica processes a disjoint subset of prompts, after which we gather the sketches and run the tournament over the complete candidate set. This parallelization therefore preserves both the selected winner and the committed update.

Our Training Recipe. With the computational workflow fixed, we isolate the effect of selection under a common experimental recipe. We partition DAPO-Math-17K [Yu et al., 2025] into three disjoint banks of $2{,}048$ prompts, denoted by $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{val}}$, and $\mathcal{D}_{\text{eval}}$. At each step, we sample a seeded batch of B distinct prompts from $\mathcal{D}_{\text{train}}$ to approximate sampling from d_0. We use $\mathcal{D}_{\text{val}}$ to independently track $J(\theta)$ throughout training, while gradient-geometry diagnostics are computed on training batches, as reported in Fig. 1. After training, we estimate the final in-distribution objective on $\mathcal{D}_{\text{eval}}$ using the same policy parameterization and decoding configuration, including the temperature and top-k sampling settings. We additionally report pass@1, response length, and performance on public OOD benchmarks in Tab. 1. Once B is fixed, all comparisons share the optimizer, scheduler, rollout, and evaluation budgets from Tab. 2 to 7.

4.2 Greedy Selection versus the Baselines

Comparison with Uniform Selection & Batch SGD. We begin with two complementary baselines that isolate different aspects of greedy selection. Batch SGD supplies the population-gradient baseline used in our theory, while uniform selection preserves the single-candidate update structure but removes the selection criterion; as an additional interpretation, it also coincides in distribution with the $B = 1$ case. With the optimizer and learning-rate schedule held fixed, greedy reaches

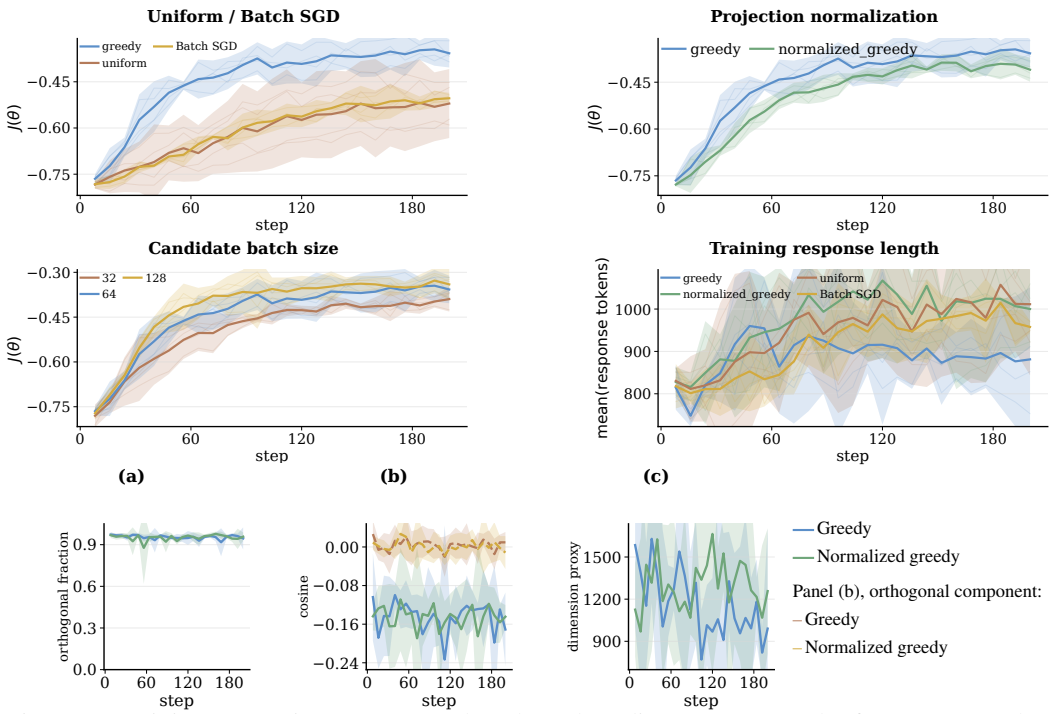


Figure 1: Training comparisons, response length, and gradient geometry. The first two rows show monitor-bank objectives for greedy versus uniform/Batch SGD, greedy versus normalized greedy, greedy batch sizes, and training-batch mean response length. The bottom row shows (a) winner orthogonal energy fraction $\|\nabla J_{x^*,\perp}\|^2/\|\nabla J_{x^*}\|^2$; (b) cosine of the winner (solid) or its orthogonal component (dashed) with the next population-gradient change; (c) reciprocal mean squared off-diagonal pair cosine of candidate orthogonal components, a coherence-based dimension proxy. The cosine panel uses consecutive raw measurements before subsampling; these first-order diagnostics do not directly test Assump. 3. Gradient estimates use the same fixed JL operator within each run. Curves use actual monitor steps without smoothing, with pointwise Student-t 95% intervals over four seeds, clipped only for display where necessary.

-0.357 ± 0.046 on the independent monitor bank $\mathcal{D}_{\text{val}}$, compared with -0.521 ± 0.112 for uniform selection and -0.504 ± 0.018 for Batch SGD. Across the observed training horizon, greedy remains ahead of both baselines, with two-sided 95% Student-t intervals computed over four seeds.

5 Normalization Reveals Non-Parallel Trajectory Effects

5.1 Removing the Parallel Advantage

The Normalized Greedy Oracle. The preceding comparison shows that the greedy oracle leads to strong empirical gains. Nevertheless, one can reasonably argue that this is due to ∇J_{x^*} having a larger component along ∇J than ∇J itself. We quantify this advantage by

$$\rho(\theta) = \frac{\langle \nabla J(\theta), \nabla J_{x^*}(\theta) \rangle}{\langle \nabla J(\theta), \nabla J(\theta) \rangle} \geq 1. \tag{6}$$

We remove this scalar advantage by defining the *normalized greedy flow* on the same $\mathrm{d}t$ as $\mathrm{d}\theta_t^{(i)} = \rho(\theta_t^{(i)})^{-1} \nabla J_{x^*}(\theta_t^{(i)})\,\mathrm{d}t$. At a common parameter $\theta_t^{(i)}$, its projection onto ∇J exactly matches that of vanilla because $\left\langle \nabla J(\theta_t^{(i)}), \rho(\theta_t^{(i)})^{-1} \nabla J_{x^*}(\theta_t^{(i)}) \right\rangle = \|\nabla J(\theta_t^{(i)})\|_2^2$. Consequently, a strict advantage of $\theta_t^{(i)}$ over $\theta_t^{(e)}$ cannot be attributed to a larger instantaneous parallel component at the same state; it must arise through the trajectory induced by ∇J_{x^*}'s orthogonal component.

For every admissible θ with $\nabla J(\theta) \neq 0$, define

$$W(\theta) = \|\nabla J(\theta)\|_2^2, \qquad \rho_x(\theta) = \frac{\langle \nabla J_x(\theta), \nabla J(\theta) \rangle}{W(\theta)}, \qquad \rho(\theta) = \max_{x \in \mathcal{X}} \rho_x(\theta) \geq 1. \tag{7}$$

Comparison with Inner-Product Normalization. The preceding gain could be trivial if the winner merely enlarged the effective step along ∇J. We therefore divide the selected update by its projection ratio[1], matching vanilla's instantaneous objective-growth rate while preserving the selected direction. Normalized greedy still reaches -0.410 ± 0.039, retaining most of the advantage over uniform selection and Batch SGD. This rules out pure parallel rescaling as a complete explanation and motivates the orthogonal-steering interpretation examined below.

5.2 Observable Orthogonal Geometry and Temporal Diagnostics

We decompose the winner as $\nabla J_{x^*,\|}(\theta) = \rho(\theta)\nabla J(\theta)$ and $\nabla J_{x^*,\perp}(\theta) = \nabla J_{x^*}(\theta) - \nabla J_{x^*,\|}(\theta)$, and similarly write $\nabla J_{x_i,\perp}(\theta) = \nabla J_{x_i}(\theta) - \rho_{x_i}(\theta)\nabla J(\theta)$ for each sampled candidate. The following diagnostics examine the magnitude, selection, and subsequent evolution of these orthogonal components.

Training Diagnostics. We next examine how the orthogonal component appears in the observed training dynamics. The right-hand panels of Fig. 1 report three complementary diagnostics. First, the selected updates retain a large fraction of orthogonal energy, indicating that prompt selection preserves substantial motion beyond the estimated population-gradient direction. Second, the signed cosine with the next population-gradient change reveals a distinct temporal-alignment pattern for the full winner and its orthogonal component, suggesting that the retained component participates in the subsequent evolution of the gradient field. Third, the coherence-based dimension proxy indicates that the candidate orthogonal components occupy a broad collection of directions rather than collapsing onto a single residual direction. Taken together, these measurements show that the orthogonal component remains geometrically substantial and is reflected in the evolving optimization trajectory, motivating the steering formalization developed in Sec. 6.

6 A Sufficient Formalization of Orthogonal Steering

The observed orthogonal motion and objective ordering raise a further question: under what conditions can the former imply the latter? We now formalize one possible mechanism. Along a purely orthogonal displacement, the first-order change in W is $\langle \nabla W, \nabla J_x - \rho_x \nabla J \rangle$, so its sign describes whether this displacement enters a region with larger population-gradient power. This local effect alone does not establish global dominance; the assumptions below couple it to prompt selection and control its propagation across trajectories.

Leveled Guarantees for a Desirable Oracle. Before formally determining the *goodness* of a flow at a common t, we distinguish three properties: *initial dominance* asks whether one flow improves J at least as quickly within a nonzero initial window; *convergence* asks whether its objective approaches J^* as the available horizon grows; and *global dominance* asks whether its objective is no smaller than the baseline objective at every t. Under ordinary regularity and a strict initial greedy margin, initial dominance follows over a local window, and a standard PL inequality further supplies convergence to the optimum (App. I). Global dominance requires additional structure controlling how winner-induced steering changes future population speed and can remain uninformative if it is explained only by parallel rescaling.

Our Formal Assumptions. The assumptions below formalize the additional conditions required to extend this local mechanism to global dominance. Moreover, the proof must reuse the maximizing definition of x^* to couple the selected projection with its orthogonal effect; otherwise, the result would apply equally to an arbitrary candidate with convenient parallel and orthogonal components and would not explain the greedy selection mechanism.

[1]The exact-flow identity uses population expectations and guarantees $\rho(\theta) \geq 1$ because the maximum projection is no smaller than its population mean. In the finite-batch implementation, the winner is selected with a leave-one-out mean, whereas the normalization denominator uses the full-batch clean-gradient mean, as specified in Alg. 2. Consequently, the empirical scaling factor is an estimator of the exact inner ratio and need not inherit the pointwise inequality $\widehat{\rho} \geq 1$.

The preceding experiments show that normalized greedy retains a substantial portion of the raw greedy improvement. This observation motivates studying a non-parallel trajectory mechanism, but does not by itself establish the projection–steering condition introduced below. We therefore introduce a set of sufficient assumptions for analyzing this regime.

Assumption 1 (Parameterization Regularity). *The compared dynamics remain in $\Theta_0 \subseteq \mathbb{R}^d$ from θ_0. For every x, y, θ, the score $\nabla \log \pi_\theta(y|x) = \sum_{h=0}^{|y|-1} \nabla \log \pi_\theta(y_h|x, y_{<h})$ exists and, for some $G_1, G_2 > 0$,*

$$\|\nabla \log \pi_\theta(y|x)\|_2 \le G_1, \qquad \|\nabla \log \pi_{\theta_1}(y|x) - \nabla \log \pi_{\theta_2}(y|x)\|_2 \le G_2 \|\theta_1 - \theta_2\|_2 . \tag{8}$$

We also assume that, every switching field $\in \{\theta_t^{(g)}, \theta_t^{(e)}, \theta_t^{(i)}\}$ admits a unique absolutely continuous Carathéodory solution in Θ_0.

Assumption 2 (Level-Speed Regularity). *There exists $L \ge 0$ such that, for every $\theta_1, \theta_2 \in \Theta_0$ with $J(\theta_1) = J(\theta_2)$ and $W(\theta_1), W(\theta_2) > 0$,*

$$\left| \frac{\langle \nabla W(\theta_1), \nabla J(\theta_1) \rangle}{W^2(\theta_1)} - \frac{\langle \nabla W(\theta_2), \nabla J(\theta_2) \rangle}{W^2(\theta_2)} \right| \le L \left| \log \frac{W(\theta_1)}{W(\theta_2)} \right| . \tag{9}$$

Assumption 3 (Projection–Steering Regularity). *There exist $0 < \lambda \le \Lambda < \infty$ and $\epsilon_c \ge 0$ such that, for every θ and every pair satisfying $\rho_x(\theta) \ge \rho_z(\theta)$,*

$$\begin{aligned} \lambda(\rho_x(\theta) - \rho_z(\theta)) - \epsilon_c \le \frac{\langle \nabla W(\theta), \nabla J_{x,\perp}(\theta) - \nabla J_{z,\perp}(\theta) \rangle}{W^2(\theta)} \\ \le \Lambda(\rho_x(\theta) - \rho_z(\theta)) + \epsilon_c . \end{aligned} \tag{10}$$

Assumption 4 (Minimal Improvement Ratio). *There exist constants $\rho_{\min} > 1$ such that*

$$\rho(\theta) \ge \rho_{\min}, \qquad \forall \theta \in \Theta_0 . \tag{11}$$

Assumption 5 (PL Envelope). *For convergence to a common optimum only, there exists $\nu > 0$ such that*

$$W(\theta) \ge 2\nu(J^* - J(\theta)). \tag{12}$$

Interpretation and Empirical Support. **A1–A2** supply the regularity needed for the flows, where the latter performs comparison at equal objective levels. The central structural condition is **A3**, which gives one analytically tractable meaning to favorable orthogonal steering after the candidate's instantaneous parallel advantage has been removed. Replacing $\rho_x \nabla J$ by ∇J while retaining $\nabla J_x - \rho_x \nabla J$ gives $\dot\theta = \nabla J + \nabla J_x - \rho_x \nabla J$ and $\mathrm{d}r/\mathrm{d}t = \left\langle \nabla J, \dot\theta \right\rangle = W$. Hence

$$\frac{\mathrm{d}\log W}{\mathrm{d}r} = \frac{\mathrm{d}\log W}{\mathrm{d}W} \left\langle \nabla W, \frac{\mathrm{d}\theta}{\mathrm{d}t} \right\rangle \frac{\mathrm{d}t}{\mathrm{d}r} = \frac{\langle \nabla W, \nabla J \rangle}{W^2} + \frac{\langle \nabla W, \nabla J_{x,\perp} \rangle}{W^2}. \tag{13}$$

The second term is the candidate's orthogonal contribution to future population-gradient power per unit objective progress. A3 allows local violations through ϵ_c, while requiring candidates with larger projection coefficients to have correspondingly better orthogonal responses up to this slack. This condition is not claimed to be the unique explanation of the observed steering, nor do our first-order logs directly evaluate ∇W. Instead, Fig. 1 supplies only partial motivation: selected updates retain substantial orthogonal energy, the temporal cosine exhibits nontrivial but not uniformly favorable alignment, and candidate residuals have a large coherence-based dimension proxy. These observations further motivate us to study orthogonal trajectory effects.

Normalized Greedy Dominates Vanilla. We first compare normalized greedy with vanilla under a fixed parameterization and exact population expectations. Thm. 1 shows that normalized greedy remains globally ahead under the common clock whenever the regularity and steering constants are jointly compatible; in particular, it suffices that $\lambda(\rho_{\min} - 1) - \epsilon_c > 0$. This holds even after removing the winner's instantaneous parallel advantage. The remaining gain therefore arises from the winner-specific orthogonal steering in Assump. 3, whose effect is propagated across objective levels by Assump. 2, rather than from a trivial rescaling of ∇J.

Theorem 1 (Global Dominance for Normalized Greedy). *Under Assumptions 1-5, when $\lambda(\rho_{\min} - 1) - \epsilon_c > 0$, then for every $t \in [0, T)$,*

$$J(\theta_t^{(i)}) \ge J(\theta_t^{(e)}). \tag{14}$$

Our proof in App. J.1 reparameterizes both trajectories by their attained objective level r, so that reaching time becomes the integral of $1/W$ over objective levels. The chain rule then yields a recur-

sion for the equal-level log-power ratio. Assump. 3 and the maximizing definition of x^* provide its positive steering term, while Assump. 2 controls cross-trajectory distortion. Grönwall's inequality consequently establishes power dominance at every jointly reachable level, which converts directly into first-arrival-time dominance through the preceding integral representation. Finally, the PL condition rules out distinct suboptimal limits, extending the result whether or not the horizon T includes the first convergence time.

Greedy versus Normalized Greedy. As a direct corollary, Thm. 2 shows that raw greedy further dominates normalized greedy solely through its additional parallel scaling.

Theorem 2 (Dominance for the Greedy Flow)**.** *Under Assumptions 1-5, when* $\lambda(\rho_{\min} - 1) - \epsilon_c > 0$*, then for every* $t \in [0, T)$*,*

$$J(\theta_t^{(g)}) \geq J(\theta_t^{(i)}) \geq J(\theta_t^{(e)}). \tag{15}$$

See our proof in App. J.2, where we show that the raw and normalized greedy vector fields are pointwise collinear, with the former equal to $\rho(\theta)$ times the latter. Uniqueness therefore identifies their trajectories up to a time reparameterization $s'(t) = \rho(\theta_{s(t)}^{(i)}) \geq \rho_{\min} > 1$, resulting in $s(t) \geq t$. Since J is nondecreasing along this common trajectory, we therefore complete the proof.

7 Additional Empirical Observations

Having presented one sufficient theoretical mechanism for the observed ordering, we return to empirical consequences beyond the flow theorem.

7.1 Candidate-Pool Size

Comparison under Batch-size Variation. Having established that the effect survives normalization, we next examine the finite-candidate approximation underlying the practical tournament. Increasing B from 32 to 64 improves the terminal monitor objective from -0.390 to -0.357, whereas increasing it further to 128 yields only the smaller improvement to -0.340. The final evaluations similarly place $B = 64$ and $B = 128$ close on most metrics. This empirical diminishing return suggests that a modest candidate pool already captures much of the attainable selection gain.

7.2 Final Evaluation and Response Behavior

Final Evaluations. The final evaluations test whether the observed monitor-objective advantage transfers beyond the quantity covered by the theorem. Across the in-distribution bank and six public OOD benchmarks, raw greedy is generally the strongest method, while normalized greedy is consistently strongest or second strongest among the remaining fixed-B methods. We also observe additional empirical associations not predicted by our theory: raw greedy generally produces shorter responses while maintaining or improving verifier reward and pass@1, and Tab. 1 additionally reports boxed-answer rates to examine response-format compliance.

Other Training Reports. For completeness, App. F.2 reports cumulative rollout counts and per-step wall-clock time for the four selection methods at $B = 64$ in Fig. 3, providing complementary views of their data and computational costs. Apps. F.3–F.17 report the complete set of secondary in-distribution and OOD evaluation metrics in Tabs. 8–22.

8 Conclusions, Discussions, and Limitations

Conclusion. We introduced a greedy flow that selects the prompt-conditioned update with the largest estimated population improvement, then normalized its parallel component to isolate nontrivial trajectory effects. Our analysis identifies a sufficient projection–steering condition under which normalized greedy globally dominates standard gradient ascent, and our leave-one-out tournament provides a practical approximation. Experiments on LLM post-training show a consistent performance ordering throughout training, with promising transfer across OOD final evaluation banks.

The Origin of Orthogonal Steering. Curious readers may find a deeper question underlying our analysis: why does selecting a winner solely by its parallel projection also produce a favorable,

Verifier reward (↑)

Method	id_2048	math500	amc23	aime24	aime25	minerva	olympiad
greedy	**-0.37±0.06**	**0.22±0.04**	**0.05±0.07**	**-0.80±0.03**	**-0.78±0.02**	**-0.56±0.01**	**-0.21±0.04**
normalized_greedy	**-0.40±0.02**	**0.19±0.02**	**0.01±0.04**	**-0.83±0.02**	**-0.78±0.05**	**-0.58±0.01**	**-0.24±0.01**
uniform	-0.54±0.13	0.06±0.17	-0.19±0.21	-0.89±0.08	-0.87±0.08	-0.64±0.10	-0.37±0.17
Batch SGD	-0.53±0.03	0.07±0.05	-0.15±0.04	-0.85±0.04	-0.85±0.03	-0.65±0.02	-0.35±0.04
greedy $B = 32$	-0.41±0.01	0.20±0.02	-0.00±0.06	-0.83±0.04	**-0.78±0.02**	-0.57±0.01	-0.24±0.01
greedy $B = 64$	**-0.37±0.06**	**0.22±0.04**	**0.05±0.07**	**-0.80±0.03**	**-0.78±0.02**	**-0.56±0.01**	**-0.21±0.04**
greedy $B = 128$	**-0.36±0.04**	**0.22±0.02**	**0.07±0.05**	**-0.80±0.04**	-0.81±0.04	**-0.55±0.02**	**-0.21±0.03**

Pass@1 (%) (↑)

Method	id_2048	math500	amc23	aime24	aime25	minerva	olympiad
greedy	**32.28±3.17**	**61.22±1.83**	**52.99±3.54**	**12.08±1.38**	**11.72±1.13**	**22.35±0.22**	**39.94±2.01**
normalized_greedy	**31.50±0.63**	**60.00±1.05**	**51.60±1.95**	**12.47±1.10**	**12.60±3.19**	**21.08±0.65**	**39.48±0.82**
uniform	24.98±6.99	53.47±8.76	41.95±10.01	10.31±4.83	8.54±3.75	18.18±5.02	33.06±8.52
Batch SGD	25.45±1.77	53.80±2.37	43.61±1.90	11.22±2.06	9.32±1.37	17.75±0.98	34.12±1.91
greedy $B = 32$	30.92±0.62	60.46±0.98	50.72±3.17	**12.08±2.30**	**12.06±1.19**	21.38±0.31	39.23±0.52
greedy $B = 64$	**32.28±3.17**	**61.22±1.83**	**52.99±3.54**	**12.08±1.38**	**11.72±1.13**	**22.35±0.22**	**39.94±2.01**
greedy $B = 128$	**32.86±1.92**	**61.18±0.86**	**54.02±2.32**	**11.46±3.48**	9.90±2.17	**22.54±1.03**	**40.13±1.10**

Mean response length (↓)

Method	id_2048	math500	amc23	aime24	aime25	minerva	olympiad
greedy	**884.90±170.52**	**559.06±56.44**	**814.35±115.40**	**1080.40±245.73**	**911.12±161.95**	**568.76±58.49**	**822.69±115.13**
normalized_greedy	1018.29±113.05	613.20±37.31	925.27±86.08	1284.55±241.23	**1054.27±143.65**	**616.21±22.71**	936.97±83.26
uniform	1013.76±130.83	**608.07±72.03**	936.27±83.42	1292.51±279.49	1100.81±121.16	645.36±36.63	939.11±103.16
Batch SGD	**992.74±34.84**	612.93±10.71	**923.72±38.63**	**1263.13±44.42**	1065.57±19.28	654.44±14.66	**930.31±26.74**
greedy $B = 32$	991.03±43.36	602.41±17.81	906.47±40.14	1272.08±160.53	1044.83±49.86	587.74±68.60	916.05±27.35
greedy $B = 64$	**884.90±170.52**	**559.06±56.44**	**814.35±115.40**	**1080.40±245.73**	**911.12±161.95**	**568.76±58.49**	**822.69±115.13**
greedy $B = 128$	**907.65±252.10**	**564.55±120.15**	**817.16±212.07**	**1074.57±299.60**	**906.74±274.06**	**576.35±112.50**	**830.02±184.27**

Boxed-answer rate (%) (↑)

Method	id_2048	math500	amc23	aime24	aime25	minerva	olympiad
greedy	**64.59±33.23**	**86.75±15.43**	**88.30±17.03**	**86.30±14.16**	**91.28±11.90**	**75.40±23.31**	**89.16±13.32**
normalized_greedy	**55.09±33.38**	**76.83±22.07**	**79.18±23.16**	**73.85±24.44**	**81.80±19.68**	**65.19±20.87**	**79.10±21.10**
uniform	34.65±24.67	55.93±22.64	59.26±19.28	55.49±16.91	62.11±19.77	44.53±24.27	57.83±21.91
Batch SGD	33.79±3.65	56.73±3.96	61.04±2.69	58.98±5.00	63.31±4.76	44.73±4.04	60.39±4.10
greedy $B = 32$	43.45±24.76	69.25±23.30	70.90±26.75	67.92±25.04	74.58±28.42	56.09±17.74	71.83±24.72
greedy $B = 64$	**64.59±33.23**	**86.75±15.43**	**88.30±17.03**	**86.30±14.16**	**91.28±11.90**	**75.40±23.31**	**89.16±13.32**
greedy $B = 128$	**77.88±20.53**	**93.23±8.02**	**95.31±5.02**	**92.53±3.25**	**96.98±1.90**	**81.15±18.46**	**94.63±3.87**

Table 1: Final evaluation across four seeds. Entries are means ± two-sided 95% Student-t confidence intervals. Black bold marks the best mean and gray bold the second-best distinct mean separately within the fixed-B selector block and the greedy batch-size block; arrows state the metric direction.

temporally second-order orthogonal steering effect? We share this curiosity and leave the origin of this coupling unresolved. Our experiments document substantial orthogonal motion, while leaving the origin and effect of this coupling unresolved. Whether this condition is the correct underlying explanation, whether the coupling is specific to LLM RLVR, or whether it follows from more general properties of high-dimensional per-sample gradients remains an important direction for future work.

Computational Regime. Our method uses additional parallel backpropagation to select an update while holding the rollout and optimizer-step budgets fixed under a common batch configuration B. It is most attractive when rollouts or optimizer steps are expensive and parallel automatic differentiation is comparatively affordable. Its benefits under different hardware, communication, and inference-cost regimes remain to be studied.

Misspecification and Extensions. In the experiments, we use REINFORCE with SGD to align the implementation with our analysis. More generally, candidates may be updates produced by an arbitrary algorithm and optimizer, including PPO, GRPO, importance clipping, KL regularization, and adaptive optimization. Our implementation supports this misspecified setting through counterfactual update probes, as detailed in App. F.18. Extending the guarantees to these general update sets remains an important direction.

REFERENCES

[1] Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pp. 557–563, New York, NY, USA, 2006. Association for Computing Machinery. ISBN 1595931341. doi: 10.1145/1132516.1132597. URL https://doi.org/10.1145/1132516.1132597.

[2] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cedric Renggli. The convergence of sparsified gradient methods. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/314450613369e0ee72d0da7f6fee773c-Paper.pdf.

[3] Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/2f37d10131f2a483a8dd005b3d14b0d9-Paper.pdf.

[4] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL https://arxiv.org/abs/2502.01456.

[5] 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.

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

[7] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290.

[8] William Johnson and Joram Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, 01 1984. doi: 10.1090/conm/026/737400.

[9] K J Joseph, Vamshi Teja R, Krishnakant Singh, and Vineeth N Balasubramanian. Submodular batch selection for training deep neural networks. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 2677–2683. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/372. URL https://doi.org/10.24963/ijcai.2019/372.

[10] Angelos Katharopoulos and Francois Fleuret. Not all samples are created equal: Deep learning with importance sampling. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2525–2534. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/katharopoulos18a.html.

[11] Kenji Kawaguchi and Haihao Lu. Ordered sgd: A new stochastic optimization framework for empirical risk minimization, 2020. URL https://arxiv.org/abs/1907.04371.

[12] Aybora Koksal and A. Aydin Alatan. Few-shot vision-language reasoning for satellite imagery via verifiable rewards, 2025. URL https://arxiv.org/abs/2507.21745.

[13] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention, 2023. URL https://arxiv.org/abs/2309.06180.

[14] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

[15] Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C. H. Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning, 2022a. URL https://arxiv.org/abs/2207.01780.

[16] Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 21314–21328. Curran Associates, Inc., 2022b. doi: 10.52202/068431-1549. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8636419dea1aa9fbd25fc4248e702da4-Paper-Conference.pdf.

[17] Jiate Liu, Yiqin Zhu, Kaiwen Xiao, QIANG FU, Xiao Han, Yang Wei, and Deheng Ye. RLTF: Reinforcement learning from unit test feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=hjYmsV6nXZ.

[18] 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, 2025. URL https://arxiv.org/abs/2503.20783.

[19] Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks, 2016. URL https://arxiv.org/abs/1511.06343.

[20] Haihao Lu, Robert Freund, and Vahab Mirrokni. Accelerating greedy coordinate descent methods. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3257–3266. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/lu18b.html.

[21] Michael Muehlebach and Michael Jordan. Continuous-time lower bounds for gradient-based algorithms. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7088–7096. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/muehlebach20a.html.

[22] Julie Nutini, Mark Schmidt, Issam Laradji, Michael Friedlander, and Hoyt Koepke. Coordinate descent converges faster with the gauss-southwell rule than random selection. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1632–1641, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/nutini15.html.

[23] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[24] Yujuan Pang, Jiaxin Li, Xin Sheng, Ran Peng, and Yong Ma. Beyond variance: Prompt-efficient rlvr via rare-event amplification and bidirectional pairing, 2026. URL https://arxiv.org/abs/2602.03452.

[25] Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale, 2023. URL https://arxiv.org/abs/2303.14186.

[26] Dmytro Perekrestenko, Volkan Cevher, and Martin Jaggi. Faster coordinate descent via adaptive importance sampling, 2017. URL https://arxiv.org/abs/1703.02518.

[27] Atal Narayan Sahu, Aritra Dutta, Ahmed M. Abdelmoniem, Trambak Banerjee, Marco Canini, and Panos Kalnis. Rethinking gradient sparsification as total error minimization, 2021. URL https://arxiv.org/abs/2108.00951.

[28] Vatsal Shah, Xiaoxia Wu, and Sujay Sanghavi. Choosing the sample with lowest loss makes sgd robust, 2020. URL https://arxiv.org/abs/2001.03316.

[29] 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.

[30] 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*, EuroSys '25, pp. 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

[31] Shaohuai Shi, Kaiyong Zhao, Qiang Wang, Zhenheng Tang, and Xiaowen Chu. A convergence analysis of distributed sgd with communication-efficient gradient sparsification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 3411–3417. International Joint Conferences on Artificial Intelligence Organization, 7 2019. doi: 10.24963/ijcai.2019/473. URL https://doi.org/10.24963/ijcai.2019/473.

[32] Parshin Shojaee, Aneesh Jain, Sindhu Tipirneni, and Chandan K. Reddy. Execution-based code generation using deep reinforcement learning, 2023. URL https://arxiv.org/abs/2301.13816.

[33] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

[34] Xinyu Tang, Zhenduo Zhang, Yurou Liu, Xin Zhao, zujie wen, Zhiqiang Zhang, and JUN ZHOU. Towards high data efficiency in reinforcement learning with verifiable reward. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.),

International Conference on Learning Representations, volume 2026, pp. 32611–32635, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/374050dc3f211267bd6bf0ea24eae184-Paper-Conference.pdf.

[35] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

[36] 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, 2022. URL https://arxiv.org/abs/2211.14275.

[37] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025. URL https://arxiv.org/abs/2504.20571.

[38] Andre Wibisono, Ashia C. Wilson, and Michael I. Jordan. A variational perspective on accelerated methods in optimization. *Proceedings of the National Academy of Sciences*, 113(47), November 2016. ISSN 1091-6490. doi: 10.1073/pnas.1614734113. URL http://dx.doi.org/10.1073/pnas.1614734113.

[39] Jianghao Wu, Jianfei Cai, Weiqiang Wang, Jin Ye, Daniel F. Schmidt, and Yasmeen George. Single-rollout hidden-state dynamics for training-free rlvr data selection, 2026. URL https://arxiv.org/abs/2605.28631.

[40] Zhiping Xiao, Yusheng Zhao, Qixin Zhang, Jiaye Xie, Wanjia Zhao, Weizhi Zhang, Xiao Luo, Philip S. Yu, and Ming Zhang. Sample lottery: Unsupervised discovery of critical instances for LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=76OZBE4Rb6.

[41] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[42] Hao Yi, Yulan Hu, Xin Li, Sheng Ouyang, Lizhong Ding, and Yong Liu. Learn more with less: Uncertainty consistency guided query selection for rlvr. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 18618–18634, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/1fb18f013463e77021b7ac4bdc374b6d-Paper-Conference.pdf.

[43] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, juncai liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole

Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

[44] Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025. URL https://arxiv.org/abs/2504.05118.

[45] Yunjian Zhang, Sudong Wang, Yang Li, Peiran Xu, Conghao Zhou, Xiaoyue Ma, Jianing Li, and Yao Zhu. Resource-efficient reinforcement for reasoning large language models via dynamic one-shot policy refinement, 2026. URL https://arxiv.org/abs/2602.00815.

[46] Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling for regularized loss minimization. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1–9, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/zhaoa15.html.

[47] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

[48] Erle Zhu, Dazhi Jiang, Yuan Wang, Xujun Li, Jiale Cheng, Yuxian Gu, Yilin Niu, Aohan Zeng, Jie Tang, Minlie Huang, and Hongning Wang. Data efficient RLVR via off-policy influence guidance. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 46167–46192, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2141. URL https://aclanthology.org/2026.acl-long.2141/.

[49] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

[50] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, pp. 928–935. AAAI Press, 2003. ISBN 1577351894.

A Declaration of our LLM Usage

We used generative AI tools to polish the manuscript, discuss and refine proof ideas, and iteratively maintain the codebase. The authors specified and reviewed all implementation details and testing requirements, and validated AI-assisted code changes through extensive production-environment testing, including API-correctness, smoke, and stress tests. The authors reviewed all AI-assisted work and take full responsibility for the final text, claims, proofs, code, and experimental results.

B Reproducibility Statement

The implementation and experimental workflow are described in Sec. 4, including candidate construction, leave-one-out scoring, gradient projection, and update commitment. The model and data configuration, optimization recipe, distributed execution, projection settings, evaluation protocol, and complete run matrix are documented in Tabs. 2–7. We disclose the primary results in Tab. 1 and the full set of secondary evaluation results in the appendix tables, together with seed-level uncertainty and training diagnostics. The source code used to run the experiments is released as part of the supplementary review materials.

C Pseudo Codes for the Batch Greedy Algorithms

D Finite-Sample Approximation for the Greedy Oracle

Algorithm 1 Batch Greedy-Prompt Tournament

Require: d_0, B, G, θ_0, and $\{\mu_k\}_{k=0}^{K-1}$.

1: **for** $k = 0, \dots, K-1$ **do**

2: Sample $x_0, \dots, x_{B-1} \overset{i.i.d.}{\sim} d_0$.

3: **for** $i = 0, \dots, B-1$ **do**

4: Sample $y_{i,0}, \dots, y_{i,G-1} \overset{i.i.d.}{\sim} \pi_{\theta_k}(\cdot \mid x_i)$ and compute their rewards.

5: Construct the candidate

$$\widehat{\nabla J_{x_i}}(\theta_k) \leftarrow \frac{1}{G} \sum_{j=0}^{G-1} R(x_i, y_{i,j}) \sum_{h=0}^{H-1} \nabla \log \pi_{\theta_k}(y_{i,j,h} \mid x_i, y_{i,j,<h}).$$

6: **end for**

7: Construct the leave-one-out estimation for each i,

$$\widehat{\nabla J}_i(\theta_k) \leftarrow \frac{1}{B-1} \sum_{\substack{i'=0 \\ i' \neq i}}^{B-1} \widehat{\nabla J_{x_{i'}}}(\theta_k).$$

8: Select and commit

$$\widehat{i^*} \leftarrow \underset{i \in \{0,\dots,B-1\}}{\arg\max} \left\langle \widehat{\nabla J_{x_i}}(\theta_k), \widehat{\nabla J}_i(\theta_k) \right\rangle, \qquad \theta_{k+1} \leftarrow \theta_k + \mu_k \widehat{\nabla J_{x_{\widehat{i^*}}}}(\theta_k).$$

9: **end for**

Algorithm 2 Normalized Batch Greedy-Prompt Tournament

Require: The inputs and candidate construction of Alg. 1.

1: **for** $k = 0, \dots, K-1$ **do**

2: Construct $\{\widehat{\nabla J_{x_i}}(\theta_k), \widehat{\nabla J}_i(\theta_k)\}_{i=0}^{B-1}$ as in Alg. 1, and set

$$\widehat{\nabla J}(\theta_k) \leftarrow \frac{1}{B} \sum_{i=0}^{B-1} \widehat{\nabla J_{x_i}}(\theta_k).$$

3: Select

$$\widehat{i^*} \leftarrow \underset{i \in \{0,\dots,B-1\}}{\arg\max} \left\langle \widehat{\nabla J_{x_i}}(\theta_k), \widehat{\nabla J}_i(\theta_k) \right\rangle.$$

4: Normalize

$$\widetilde{\mu}_k \leftarrow \mu_k \cdot \frac{\left\langle \widehat{\nabla J}(\theta_k), \widehat{\nabla J}(\theta_k) \right\rangle}{\left\langle \widehat{\nabla J_{x_{\widehat{i^*}}}}(\theta_k), \widehat{\nabla J}(\theta_k) \right\rangle},$$

5: and commit

$$\theta_{k+1} \leftarrow \theta_k + \widetilde{\mu}_k \widehat{\nabla J_{x_{\widehat{i^*}}}}(\theta_k).$$

6: **end for**

E Preliminaries

E.1 RLVR as a Contextual Bandit and Policy Gradient

During the post-training stage of large language models, RLVR can be formulated as a contextual bandit specified by the tuple $(\mathcal{X}, d_0, \mathcal{V}, R)$, where $\mathcal{X}$ denotes the prompt space, $d_0 \in \Delta(\mathcal{X})$ denotes the prompt distribution associated with a given task (e.g., code generation [Le et al., 2022a] or mathematical problem solving [Uesato et al., 2022]), $\mathcal{V}$ denotes the response space induced by the token vocabulary $\mathcal{V}$, and $R : \mathcal{X} \times \mathcal{V} \to \mathbb{R}$ is a verifier-based reward function[2] that maps a prompt-response pair (x, y) to a scalar value reflecting the quality of the generated response. We further assume that R is bounded so that $\forall (x, y) \in \mathcal{X} \times \mathcal{V}$, $R(x, y) \in [0, R_{max}]$ for convenience of our subsequent analyses.

The objective of RLVR is to optimize the policy $\pi_\theta : \mathcal{X} \to \Delta(\mathcal{V})$, parameterized by a base model, so as to maximize the *expected return* $J(\theta) = \mathbb{E}_{x \sim d_0,\, y \sim \pi_\theta(\cdot|x)}[R(x, y)]$. For convenience, we use the shorthand $\mathbb{E}_{\pi_\theta}[\cdot] := \mathbb{E}_{x \sim d_0,\, y \sim \pi_\theta(\cdot|x)}[\cdot]$ for this expectation. The policy is autoregressively factorized according to the next-token prediction mechanism: $\pi_\theta(y|x) = \prod_{h=0}^{|y|-1} \pi_\theta(y_h|x, y_{<h})$. Under the vanilla *REINFORCE* algorithm, the policy is learned via the standard gradient-ascent update $\theta_{k+1} \leftarrow \theta_k + \eta_k \nabla J(\theta_k)$, where

$$\nabla J(\theta) = \mathbb{E}_{\pi_\theta} \left[R(x, y) \sum_{h=0}^{|y|-1} \nabla_\theta \log \pi_\theta(y_h|x, y_{<h}) \right], \tag{16}$$

and η_k is the step size. In principle, one can always replace $R(x, y)$ in Eq. 16 with the centered return $R(x, y) - \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[R(x, y) \,|\, x]$, which leaves the expected policy gradient unbiased with reduced variance. This formulation provides a common foundation for a broad family of policy optimization algorithms, including PPO [Ziegler et al., 2020; Stiennon et al., 2022; Ouyang et al., 2022], GRPO [Shao et al., 2024], and various X-PO variants [Liu et al., 2025; Yu et al., 2025; Yue et al., 2025; Zheng et al., 2025].

E.2 Gradient Flow & Discretization

Learning Rates and Their Limits. As discussed in Sec. 1, we use gradient flows to isolate the first-order behavior of policy optimization while abstracting away the higher-order discretization terms that would otherwise complicate the subsequent analysis.

[2]The reward is computed by a *verifier* and may incorporate additional penalties (e.g., $R(x, y) = R_{\text{acc}}(x, y) + \lambda \cdot R_{\text{overlong}}(x, y)$ that incorporates verified accuracy and overlong penalty).

Let $f : \Theta \to \mathbb{R}$ denote the target function to be maximized. We first prescribe a family of learning-rate schedules

$$\left\{ \left\{ \mu_k^{(m)} \right\}_{k=0}^{\infty} \right\}_{m=1}^{\infty}, \tag{17}$$

where m indexes the particular manner in which the discrete schedules approach a vanishing-mesh limit, rather than as the optimization-step index. For each fixed m, the index k enumerates the updates within that schedule. The family is admissible when

$$\sup_{k \geq 0} \mu_k^{(m)} \longrightarrow 0 \qquad \text{as } m \to \infty, \tag{18}$$

and its cumulative learning-rate budget converges to a prescribed horizon $T \in (0, \infty]$. For example, with $K_0 \in \mathbb{N}_{>0}$, one admissible family such that $T < \infty$ can be

$$\mu_k^{(m)} = \frac{1}{m} \mathbb{I}\left[k < mK_0\right], \qquad \forall k \geq 0, \tag{19}$$

for which

$$\sup_k \mu_k^{(m)} = \frac{1}{m} \longrightarrow 0, \qquad \sum_{k=0}^{\infty} \mu_k^{(m)} = \sum_{k=0}^{mK_0 - 1} \mu_k^{(m)} \longrightarrow T = K_0. \tag{20}$$

Flows induced by Limits under the Common Learning Rates. Note that the index m specifies one prescribed manner in which the entire learning-rate schedule approaches the zero sequence. This manner of approaching the vanishing-mesh limit is **fixed** before any two update rules are compared. In particular, for two algorithms A and B, we consider, for every m, the discrete updates

$$\theta_{k+1}^{\mathsf{A},(m)} = \theta_k^{\mathsf{A},(m)} + \mu_k^{(m)} g_k^{\mathsf{A},(m)}, \qquad \theta_{k+1}^{\mathsf{B},(m)} = \theta_k^{\mathsf{B},(m)} + \mu_k^{(m)} g_k^{\mathsf{B},(m)}, \tag{21}$$

with the common initialization

$$\theta_0^{\mathsf{A},(m)} = \theta_0^{\mathsf{B},(m)} = \theta_0. \tag{22}$$

Thus, at every fixed pair (m, k), the two algorithms use exactly the same learning rate $\mu_k^{(m)}$. They may differ only through the update controls $g_k^{\mathsf{A},(m)}$ and $g_k^{\mathsf{B},(m)}$. We then define the common cumulative *learning-rate clock*

$$t_k^{(m)} = \sum_{j=0}^{k-1} \mu_j^{(m)}. \tag{23}$$

Let $\theta^{\mathsf{A},(m)}(t)$ and $\theta^{\mathsf{B},(m)}(t)$ denote the piecewise-linear interpolations of the corresponding iterates on this common time grid. Explicitly, for

$$t_k^{(m)} \leq t < t_{k+1}^{(m)}, \tag{24}$$

we set

$$\theta^{\mathsf{A},(m)}(t) = \theta_k^{\mathsf{A},(m)} + \frac{t - t_k^{(m)}}{\mu_k^{(m)}} \left[\theta_{k+1}^{\mathsf{A},(m)} - \theta_k^{\mathsf{A},(m)} \right], \tag{25}$$

and define $\theta^{\mathsf{B},(m)}(t)$ analogously. Whenever these interpolations converge locally uniformly as $m \to \infty$, we write

$$\theta_t^{(c),\mathsf{A}} = \lim_{m \to \infty} \theta^{\mathsf{A},(m)}(t), \qquad \theta_t^{(c),\mathsf{B}} = \lim_{m \to \infty} \theta^{\mathsf{B},(m)}(t). \tag{26}$$

A comparison between the two flows always means a comparison between $\theta_t^{(c),\mathsf{A}}$ and $\theta_t^{(c),\mathsf{B}}$ at the same value of the common cumulative learning-rate coordinate t.

This definition therefore takes a synchronous limit. We first fix one family

$$\left\{ \left\{ \mu_k^{(m)} \right\}_{k=0}^{\infty} \right\}_{m=1}^{\infty}, \tag{27}$$

use that same family to generate the discrete trajectories of all algorithms under comparison, and only then let $m \to \infty$. For example, the two families

$$\mu_k^{(m)} = \frac{1}{m} \mathbb{I}\left[k < \lfloor mK_0 \rfloor\right] \qquad \text{and} \qquad \widetilde{\mu}_k^{(m)} = \frac{1}{m^2} \mathbb{I}\left[k < \lfloor m^2 K_0 \rfloor\right] \tag{28}$$

both have vanishing mesh and cumulative horizon K_0. Either family may be used to define the common limit. However, assigning the first family only to A and the second family only to B would not constitute the comparison adopted here: it would vary both the update rule and its discrete approximation. Although the two schemes may converge to comparable limits under additional numerical consistency conditions, such a comparison no longer isolates the effect of the update control. We therefore require the same prescribed family at every m.

On the Choice of Comparison. Our definition is one among several legitimate comparison protocols in optimization. Online gradient methods are commonly analyzed under a prescribed learning-rate sequence shared by the updates under consideration [Zinkevich, 2003]. Analyses of greedy and random coordinate selection often hold the update formula fixed and change only the selection rule [Nutini et al., 2015]; when the coordinates have different smoothness constants, a common rule such as $\frac{1}{L_i}$, rather than a common numerical step size, may instead be used [Lu et al., 2018]. Other work studies continuous-time dynamics directly and regards time dilation as part of the dynamical description [Wibisono et al., 2016], while continuous-time complexity results may impose an explicit time-normalization constraint to prevent arbitrary acceleration through a reparameterization of time [Muehlebach & Jordan, 2020].

Our Justification. Our common-family convention is particularly natural for the present problem. First, the prompt-wise gradients are controlled by uniform boundedness and smoothness assumptions across x. Boundedness controls the displacement produced by one update, while the common prompt-wise smoothness constant gives comparable smoothness-derived safe learning rates across candidate prompts. Thus, changing x does not require assigning a fundamentally different learning rate scale to each candidate.

Second, the object of interest is the effect of the gradient-control rule itself. Holding the learning-rate family fixed follows the standard comparison in which random, greedy, and approximate-greedy selection rules are inserted into a common update mechanism [Nutini et al., 2015; Lu et al., 2018]. The resulting difference between the limiting trajectories is consequently induced by the selected controls rather than by an independently tuned time coordinate.

Third, this convention exposes the most immediate degeneracy of a same-clock comparison. If one candidate were merely

$$\nabla J_x(\theta) = C\nabla J(\theta) \tag{29}$$

for some $C > 1$, then selecting it would produce an apparent improvement that is entirely explained by a scalar acceleration along the population-gradient direction.

Fortunately, our subsequent analysis **does** separate this parallel rescaling from the geometry of the selected update by considering the exact inner-ratio-normalized control. Under this normalization, the component parallel to ∇J agrees with that of the vanilla flow, so any remaining difference is generated by the winner's orthogonal component. Both the theoretical argument and the corresponding experiments therefore distinguish genuine steering from the trivial replacement of the learning rate by C; moreover, the winner definition is used repeatedly in controlling the evolution induced by this orthogonal component.

The Definition of Clocks as an Extension. More generally, selecting a convention for comparing continuous-time algorithms amounts to selecting a common clock. If θ_t satisfies

$$\frac{\mathrm{d}\theta_t}{\mathrm{d}t} = g(\theta_t), \tag{30}$$

then a strictly increasing time change $t = \tau(s)$ gives

$$\frac{\mathrm{d}\theta_{\tau(s)}}{\mathrm{d}s} = \dot{\tau}(s)g(\theta_{\tau(s)}). \tag{31}$$

The two equations may trace the same geometric orbit at different speeds, while their objective values at the same numerical time need not agree. Time-dilation and renormalized-flow analyses make this distinction explicit [Wibisono et al., 2016]. In this work, the clock is fixed by the common cumulative learning-rate budget, and all flow comparisons are understood with respect to that clock.

Carathéodory Conditions. Consider a possibly nonsmooth vector field $\dot{\theta}_t = g(t, \theta_t)$. The Carathéodory conditions require $g(\cdot, \theta)$ to be measurable in t, $g(t, \cdot)$ to be continuous for almost every t, and $\|g(t, \theta)\|_2$ to have a locally integrable bound. A Carathéodory solution is an absolutely

continuous trajectory satisfying

$$\theta_t = \theta_0 + \int_0^t g(s, \theta_s)\ \mathrm{d}s,$$

and therefore the differential equation almost everywhere. This formulation permits measurable switching among selected prompts without requiring a continuous winner. A locally integrable Lipschitz bound in θ additionally guarantees uniqueness.

E.3 Other Related Work

Ultimately, we would like to thank the studies that have motivated our work, provided technical precedents, or are closely related to our problem formulation and proposed solutions.

RLVR for LLMs. Under the formulation in Sec. E.1, RLVR can further employ task-specific learning signals, such as symbolic answer verification or executable unit tests, which are exploited by early code-generation systems. For instance, CodeRL [Le et al., 2022b] trains an actor and a learned critic from unit-test outcomes, PPOCoder [Shojaee et al., 2023] applies PPO with execution and structure-based feedback, and RLTF [Liu et al., 2023] performs online policy optimization using multi-granularity compiler and unit-test signals.

In terms of mathematical reasoning, DeepSeekMath [Shao et al., 2024] introduced GRPO algorithms, which estimates relative advantages from multiple responses to the same problem without a learned critic; Tulu 3 [Lambert et al., 2025] incorporated verifiable rewards into a fully open post-training recipe; and DeepSeek-R1 [Guo et al., 2025] showed that large-scale rule-reward training can elicit long reasoning, reflection, and self-verification.

For long-horizon learning, Kimi k1.5 [Team et al., 2025] studies long-context and multimodal RL scaling, Open-Reasoner-Zero [Hu et al., 2025] shows that a comparatively minimal PPO configuration can reproduce large-scale reasoning improvements, and DAPO [Yu et al., 2025] introduces decoupled clipping, dynamic sampling, and token-level loss aggregation for stable large-scale RLVR. PRIME [Cui et al., 2025] instead learns implicit process rewards from outcome-labeled rollouts, connecting outcome-based RLVR with denser process-level credit assignment.

Few-shot RLVR Variants. While learning from full-bank data is standard practice, recent studies have demonstrated that policy optimization based-on few-shot samples can achieve exceptional training outcomes.

Wang et al. [2025] shows that the generalization across the entire prompt bank can be achieved by simply fitting the policy to a single carefully curated problem. Followed by this, the few-shot vision-language RLVR [Koksal & Alatan, 2025] extends the one-shot phenomenon to the scenarios including classification, visual Q&A tasks, and grounding problems.

Concerning data efficiency, there are other studies that focus on the problem difficulty, diversity, influence, uncertainty, hidden-state dynamics, or exploration potential [Tang et al., 2026; Yi et al., 2026; Zhu et al., 2026; Wu et al., 2026]. They respect the problem formulation of prompt selection, but typically select a static subset for their actual policy optimization.

Gradient Selection. A related line of work studies how to select an update from a structured collection of gradient information. The central question is not how to estimate one prescribed gradient more accurately, but which coordinate, example, block, or candidate should determine the next update.

Classical coordinate descent selects a component of a single gradient. Gauss–Southwell favors a large partial derivative, while Gauss–Southwell–Lipschitz accounts for coordinate-wise smoothness. Random and greedy rules are typically compared under a shared step size, a common coordinate-dependent rule such as $1/L_i$, or exact one-dimensional minimization [Nutini et al., 2015]. Accelerated greedy coordinate methods extend this shared-update framework and establish iteration-complexity or objective-suboptimality guarantees [Lu et al., 2018].

Importance sampling instead changes how examples or coordinates are sampled, using gradient norms, smoothness constants, residuals, or duality gaps. The selected update is usually reweighted to preserve an unbiased or otherwise controlled estimator of the original gradient [Zhao & Zhang, 2015; Perekrestenko et al., 2017; Katharopoulos & Fleuret, 2018]. Other methods deliberately

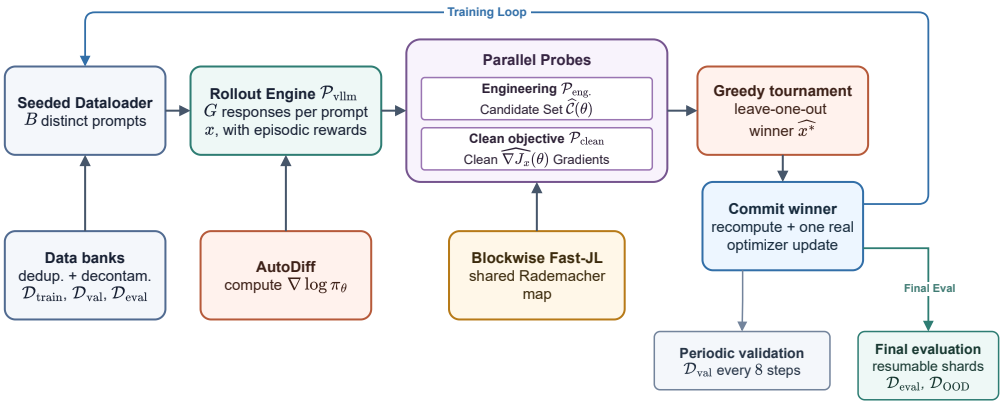


Figure 2: **Overview of the greedy leave-one-out training workflow.** For each of the B sampled prompts, G responses are generated to construct a candidate update and estimate its clean gradient. Each candidate is scored against the leave-one-out mean of the remaining clean-gradient estimates, and the highest-scoring candidate is selected. All candidate probes are rolled back, and only the selected update is committed to the optimizer.

bias the training stream according to loss, uncertainty, robustness, or diversity, including online batch selection, Active Bias, Ordered SGD, low-loss selection, and submodular batch construction [Loshchilov & Hutter, 2016; Chang et al., 2017; Kawaguchi & Lu, 2020; Shah et al., 2020; Joseph et al., 2019].

Gradient sparsification selects coordinates from an already aggregated gradient to reduce communication. Its guarantees rely on contraction, residual accumulation, or error feedback rather than on choosing among full candidate updates [Alistarh et al., 2018; Shi et al., 2019; Sahu et al., 2021]. More broadly, these methods compare discrete iterations, processed examples, or communication rounds under a specified update rule. Continuous-time comparisons are less common and remain meaningful only after fixing a common clock.

Our setting instead selects among full prompt-induced policy gradients and does not reweight the winner into an unbiased population-gradient estimator. The winner maximizes its estimated first-order improvement on a separate population objective. This introduces a specific degeneracy: selection may appear beneficial only because the winner induces a larger effective step along the population gradient. Our inner-ratio control removes this parallel-speed advantage while preserving the selected direction, thereby isolating the contribution of its orthogonal geometry.

Component	Configuration
Hardware	One node with 8× NVIDIA H800 80 GB GPUs
Model	`Qwen/Qwen3-8B-Base`
Adaptation	LoRA rank 32, $\alpha = 256$, targeting `all-linear` modules
Training data	`YouJiacheng/DAPO-Math-17k-dedup`
Data split	Three disjoint banks of 2,048 prompts for training, monitoring, and in-distribution final evaluation
Decontamination	Exact-match and word-5-gram Jaccard filtering at threshold 0.8 against all OOD evaluation banks
Sequence limits	1,536 prompt tokens and 4,096 response tokens; left truncation
Precision	BF16 with FlashAttention 2, padding removal, and gradient checkpointing
Regularization	Attention, embedding, and residual dropout set to zero
Prompt template	Empty

Table 2: Model and data configuration.

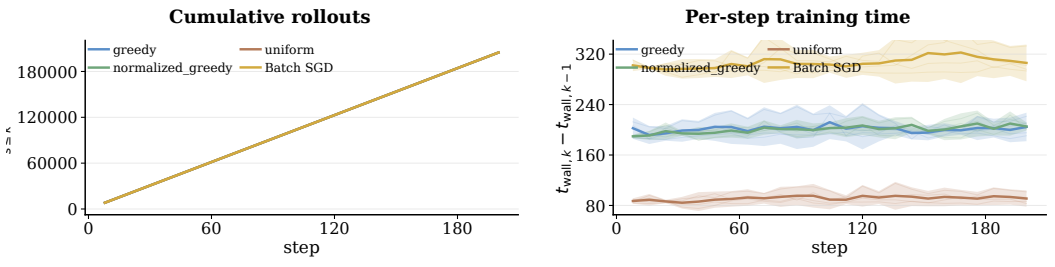
Component	Configuration
Training horizon	200 optimizer steps
Estimator	Baseline-free REINFORCE with fixed divisor $BGL_{\max}$
Prompt batch	$B \in \{32, 64, 128\}$, as specified by the experiment matrix
Rollouts	$G = 16$ responses per prompt
Optimizer	SGD with learning rate 0.25, momentum 0, and weight decay 0
Gradient clipping	Threshold 10^9, serving as a finite effective no-op
LR schedule	10-step warmup followed by a half-cosine decay to 0.1 of the peak learning rate
PPO epochs	One
PPO mini-batch	B, yielding one committed optimizer update per training step
Actor microbatch	One per GPU for greedy-family probes and commits; two otherwise
Log-prob microbatch	$\min(256, 16B)$ per GPU
Loss aggregation	`seq-mean-token-sum`
KL and entropy	KL reward, KL loss, entropy bonus, and rollout importance sampling disabled
Reward	DAPO continuous Math-Verify reward
Overlong penalty	Buffer length 819, penalty factor 1.0

Table 3: Training objective and optimization configuration.

F Other Experimental Details

F.1 Other Implementation Details

F.2 Training Traces



(a) Cumulative number of generated rollouts.

(b) Wall-clock seconds per training step.

Figure 3: Conventional training traces for the four selection methods at $B = 64$. Curves show seed means and pointwise Student-t 95% confidence intervals. Confidence bands are clipped to panel limits where needed for stable visualization; the computed intervals are unchanged.

Component	Configuration
Sampling	Enabled with temperature 1, top-$p = 1$, and top-$k = -1$
Generation microbatch	$\min(128, B)$ distinct prompts
vLLM execution	Synchronous generation with cache-engine release and prefix caching enabled
Maximum vLLM batch	131,072 tokens and 1,024 sequences
vLLM memory fraction	0.80
Tensor parallelism	TP $= 2$ for vLLM generation
Sequence parallelism	Ulysses SP $= 2$ for the training actor and reference model
Actor data parallelism	DP $= 4$ across eight GPUs
FSDP	Full actor FSDP with parameter and optimizer offload; reference-parameter offload enabled
Ray execution	Local initialization on one node with eight GPUs

Table 4: Generation and distributed-execution configuration.

Component	Configuration
Probe replicas	Eight isolated one-GPU replica groups
Probe dispatch	All candidates dispatched in one distributed RPC
Probe microbatch	One candidate PPO microbatch per GPU
Forward chunking	Four probe microbatches fused per forward chunk
Replica state	Frozen BF16 base model with gradient checkpointing
Replica lifecycle	Synchronized before probing and parked on CPU between probe stages
Projection	Run-fixed implicit Rademacher map implemented by `fast_jl`
Projection dimension	$m = 262{,}144$
Projection partitioning	32 independent round-robin chunks of 8,192 coordinates each
Maximum projected rows	32 rows per `fast_jl` launch
Projection normalization	Raw sign sketches with inner products divided by m
Tournament device	CUDA
Greedy warmup	Zero steps
Tournament logging	Every training step

Table 5: Probe and projection configuration.

Entries are means ± two-sided 95% confidence intervals across available seeds. Black bold marks the best mean and gray bold the second-best distinct mean within each comparison block; arrows give the preferred direction. Student-t critical values are used for $n = 2, 3, 4$.

F.3 Final accuracy

Table 8: Final accuracy (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro	micro
Selection methods at $B = 64$									
greedy	12.08 ± 1.38	11.72 ± 1.13	**52.99 ± 3.54**	**32.28 ± 3.17**	**61.22 ± 1.83**	**22.35 ± 0.22**	**39.94 ± 2.01**	**33.23 ± 1.31**	**36.88 ± 2.37**
normalized_greedy	**12.47 ± 1.10**	**12.60 ± 3.19**	51.60 ± 1.95	31.50 ± 0.63	60.00 ± 1.05	21.08 ± 0.65	39.48 ± 0.82	32.68 ± 0.68	36.08 ± 0.62
uniform	10.31 ± 4.83	8.54 ± 3.75	41.95 ± 10.01	24.98 ± 6.99	53.47 ± 8.76	18.18 ± 5.02	33.06 ± 8.52	27.21 ± 6.76	29.88 ± 7.34
Batch SGD	11.22 ± 2.06	9.32 ± 1.37	43.61 ± 1.90	25.45 ± 1.77	53.80 ± 2.37	17.75 ± 0.98	34.12 ± 1.91	27.90 ± 1.54	30.38 ± 1.80
greedy batch-size ablation									
greedy, $B = 32$	**12.08 ± 2.30**	**12.06 ± 1.19**	50.72 ± 3.17	30.92 ± 0.62	60.46 ± 0.98	21.38 ± 0.31	39.23 ± 0.52	32.41 ± 0.60	35.77 ± 0.50
greedy, $B = 64$	**12.08 ± 1.38**	11.72 ± 1.13	52.99 ± 3.54	32.28 ± 3.17	**61.22 ± 1.83**	22.35 ± 0.22	39.94 ± 2.01	**33.23 ± 1.31**	36.88 ± 2.37
greedy, $B = 128$	11.46 ± 3.48	9.90 ± 2.17	**54.02 ± 2.32**	**32.86 ± 1.92**	61.18 ± 0.86	**22.54 ± 1.03**	**40.13 ± 1.10**	33.16 ± 0.92	**37.25 ± 1.37**

F.4 Pass@32

Table 9: Pass@32 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	41.67 ± 6.85	35.00 ± 10.16	**92.50 ± 5.63**	75.83 ± 3.65	78.20 ± 2.11	44.21 ± 7.22	70.14 ± 2.35	62.51 ± 3.32
normalized_greedy	45.83 ± 6.67	**40.83 ± 10.04**	**92.50 ± 3.25**	**77.00 ± 1.53**	79.15 ± 2.21	45.86 ± 3.36	70.66 ± 0.52	64.55 ± 3.22
uniform	**46.67 ± 4.33**	35.00 ± 6.85	91.88 ± 5.97	76.27 ± 1.53	81.45 ± 2.75	**49.72 ± 3.55**	**71.40 ± 1.18**	**64.63 ± 2.76**
Batch SGD	45.83 ± 13.95	35.83 ± 5.08	90.62 ± 5.01	76.67 ± 1.44	**81.65 ± 1.26**	47.61 ± 1.47	71.22 ± 1.67	64.21 ± 1.08
greedy batch-size ablation								
greedy, $B = 32$	39.17 ± 5.08	**40.00 ± 7.50**	91.25 ± 2.30	**76.68 ± 0.75**	**79.75 ± 1.97**	**47.43 ± 3.13**	**70.55 ± 2.06**	**63.55 ± 1.54**
greedy, $B = 64$	**41.67 ± 6.85**	35.00 ± 10.16	**92.50 ± 5.63**	75.83 ± 3.65	78.20 ± 2.11	44.21 ± 7.22	70.14 ± 2.35	62.51 ± 3.32
greedy, $B = 128$	35.00 ± 3.06	30.83 ± 7.96	91.25 ± 2.30	74.87 ± 3.68	77.50 ± 1.44	41.27 ± 3.46	69.40 ± 1.45	60.02 ± 1.74

Component	Configuration
Checkpointing	Every four completed steps; retain the latest checkpoint
Periodic evaluation	Every eight completed steps
Monitor bank	All 2,048 held-out monitor prompts, with one response per prompt
Final ID evaluation	All 2,048 prompts in the held-out in-distribution bank
Final OOD evaluation	MATH-500, AMC-2023, AIME-2024, AIME-2025, Minerva Math, and Olympiad-Bench
Final responses	32 responses per prompt
Final-evaluation batch	$16\min(128, B)$ response sequences per generation batch
Decoding	Temperature 1, top-$p = 1$, top-$k = -1$, matching training and periodic evaluation except for response multiplicity
Trainer scheduling	Native save and test frequencies disabled; the plugin owns checkpointing and evaluation
Pre-training validation	Disabled
Resume validation	Recipe, source, topology, parameter, optimizer, scheduler, scaler, and RNG fingerprints

Table 6: Checkpointing and evaluation configuration.

Component	Configuration
Fixed setting	Qwen3-8B-Base, REINFORCE, SGD, and seeds $\{0, 1, 2, 3\}$
Selector study	$B = 64$ with `greedy`, `greedy_inner_ratio`, `uniform`, and `batch_mean`
Batch-size study	`greedy` with $B \in \{32, 64, 128\}$
Unique runs	$16 + 12 - 4 = 24$
Comparison axis	Matched optimizer-step index
Determinism	Production-performance mode; the master seed also determines rollout, projection, and evaluation seeds
Provenance	Each launch records resolved model and dataset revisions, file digests, source build identifier, and all Hydra overrides

Table 7: Formal experiment matrix and provenance controls.

F.5 AVERAGE@32

Table 10: Average@32 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	12.08 ± 1.38	11.72 ± 1.13	**52.99 ± 3.54**	**32.28 ± 3.17**	**61.22 ± 1.83**	**22.35 ± 0.22**	**39.94 ± 2.01**	**33.23 ± 1.31**
normalized_greedy	**12.47 ± 1.10**	**12.60 ± 3.19**	51.60 ± 1.95	31.50 ± 0.63	60.00 ± 1.05	21.08 ± 0.65	39.48 ± 0.82	32.68 ± 0.68
uniform	10.31 ± 4.83	8.54 ± 3.75	41.95 ± 10.01	24.98 ± 6.99	53.47 ± 8.76	18.18 ± 5.02	33.06 ± 8.52	27.21 ± 6.76
Batch SGD	11.22 ± 2.06	9.32 ± 1.37	43.61 ± 1.90	25.45 ± 1.77	53.80 ± 2.37	17.75 ± 0.98	34.12 ± 1.91	27.90 ± 1.54
greedy batch-size ablation								
greedy, $B = 32$	**12.08 ± 2.30**	**12.06 ± 1.19**	50.72 ± 3.17	30.92 ± 0.62	60.46 ± 0.98	21.38 ± 0.31	39.23 ± 0.52	32.41 ± 0.60
greedy, $B = 64$	**12.08 ± 1.38**	11.72 ± 1.13	52.99 ± 3.54	32.28 ± 3.17	**61.22 ± 1.83**	22.35 ± 0.22	39.94 ± 2.01	**33.23 ± 1.31**
greedy, $B = 128$	11.46 ± 3.48	9.90 ± 2.17	**54.02 ± 2.32**	**32.86 ± 1.92**	61.18 ± 0.86	**22.54 ± 1.03**	**40.13 ± 1.10**	33.16 ± 0.92

F.6 CONSISTENCY@32

Table 11: Consistency@32 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	**31.28 ± 6.40**	**32.11 ± 7.53**	**61.11 ± 5.07**	**46.03 ± 5.26**	**80.00 ± 3.72**	**50.41 ± 5.23**	**57.09 ± 4.64**	**51.15 ± 4.99**
normalized_greedy	29.01 ± 1.96	29.92 ± 0.82	57.91 ± 3.62	43.72 ± 2.15	77.37 ± 1.22	44.61 ± 4.81	54.47 ± 2.15	48.14 ± 2.21
uniform	23.59 ± 8.05	22.53 ± 5.89	47.09 ± 10.73	35.18 ± 8.65	66.42 ± 13.01	34.69 ± 11.01	45.02 ± 10.87	39.22 ± 9.64
Batch SGD	23.75 ± 1.67	24.17 ± 2.14	49.00 ± 3.19	35.71 ± 2.07	67.85 ± 3.37	35.45 ± 2.49	46.65 ± 2.08	40.37 ± 2.13
greedy batch-size ablation								
greedy, $B = 32$	27.27 ± 1.38	28.57 ± 1.74	56.60 ± 3.21	42.64 ± 0.94	76.96 ± 0.49	44.84 ± 1.07	53.53 ± 1.81	47.20 ± 0.93
greedy, $B = 64$	**31.28 ± 6.40**	32.11 ± 7.53	61.11 ± 5.07	46.03 ± 5.26	80.00 ± 3.72	50.41 ± 5.23	57.09 ± 4.64	51.15 ± 4.99
greedy, $B = 128$	30.60 ± 3.95	**32.68 ± 4.71**	**61.97 ± 3.10**	**47.56 ± 3.01**	**80.87 ± 1.54**	**52.24 ± 7.99**	**57.41 ± 2.42**	**51.91 ± 3.63**

F.7 MAJORITY-VOTE ACCURACY@32

Table 12: Majority-vote accuracy@32 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	20.00 ± 4.33	**22.50 ± 2.65**	65.00 ± 3.25	**47.29 ± 4.79**	68.25 ± 3.22	27.39 ± 1.55	49.78 ± 4.82	42.89 ± 1.49
normalized_greedy	20.00 ± 4.33	**22.50 ± 10.04**	**69.38 ± 3.81**	**49.17 ± 1.79**	**69.30 ± 1.18**	27.39 ± 1.82	**51.67 ± 1.71**	**44.20 ± 2.75**
uniform	**20.83 ± 9.06**	**21.67 ± 3.06**	**65.62 ± 5.97**	46.07 ± 4.14	**68.75 ± 1.08**	**28.31 ± 2.08**	**50.70 ± 1.93**	43.14 ± 3.64
Batch SGD	**22.50 ± 5.08**	**22.50 ± 2.65**	65.00 ± 9.74	47.16 ± 1.01	68.55 ± 0.84	**27.76 ± 0.34**	50.56 ± 1.21	**43.43 ± 2.69**
greedy batch-size ablation								
greedy, $B = 32$	**20.83 ± 6.67**	**25.00 ± 5.30**	**68.12 ± 10.46**	**48.91 ± 1.38**	**69.80 ± 1.30**	**28.49 ± 1.39**	**51.48 ± 2.05**	**44.66 ± 2.39**
greedy, $B = 64$	**20.00 ± 4.33**	**22.50 ± 2.65**	65.00 ± 3.25	**47.29 ± 4.79**	**68.25 ± 3.22**	27.39 ± 1.55	49.78 ± 4.82	**42.89 ± 1.49**
greedy, $B = 128$	**20.00 ± 4.33**	18.33 ± 3.06	**66.88 ± 6.79**	47.07 ± 3.11	68.00 ± 2.48	**27.57 ± 2.57**	**49.93 ± 2.43**	42.54 ± 1.61

F.8 PASS@4

Table 13: Pass@4 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	22.76 ± 1.98	**22.28 ± 1.52**	**74.47 ± 1.51**	**53.51 ± 3.54**	**71.08 ± 1.41**	**31.69 ± 2.13**	**55.25 ± 2.59**	**47.29 ± 1.36**
normalized_greedy	**23.23 ± 2.70**	**22.67 ± 4.29**	**75.83 ± 2.47**	**54.17 ± 0.85**	**71.61 ± 1.00**	**31.62 ± 0.91**	**56.10 ± 0.74**	**47.89 ± 1.67**
uniform	22.15 ± 6.21	19.89 ± 5.88	71.46 ± 5.05	49.34 ± 5.88	70.99 ± 2.23	31.19 ± 4.09	53.30 ± 4.34	45.48 ± 4.54
Batch SGD	**24.02 ± 3.33**	20.84 ± 2.91	72.11 ± 1.47	50.20 ± 1.39	71.07 ± 0.68	30.52 ± 0.47	53.89 ± 1.30	46.09 ± 1.25
greedy batch-size ablation								
greedy, $B = 32$	**23.02 ± 3.36**	**23.77 ± 2.13**	**74.92 ± 2.88**	**53.58 ± 1.07**	**71.94 ± 1.09**	**32.12 ± 1.12**	**55.90 ± 1.82**	**47.89 ± 1.53**
greedy, $B = 64$	**22.76 ± 1.98**	**22.28 ± 1.52**	74.47 ± 1.51	53.51 ± 3.54	**71.08 ± 1.41**	**31.69 ± 2.13**	55.25 ± 2.59	**47.29 ± 1.36**
greedy, $B = 128$	22.23 ± 3.06	19.10 ± 1.97	**75.67 ± 1.81**	**53.53 ± 2.74**	70.73 ± 1.40	30.99 ± 1.35	**55.40 ± 2.09**	46.81 ± 0.92

F.9 PASS@8

Table 14: Pass@8 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	28.47 ± 3.60	**26.27 ± 3.18**	**81.46 ± 2.49**	**62.30 ± 3.59**	73.80 ± 1.19	35.87 ± 3.92	**61.05 ± 2.75**	52.74 ± 1.70
normalized_greedy	**29.02 ± 3.66**	**27.38 ± 4.55**	**82.60 ± 3.34**	**63.30 ± 1.03**	74.53 ± 1.38	**36.01 ± 1.64**	**61.99 ± 0.85**	**53.55 ± 2.18**
uniform	28.68 ± 6.29	25.46 ± 5.63	80.83 ± 2.07	60.16 ± 4.11	**75.27 ± 2.03**	**37.29 ± 4.18**	60.70 ± 2.90	52.62 ± 3.26
Batch SGD	**31.16 ± 4.39**	25.91 ± 3.47	80.54 ± 2.22	60.87 ± 0.99	**75.14 ± 0.76**	35.99 ± 0.76	60.84 ± 1.24	**52.92 ± 1.13**
greedy batch-size ablation								
greedy, $B = 32$	**28.61 ± 3.80**	**28.79 ± 2.62**	**82.38 ± 2.23**	**62.73 ± 1.19**	**74.87 ± 1.35**	**36.99 ± 1.76**	**62.04 ± 2.14**	**53.77 ± 1.75**
greedy, $B = 64$	**28.47 ± 3.60**	**26.27 ± 3.18**	81.46 ± 2.49	**62.30 ± 3.59**	**73.80 ± 1.19**	**35.87 ± 3.92**	61.05 ± 2.75	**52.74 ± 1.70**
greedy, $B = 128$	26.94 ± 1.36	23.08 ± 3.19	**82.48 ± 1.96**	61.90 ± 3.15	73.41 ± 1.51	34.50 ± 1.77	**61.16 ± 2.12**	51.92 ± 1.11

F.10 PASS@16

Table 15: Pass@16 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	34.82 ± 5.30	30.26 ± 6.25	**87.63 ± 4.62**	**69.78 ± 3.57**	76.06 ± 1.46	40.12 ± 5.90	65.95 ± 2.85	57.80 ± 2.33
normalized_greedy	36.18 ± 3.96	**33.03 ± 5.67**	**87.88 ± 3.40**	**70.90 ± 1.15**	76.91 ± 1.83	40.71 ± 2.75	**66.72 ± 0.83**	**58.90 ± 2.52**
uniform	**36.57 ± 5.52**	30.34 ± 5.72	86.88 ± 2.82	69.13 ± 2.55	**78.69 ± 2.46**	**43.56 ± 4.42**	**66.68 ± 1.88**	58.84 ± 2.37
Batch SGD	**38.81 ± 7.43**	**30.58 ± 4.35**	86.33 ± 3.00	69.70 ± 0.97	**78.57 ± 0.97**	**41.59 ± 0.99**	66.56 ± 1.44	**58.88 ± 0.97**
greedy batch-size ablation								
greedy, $B = 32$	**34.22 ± 4.20**	**33.80 ± 3.64**	**87.64 ± 2.62**	**70.37 ± 1.09**	**77.38 ± 1.70**	**42.14 ± 2.43**	**66.93 ± 2.18**	**58.93 ± 1.79**
greedy, $B = 64$	**34.82 ± 5.30**	**30.26 ± 6.25**	**87.63 ± 4.62**	**69.78 ± 3.57**	**76.06 ± 1.46**	**40.12 ± 5.90**	**65.95 ± 2.85**	**57.80 ± 2.33**
greedy, $B = 128$	31.48 ± 0.87	26.82 ± 4.38	87.55 ± 2.20	68.96 ± 3.49	75.61 ± 1.57	37.93 ± 2.52	65.77 ± 1.66	56.30 ± 1.24

F.11 ALL-CORRECT PROMPT RATE

Table 16: All-correct prompt rate (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad
Selection methods at $B = 64$							
greedy	**0.00 ± 0.00**	**0.00 ± 0.00**	**7.50 ± 4.59**	**1.78 ± 1.19**	**23.60 ± 6.38**	**3.22 ± 1.30**	**7.86 ± 4.33**
normalized_greedy	**0.00 ± 0.00**	**0.00 ± 0.00**	**3.75 ± 5.14**	**0.52 ± 0.29**	**12.15 ± 3.72**	**1.84 ± 1.58**	**3.26 ± 0.96**
uniform	**0.00 ± 0.00**	**0.00 ± 0.00**	1.25 ± 2.30	0.09 ± 0.16	4.30 ± 7.97	0.28 ± 0.88	1.26 ± 2.35
Batch SGD	**0.00 ± 0.00**	**0.00 ± 0.00**	0.00 ± 0.00	0.02 ± 0.04	1.85 ± 0.95	0.09 ± 0.29	0.33 ± 0.23
greedy batch-size ablation							
greedy, $B = 32$	**0.00 ± 0.00**	**0.00 ± 0.00**	3.75 ± 3.98	0.52 ± 0.13	13.95 ± 2.86	1.56 ± 0.74	3.97 ± 1.63
greedy, $B = 64$	**0.00 ± 0.00**	**0.00 ± 0.00**	**7.50 ± 4.59**	**1.78 ± 1.19**	**23.60 ± 6.38**	**3.22 ± 1.30**	**7.86 ± 4.33**
greedy, $B = 128$	**0.00 ± 0.00**	**0.00 ± 0.00**	**8.12 ± 6.79**	**1.87 ± 0.66**	**25.00 ± 1.13**	**4.60 ± 1.82**	**8.12 ± 2.10**

F.12 ANY-CORRECT PROMPT RATE

Table 17: Any-correct prompt rate (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad
Selection methods at $B = 64$							
greedy	41.67 ± 6.85	35.00 ± 10.16	**92.50 ± 5.63**	75.83 ± 3.65	78.20 ± 2.11	44.21 ± 7.22	70.14 ± 2.35
normalized_greedy	**45.83 ± 6.67**	**40.83 ± 10.04**	**92.50 ± 3.25**	**77.00 ± 1.53**	79.15 ± 2.21	45.86 ± 3.36	70.66 ± 0.52
uniform	**46.67 ± 4.33**	35.00 ± 6.85	**91.88 ± 5.97**	76.27 ± 1.53	**81.45 ± 2.75**	**49.72 ± 3.55**	**71.40 ± 1.18**
Batch SGD	**45.83 ± 13.95**	**35.83 ± 5.08**	90.62 ± 5.01	**76.67 ± 1.44**	**81.65 ± 1.26**	**47.61 ± 1.47**	**71.22 ± 1.67**
greedy batch-size ablation							
greedy, $B = 32$	**39.17 ± 5.08**	**40.00 ± 7.50**	**91.25 ± 2.30**	**76.68 ± 0.75**	**79.75 ± 1.97**	**47.43 ± 3.13**	**70.55 ± 2.06**
greedy, $B = 64$	**41.67 ± 6.85**	**35.00 ± 10.16**	**92.50 ± 5.63**	**75.83 ± 3.65**	**78.20 ± 2.11**	**44.21 ± 7.22**	**70.14 ± 2.35**
greedy, $B = 128$	35.00 ± 3.06	30.83 ± 7.96	**91.25 ± 2.30**	74.87 ± 3.68	77.50 ± 1.44	41.27 ± 3.46	69.40 ± 1.45

F.13 RAW VERIFIER REWARD

Table 18: Raw verifier reward (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro	micro
Selection methods at $B = 64$									
greedy	**-0.76 ± 0.03**	**-0.77 ± 0.02**	**0.06 ± 0.07**	**-0.35 ± 0.06**	**0.22 ± 0.04**	**-0.55 ± 0.00**	**-0.20 ± 0.04**	**-0.34 ± 0.03**	**-0.26 ± 0.05**
normalized_greedy	**-0.75 ± 0.02**	**-0.75 ± 0.06**	**0.03 ± 0.04**	**-0.37 ± 0.01**	**0.20 ± 0.02**	**-0.58 ± 0.01**	**-0.21 ± 0.02**	**-0.35 ± 0.01**	**-0.28 ± 0.01**
uniform	-0.79 ± 0.10	-0.83 ± 0.07	-0.16 ± 0.20	-0.50 ± 0.14	0.07 ± 0.18	-0.64 ± 0.10	-0.34 ± 0.17	-0.46 ± 0.14	-0.40 ± 0.15
Batch SGD	-0.78 ± 0.04	-0.81 ± 0.03	-0.13 ± 0.04	-0.49 ± 0.04	0.08 ± 0.05	-0.64 ± 0.02	-0.32 ± 0.04	-0.44 ± 0.03	-0.39 ± 0.04
greedy batch-size ablation									
greedy, $B = 32$	**-0.76 ± 0.05**	**-0.76 ± 0.02**	0.01 ± 0.06	-0.38 ± 0.01	0.21 ± 0.02	-0.57 ± 0.01	-0.22 ± 0.01	-0.35 ± 0.01	-0.28 ± 0.01
greedy, $B = 64$	**-0.76 ± 0.03**	**-0.77 ± 0.02**	**0.06 ± 0.07**	**-0.35 ± 0.06**	**0.22 ± 0.04**	**-0.55 ± 0.00**	**-0.20 ± 0.04**	**-0.34 ± 0.03**	**-0.26 ± 0.05**
greedy, $B = 128$	**-0.77 ± 0.07**	-0.80 ± 0.04	**0.08 ± 0.05**	**-0.34 ± 0.04**	**0.22 ± 0.02**	**-0.55 ± 0.02**	**-0.20 ± 0.02**	**-0.34 ± 0.02**	**-0.26 ± 0.03**

F.14 OVERLONG-PENALIZED RATE

Table 19: Overlong-penalized rate (↓) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad
Selection methods at $B = 64$							
greedy	**4.82 ± 1.84**	**1.30 ± 0.50**	**0.94 ± 0.37**	**1.91 ± 0.92**	**0.50 ± 0.27**	**0.25 ± 0.32**	**1.74 ± 0.76**
normalized_greedy	9.01 ± 3.14	**3.52 ± 1.36**	**2.13 ± 0.94**	**3.74 ± 1.33**	**0.93 ± 0.26**	**0.41 ± 0.19**	**3.11 ± 0.62**
uniform	10.36 ± 4.60	4.71 ± 2.25	3.09 ± 1.22	4.57 ± 1.02	1.18 ± 0.27	0.77 ± 0.59	3.74 ± 0.85
Batch SGD	**8.39 ± 0.74**	3.83 ± 1.27	2.71 ± 0.47	3.99 ± 0.28	1.09 ± 0.04	0.68 ± 0.18	3.31 ± 0.31
greedy batch-size ablation							
greedy, $B = 32$	7.99 ± 2.79	2.84 ± 0.76	1.86 ± 0.28	3.26 ± 0.64	0.81 ± 0.33	**0.25 ± 0.25**	2.79 ± 0.12
greedy, $B = 64$	**4.82 ± 1.84**	**1.30 ± 0.50**	**0.94 ± 0.37**	**1.91 ± 0.92**	**0.50 ± 0.27**	**0.25 ± 0.32**	**1.74 ± 0.76**
greedy, $B = 128$	**4.04 ± 4.26**	**0.76 ± 1.15**	**0.88 ± 0.99**	**1.69 ± 1.58**	**0.42 ± 0.36**	0.32 ± 0.29	**1.52 ± 1.48**

F.15 TRUNCATION RATE

Table 20: Truncation rate (↓) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad
Selection methods at $B = 64$							
greedy	**3.07 ± 2.45**	**0.91 ± 0.56**	**0.64 ± 0.41**	**1.11 ± 0.78**	**0.27 ± 0.20**	**0.16 ± 0.20**	**0.98 ± 0.48**
normalized_greedy	6.90 ± 4.34	**2.58 ± 1.14**	**1.66 ± 0.83**	**2.68 ± 1.27**	**0.70 ± 0.25**	**0.31 ± 0.17**	**2.21 ± 0.73**
uniform	8.31 ± 3.79	3.44 ± 1.64	2.36 ± 1.18	3.48 ± 0.80	0.89 ± 0.21	0.52 ± 0.38	2.71 ± 0.68
Batch SGD	**6.38 ± 0.37**	2.97 ± 1.25	2.15 ± 0.30	2.96 ± 0.30	0.78 ± 0.08	0.43 ± 0.09	2.35 ± 0.29
greedy batch-size ablation							
greedy, $B = 32$	6.54 ± 3.42	2.01 ± 0.77	1.25 ± 0.43	2.39 ± 0.68	0.60 ± 0.20	**0.20 ± 0.18**	2.04 ± 0.34
greedy, $B = 64$	**3.07 ± 2.45**	**0.91 ± 0.56**	**0.64 ± 0.41**	**1.11 ± 0.78**	**0.27 ± 0.20**	**0.16 ± 0.20**	**0.98 ± 0.48**
greedy, $B = 128$	**1.90 ± 2.69**	**0.34 ± 0.49**	**0.37 ± 0.56**	**0.79 ± 0.86**	**0.23 ± 0.26**	0.20 ± 0.14	**0.74 ± 0.82**

F.16 MEAN OVERLONG PENALTY

Table 21: Mean overlong penalty (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	**-0.04 ± 0.02**	**-0.01 ± 0.01**	**-0.01 ± 0.00**	**-0.02 ± 0.01**	**-0.00 ± 0.00**	**-0.00 ± 0.00**	**-0.01 ± 0.01**	**-0.01 ± 0.01**
normalized_greedy	-0.08 ± 0.04	**-0.03 ± 0.01**	**-0.02 ± 0.01**	**-0.03 ± 0.01**	**-0.01 ± 0.00**	**-0.00 ± 0.00**	**-0.03 ± 0.01**	**-0.03 ± 0.01**
uniform	-0.09 ± 0.04	-0.04 ± 0.02	-0.03 ± 0.01	-0.04 ± 0.01	-0.01 ± 0.00	-0.01 ± 0.00	-0.03 ± 0.01	-0.04 ± 0.01
Batch SGD	**-0.07 ± 0.01**	-0.03 ± 0.01	-0.02 ± 0.00	-0.03 ± 0.00	-0.01 ± 0.00	-0.01 ± 0.00	-0.03 ± 0.00	-0.03 ± 0.00
greedy batch-size ablation								
greedy, $B = 32$	-0.07 ± 0.03	-0.02 ± 0.00	-0.01 ± 0.00	-0.03 ± 0.01	-0.01 ± 0.00	**-0.00 ± 0.00**	-0.02 ± 0.00	-0.02 ± 0.01
greedy, $B = 64$	**-0.04 ± 0.02**	**-0.01 ± 0.01**	**-0.01 ± 0.00**	**-0.02 ± 0.01**	**-0.00 ± 0.00**	**-0.00 ± 0.00**	**-0.01 ± 0.01**	**-0.01 ± 0.01**
greedy, $B = 128$	**-0.03 ± 0.04**	**-0.01 ± 0.01**	**-0.01 ± 0.01**	**-0.01 ± 0.01**	**-0.00 ± 0.00**	-0.00 ± 0.00	**-0.01 ± 0.01**	**-0.01 ± 0.01**

F.17 UNIQUE PREDICTIONS@32

Table 22: Unique predictions@32 (↑) across evaluation benchmarks.

Method	aime24	aime25	amc23	id-2048	math500	minerva	olympiad	macro
Selection methods at $B = 64$								
greedy	17.16 ± 3.04	17.55 ± 3.45	7.81 ± 1.70	11.97 ± 2.13	4.69 ± 0.97	11.49 ± 2.03	9.28 ± 1.72	11.42 ± 2.08
normalized_greedy	18.27 ± 0.98	18.47 ± 0.86	9.04 ± 0.95	12.97 ± 0.92	5.61 ± 0.47	13.51 ± 1.31	10.38 ± 0.77	12.61 ± 0.86
uniform	**20.71 ± 3.56**	**21.73 ± 2.95**	**12.16 ± 3.42**	**15.98 ± 3.04**	**8.70 ± 3.66**	**17.32 ± 4.62**	**13.50 ± 3.69**	**15.73 ± 3.51**
Batch SGD	**20.34 ± 0.70**	**21.31 ± 1.31**	**11.51 ± 1.12**	**15.77 ± 0.64**	**8.28 ± 0.93**	**17.15 ± 0.99**	**13.13 ± 0.69**	**15.36 ± 0.88**
greedy batch-size ablation								
greedy, $B = 32$	**18.41 ± 1.18**	**19.26 ± 0.25**	**9.24 ± 0.93**	**13.32 ± 0.29**	**5.74 ± 0.11**	**13.32 ± 0.32**	**10.58 ± 0.42**	**12.84 ± 0.32**
greedy, $B = 64$	**17.16 ± 3.04**	**17.55 ± 3.45**	**7.81 ± 1.70**	**11.97 ± 2.13**	**4.69 ± 0.97**	**11.49 ± 2.03**	**9.28 ± 1.72**	**11.42 ± 2.08**
greedy, $B = 128$	16.69 ± 1.48	17.10 ± 0.59	7.46 ± 0.86	11.36 ± 1.05	4.46 ± 0.52	10.90 ± 2.40	9.05 ± 0.99	11.00 ± 1.09

F.18 CANDIDATE SETS AND COUNTERFACTUAL UPDATES

Candidate Sets with Misspecification. Our main formulation in Sec. D starts from a clean candidate set and its leave-one-out mean-gradient estimates. We now extend this abstraction to an arbitrary learning algorithm $\mathcal{A}$ and optimizer $\mathcal{O}$. For prompt group i, let $\theta^+_{t,i}$ denote the parameters obtained by applying one counterfactual $(\mathcal{A}, \mathcal{O})$ update from the shared pre-update state θ_t. We define its engineering direction as $u_{t,i} = (\theta^+_{t,i} - \theta_t)/\eta_t$, producing the candidate set $\mathcal{U}_t = \{u_{t,i}\}_{i=1}^B$. Separately, let $g_{t,i}$ be the clean population-objective gradient estimate for the same prompt group, and let $\bar{g}_{t,-i} = (B-1)^{-1} \sum_{j \neq i} g_{t,j}$. Greedy selection maximizes $\langle u_{t,i}, \bar{g}_{t,-i} \rangle$. The difference $\delta_{t,i} = u_{t,i} - g_{t,i}$ captures misspecification induced by the training algorithm, optimizer, clipping, regularization, or other transformations between the clean gradient and the resulting update.

We construct each $u_{t,i}$ by executing the actual actor-update path counterfactually. Before probing, we snapshot the policy, optimizer, scheduler, gradient scaler, buffers, and RNG state. Each probe starts from this common snapshot, applies the configured algorithm and optimizer to one prompt group, records the resulting parameter displacement, and projects it into the shared sketch space. We then restore the complete snapshot before evaluating the next probe. The clean-gradient sketches are computed separately and are used only to form the leave-one-out estimates. Thus, every candidate reflects the configured training procedure, while every score is evaluated against an independent estimate of the target objective.

The surrounding training step remains unchanged. Rollout generation, reward computation, reference-model evaluation, and all other non-policy operations follow the original `verl` procedure. Only the actor update enters the candidate framework: we construct the counterfactual updates, select a winner, restore the common pre-update state, and commit the selected update through the same production optimizer path. The committed step therefore differs from the original procedure only through its prompt-level selection rule.

G AN ANALYTICAL TOY EXAMPLE

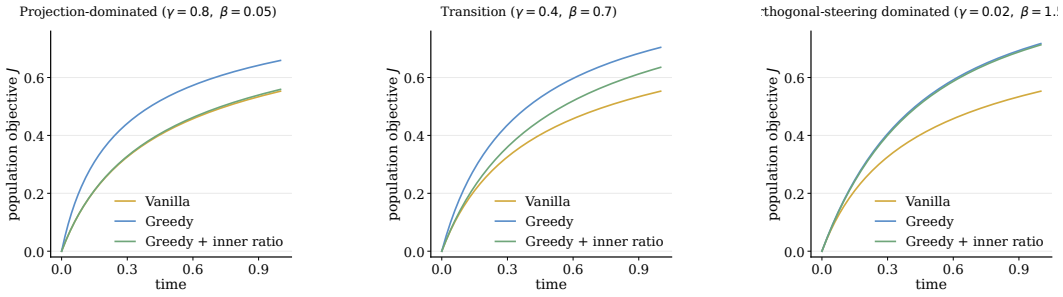


Figure 4: A continuous analytic transition from projection-dominated improvement to nontrivial orthogonal steering. Each panel compares vanilla, raw greedy, and inner-ratio-normalized greedy under the same clock.

Construction. We construct the example by separating the population-gradient direction from a second direction that can change its future magnitude without changing the objective instantaneously. Let $\theta = (u, v) \in \Theta_0 \subset \mathbb{R}^2$, where Θ_0 is a compact region that excludes the origin and lies within a fixed smooth branch of the polar angle. Define

$$r = \sqrt{u^2 + v^2}, \qquad \phi = \operatorname{atan2}(v, u), \qquad U(\theta) = \log r + \phi, \qquad H(\theta) = -\log r + \phi. \tag{32}$$

Direct calculation gives

$$\langle \nabla U(\theta), \nabla H(\theta) \rangle = 0, \qquad \|\nabla U(\theta)\|_2^2 = \|\nabla H(\theta)\|_2^2 = 2e^{-U(\theta)+H(\theta)}. \tag{33}$$

We use U to define the population objective and H to define an orthogonal potential:

$$J(\theta) = 1 - e^{-U(\theta)}, \qquad P(\theta) = 1 - e^{-H(\theta)}. \tag{34}$$

Hence

$$\langle \nabla J(\theta), \nabla P(\theta) \rangle = 0. \tag{35}$$

Consider two equally likely prompts whose exact objectives are

$$J_+(\theta) = (1+\gamma)J(\theta) + \eta P(\theta), \qquad J_-(\theta) = (1-\gamma)J(\theta) - \eta P(\theta), \tag{36}$$

where $0 < \gamma < 1$ and $\eta > 0$. Their population mean is exactly

$$\frac{1}{2}\left(J_+(\theta) + J_-(\theta)\right) = J(\theta), \tag{37}$$

while

$$\rho_+(\theta) = 1 + \gamma, \qquad \rho_-(\theta) = 1 - \gamma. \tag{38}$$

Thus $x^*(\theta) = +$ everywhere, and

$$\nabla J_+ = (1+\gamma)\nabla J + \eta \nabla P. \tag{39}$$

Raw greedy therefore follows

$$\mathrm{d}\theta_t^{(g)} = \left((1+\gamma)\nabla J(\theta_t^{(g)}) + \eta \nabla P(\theta_t^{(g)})\right) \mathrm{d}t, \tag{40}$$

whereas exact inner-ratio normalization gives

$$\mathrm{d}\theta_t^{(i)} = \left(\nabla J(\theta_t^{(i)}) + \frac{\eta}{1+\gamma} \nabla P(\theta_t^{(i)})\right) \mathrm{d}t. \tag{41}$$

The two parameters have separate roles: γ controls the parallel advantage removed by normalization, whereas $\eta/(1+\gamma)$ controls the orthogonal motion that remains.

We next verify the assumptions on any compact reachable region Θ_0 contained in the chosen smooth branch. First, all displayed objectives and vector fields are smooth on Θ_0; their gradients are bounded and Lipschitz, and the resulting flows are uniquely defined. Thus Assump. 1 holds. Second,

$$W(\theta) = \|\nabla J(\theta)\|_2^2 = 2e^{-3U(\theta)+H(\theta)} = 2(1-J(\theta))^3 e^{H(\theta)}. \tag{42}$$

Moreover,

$$\frac{\langle \nabla W(\theta), \nabla J(\theta)\rangle}{W^2(\theta)} = -3e^{U(\theta)}. \tag{43}$$

States at the same objective level have the same U, so the last quantity is identical at any pair of equal-level states. Assump. 2 therefore holds exactly with $L = \epsilon_r = 0$.

For Assump. 3, the candidate residuals after removing their parallel components are

$$\nabla J_+ - \rho_+ \nabla J = \eta \nabla P, \qquad \nabla J_- - \rho_- \nabla J = -\eta \nabla P. \tag{44}$$

Since

$$\frac{\langle \nabla W, \nabla J_+ - \rho_+ \nabla J - \nabla J_- + \rho_- \nabla J\rangle}{W^2} = 2\eta e^{2U-H} \tag{45}$$

and $\rho_+ - \rho_- = 2\gamma$, the assumption holds with $\epsilon_c = 0$ and any

$$0 < \lambda \le \frac{\eta}{\gamma} \inf_{\theta\in\Theta_0} e^{2U(\theta)-H(\theta)}. \tag{46}$$

The infimum is strictly positive because Θ_0 is compact. This calculation also shows directly why the winner's orthogonal component is favorable: along normalized greedy,

$$\frac{\mathrm{d}H}{\mathrm{d}t} = \frac{2\eta}{1+\gamma}(1-J) > 0, \tag{47}$$

while vanilla satisfies $\mathrm{d}H/\mathrm{d}t = 0$. Since $W = 2(1-J)^3 e^H$, normalized greedy reaches states with larger W at the same J.

Finally, the optional PL envelope is also nonvacuous. Taking $J^* = 1$ gives

$$\frac{W(\theta)}{2(J^* - J(\theta))} = e^{-2U(\theta)+H(\theta)}. \tag{48}$$

Its infimum on Θ_0 is strictly positive, so Assump. 5 holds for every

$$0 < \nu \le \inf_{\theta\in\Theta_0} e^{-2U(\theta)+H(\theta)}. \tag{49}$$

Consequently, every $0 < \gamma < 1$ and $\eta > 0$ produces an exact instance satisfying all the assumptions on the prescribed compact region. The boundary $\eta = 0$ is deliberately excluded: there the strict projection–steering coefficient vanishes and normalized greedy coincides with vanilla.

A Controlled Transition from Trivial to Nontrivial Dominance. The family provides a continuous transition between two qualitatively different sources of improvement. Increasing γ strengthens the winner's parallel advantage, causing raw greedy to advance mainly through an effective scalar acceleration. Increasing $\eta/(1+\gamma)$ instead strengthens the orthogonal motion retained after normalization, causing normalized greedy to enter regions with larger future population-gradient power. These effects are controlled independently without changing the objective, prompt distribution, parameter space, or winner rule.

Fig. 4 evaluates the same construction from $J(\theta_0) = H(\theta_0) = 0$ at

$$(\gamma, \eta) \in \{(0.8, 0.09),\ (0.4, 0.98),\ (0.02, 1.53)\}. \tag{50}$$

In the first setting, γ is large and $\eta/(1+\gamma)$ is small, so raw greedy substantially outperforms normalized greedy while normalized greedy remains close to vanilla. This is the projection-dominated, less informative regime. In the second setting, both mechanisms are visible and normalized greedy lies between vanilla and raw greedy. In the third setting, γ is nearly zero while $\eta/(1+\gamma)$ is large, so normalization removes little of the gain: normalized greedy nearly matches raw greedy and both dominate vanilla. This regime qualitatively resembles the empirical observation that much of the raw-greedy objective gap remains after normalization.

H PROOFS: EXISTENCE AND REGULARITY OF THE POLICY GRADIENT

Existence. Fix any $x \in \mathcal{X}$. If the score function $\nabla \log \pi_\theta(y|x)$ exists and the relevant expectation is finite, the log-derivative identity gives

$$\nabla J_x(\theta) = \mathbb{E}_{y\sim\pi_\theta(\cdot|x)} \left[R(x,y)\nabla \log \pi_\theta(y|x)\right]. \tag{51}$$

Boundedness. Let $y_1, y_2 \sim \pi_\theta(\cdot|x)$ be independent. Under the bounded-reward and bounded-score assumptions,

$$\begin{aligned}\|\nabla J_x(\theta)\|_2^2 &= \mathbb{E}_{y_1,y_2\sim\pi_\theta(\cdot|x)} \left[R(x,y_1)R(x,y_2)\left\langle \nabla \log \pi_\theta(y_1|x), \nabla \log \pi_\theta(y_2|x)\right\rangle\right] &(52)\\ &\le R_{max}^2 G_1^2. &(53)\end{aligned}$$

Therefore,

$$\|\nabla J_x(\theta)\|_2 \le R_{max} G_1. \tag{54}$$

Smoothness. We first recall a standard relation between policy coverage and total variation. For probability mass functions $\pi_1, \pi_2 \in \Delta(\Omega)$, suppose

$$\max_{\omega\in\Omega} \max\left\{\frac{\pi_2(\omega)}{\pi_1(\omega)}, \frac{\pi_1(\omega)}{\pi_2(\omega)}\right\} \le C.$$

Then

$$\frac{1}{2}\sum_{\omega\in\Omega} |\pi_1(\omega) - \pi_2(\omega)| \le \frac{C-1}{C+1} \le \frac{1}{2}\log C. \tag{55}$$

Fix a prompt x and two parameters θ_1, θ_2. Adding and subtracting the same expectation gives

$$\begin{aligned}\nabla J_x(\theta_1) - \nabla J_x(\theta_2) = &\underbrace{\mathbb{E}_{y\sim\pi_{\theta_1}(\cdot|x)} \left[R(x,y)\left(\nabla \log \pi_{\theta_1}(y|x) - \nabla \log \pi_{\theta_2}(y|x)\right)\right]}_{I_1}\\ &+ \underbrace{\left(\mathbb{E}_{y\sim\pi_{\theta_1}(\cdot|x)} - \mathbb{E}_{y\sim\pi_{\theta_2}(\cdot|x)}\right)\left[R(x,y)\nabla \log \pi_{\theta_2}(y|x)\right]}_{I_2}. &(56)\end{aligned}$$

The score-function Lipschitz assumption directly yields

$$\|I_1\|_2 \le R_{max} G_2 \|\theta_1 - \theta_2\|_2. \tag{57}$$

For the second term, bounded rewards and score functions give

$$\|I_2\|_2 \le R_{max} G_1 \sum_{y\in\mathcal{V}^H} |\pi_{\theta_1}(y|x) - \pi_{\theta_2}(y|x)|. \tag{58}$$

Moreover, the bounded-score assumption implies

$$-G_1 \|\theta_1 - \theta_2\|_2 \le \log \pi_{\theta_1}(y|x) - \log \pi_{\theta_2}(y|x) \le G_1 \|\theta_1 - \theta_2\|_2. \tag{59}$$

Applying Eq. equation 55 with $C = \exp(G_1\|\theta_1 - \theta_2\|_2)$ therefore gives

$$\begin{aligned}\|I_2\|_2 &\le 2R_{max} G_1 \frac{\exp(G_1\|\theta_1 - \theta_2\|_2) - 1}{\exp(G_1\|\theta_1 - \theta_2\|_2) + 1}\\ &\le R_{max} G_1^2 \|\theta_1 - \theta_2\|_2. &(60)\end{aligned}$$

Combining the two terms establishes

$$\|\nabla J_x(\theta_1) - \nabla J_x(\theta_2)\|_2 \le R_{max}\left(G_1^2 + G_2\right)\|\theta_1 - \theta_2\|_2. \tag{61}$$

I PROOFS: BASE PROPERTIES FOR THE RAW GREEDY ORACLE

Initial Dominance from Base Regularity. Combining Assump. 1 with the policy-gradient regularity established in App. H, we now derive the dominance result using only the regularity of the policy gradients. For convenience, let us denote $D_{\mathrm{pg}} = G_1 R_{max}$ and $L_{\mathrm{pg}} = R_{max}[G_1^2 + G_2]$ as the boundedness and Lipschitz parameters, respectively, for the policy gradients. Unrolling the calculation gives

$$\begin{aligned}\Delta(t) - \Delta(0) &= \int_0^t \Gamma(s)\,\mathrm{d}s \\ &= \int_0^t \left[\left\langle \nabla J(\theta_s^{(g)}), \nabla_{x_s^*}(\theta_s^{(g)})\right\rangle - \left\langle \nabla J(\theta_s^{(e)}), \nabla J(\theta_s^{(e)})\right\rangle\right]\mathrm{d}s.\end{aligned} \tag{62}$$

The regularity result yields the following bound on the difference between the two gradient inner products

$$\begin{aligned}&\left\langle \nabla J(\theta_s^{(g)}), \nabla J_{x_s^*}(\theta_s^{(g)})\right\rangle - \left\langle \nabla J(\theta_0), \nabla J_{x_0^*}(\theta_0)\right\rangle \\ &\geq \left\langle \nabla J(\theta_s^{(g)}), \nabla J_{x_0^*}(\theta_s^{(g)})\right\rangle - \left\langle \nabla J(\theta_0), \nabla J_{x_0^*}(\theta_0)\right\rangle \\ &= \left\langle \nabla J(\theta_s^{(g)}), \nabla J_{x_0^*}(\theta_s^{(g)})\right\rangle - \left\langle \nabla J(\theta_s^{(g)}), \nabla J_{x_0^*}(\theta_0)\right\rangle \\ &\quad + \left\langle \nabla J(\theta_s^{(g)}), \nabla J_{x_0^*}(\theta_0)\right\rangle - \left\langle \nabla J(\theta_0), \nabla J_{x_0^*}(\theta_0)\right\rangle \\ &\geq -\left\|\nabla J(\theta_s^{(g)})\right\|_2 \left\|\nabla J_{x_0^*}(\theta_s^{(g)}) - \nabla J_{x_0^*}(\theta_0)\right\|_2 - \left\|\nabla J(\theta_s^{(g)}) - \nabla J(\theta_0)\right\|_2 \left\|J_{x_0^*}(\theta_0)\right\|_2 \\ &\geq -2D_{\mathrm{pg}} \cdot L_{\mathrm{pg}} \left\|\theta_s^{(g)} - \theta_0\right\|_2 \\ &\geq -2D_{\mathrm{pg}} L_{\mathrm{pg}} \left\|\int_0^s \nabla J_{x_{s'}^*}(\theta_{s'}^{(g)})\,\mathrm{d}s'\right\|_2 \\ &\geq -2D_{\mathrm{pg}}^2 L_{\mathrm{pg}} s,\end{aligned} \tag{63}$$

and

$$\begin{aligned}&\left|\left\langle \nabla J(\theta_s^{(e)}), \nabla J(\theta_s^{(e)})\right\rangle - \langle \nabla J(\theta_0), \nabla J(\theta_0)\rangle\right| \\ &= \left|\left\langle \nabla J(\theta_s^{(e)}) + \nabla J(\theta_0), \nabla J(\theta_s^{(e)}) - \nabla J(\theta_0)\right\rangle\right| \\ &\leq 2D_{\mathrm{pg}} \cdot L_{\mathrm{pg}} \left\|\theta_s^{(e)} - \theta_0\right\|_2 \\ &= 2D_{\mathrm{pg}} L_{\mathrm{pg}} \left\|\int_0^s \nabla J(\theta_{s'}^{(e)})\,\mathrm{d}s'\right\|_2 \\ &\leq 2D_{\mathrm{pg}}^2 L_{\mathrm{pg}} s,\end{aligned} \tag{64}$$

Substituting the result in Eq. 63 and 64 back into Eq. 62,

$$\begin{aligned}\Delta(t) &\geq \int_0^t \left[-4D_{\mathrm{pg}}^2 L_{\mathrm{pg}} \cdot s + \Gamma(0)\right]\mathrm{d}s \\ &= -2D_{\mathrm{pg}}^2 L_{\mathrm{pg}} t^2 + \Gamma(0) t.\end{aligned} \tag{65}$$

Consequently, for any

$$t \leq \frac{\Gamma(0)}{2D_{\mathrm{pg}}^2 L_{\mathrm{pg}}} = \frac{\Gamma(0)}{2G_1^2 R_{max}^2 \cdot R_{max}\left[G_1^2 + G_2\right]} = \frac{\Gamma(0)}{2G_1^2\left[G_1^2 + G_2\right] R_{max}^3}, \tag{66}$$

the corresponding dominance function $\Delta(t) \geq 0$.

Convergence under Base PL Envelope. By Assump. 5 and the definition of $\theta_t^{(g)}$, it follows that

$$\frac{\mathrm{d}}{\mathrm{d}t}J(\theta_t^{(g)}) \geq W(\theta_t^{(g)}) \geq 2\nu(J^* - J(\theta_t^{(g)})). \tag{67}$$

As a direct consequence of the Grönwall's inequality,

$$0 \leq J^* - J(\theta_t^{(g)}) \leq [J^* - J(\theta_0)]\exp(-2\nu t). \tag{68}$$

converges exponentially with t.

J Proofs: Global Dominance

J.1 Normalized Greedy Flow

The Level-Set Partition. Let

$$r_0 = J(\theta_0). \tag{69}$$

By Assump. 1, both flows are absolutely continuous, and the chain rule holds almost everywhere. Moreover, along either flow,

$$\frac{\mathrm{d}}{\mathrm{d}t}J(\theta_t) = W(\theta_t). \tag{70}$$

Assump. 5 implies that $W(\theta) > 0$ whenever $J(\theta) < J^*$. Hence neither flow can become stationary at a suboptimal objective value, and both objectives are strictly increasing until they reach the common optimum J^*. We first carry out the level-set comparison on an arbitrary objective level $r > r_0$ reached by both flows; the admissible range of such levels over the prescribed horizon $[0, T)$ is determined at the end of the proof.

For the normalized greedy flow, the chain rule gives

$$J(\theta_t^{(i)}) - r_0 = \int_0^t \left\langle \nabla J(\theta_s^{(i)}), \frac{1}{\rho(\theta_s^{(i)})}\nabla J_{x^*}(\theta_s^{(i)}) \right\rangle \mathrm{d}s = \int_0^t W(\theta_s^{(i)})\,\mathrm{d}s, \tag{71}$$

and similarly,

$$J(\theta_t^{(e)}) - r_0 = \int_0^t \left\langle \nabla J(\theta_s^{(e)}), \nabla J(\theta_s^{(e)}) \right\rangle \mathrm{d}s = \int_0^t W(\theta_s^{(e)})\,\mathrm{d}s. \tag{72}$$

These Lebesgue integrals are well-defined because their integrands are measurable and locally bounded along the corresponding absolutely continuous trajectories by Assump. 1. Consequently, for every jointly reachable level $r \in (r_0, J^*)$, the first-arrival times

$$\mathtt{t}^{(i)}(r) = \inf\left\{t \in [0,T) : J(\theta_t^{(i)}) \geq r\right\}, \qquad \mathtt{t}^{(e)}(r) = \inf\left\{t \in [0,T) : J(\theta_t^{(e)}) \geq r\right\} \tag{73}$$

are well-defined and unique, with

$$J(\theta^{(i)}_{\mathtt{t}^{(i)}(r)}) = J(\theta^{(e)}_{\mathtt{t}^{(e)}(r)}) = r. \tag{74}$$

We accordingly define

$$\vartheta_r^{(i)} = \theta^{(i)}_{\mathtt{t}^{(i)}(r)}, \qquad \vartheta_r^{(e)} = \theta^{(e)}_{\mathtt{t}^{(e)}(r)}. \tag{75}$$

Thus, $\vartheta_r^{(i)}$ and $\vartheta_r^{(e)}$ are the unique states reached by the two flows at the same population-objective level r; they need not be the same point in the parameter space.

The Chain Rule. On every compact interval of jointly reachable levels contained in (r_0, J^*), Assump. 5 bounds W away from zero along both level-set parameterizations. The inverse change of variables is therefore well-defined and gives, almost everywhere,

$$\frac{\mathrm{d}r}{\mathrm{d}\mathtt{t}^{(i)}(r)} = W(\vartheta_r^{(i)}), \qquad \frac{\mathrm{d}r}{\mathrm{d}\mathtt{t}^{(e)}(r)} = W(\vartheta_r^{(e)}), \tag{76}$$

or equivalently,

$$\frac{\mathrm{d}\mathtt{t}^{(i)}(r)}{\mathrm{d}r} = \frac{1}{W(\vartheta_r^{(i)})}, \qquad \frac{\mathrm{d}\mathtt{t}^{(e)}(r)}{\mathrm{d}r} = \frac{1}{W(\vartheta_r^{(e)})}. \tag{77}$$

Therefore,

$$\mathsf{t}^{(e)}(r) - \mathsf{t}^{(i)}(r) = \int_{r_0}^{r} \left(\frac{1}{W(\vartheta_{r'}^{(e)})} - \frac{1}{W(\vartheta_{r'}^{(i)})} \right) \mathrm{d}r'. \tag{78}$$

It remains to prove that this difference is nonnegative on the entire interval of jointly reachable levels.

Power Functions on the Same Level. Taking the logarithm converts the evolution of the power function into its relative rate of change. Along either level-parameterized flow, the chain rule gives

$$\begin{aligned} \frac{\mathrm{d}\log W}{\mathrm{d}r} &= \frac{\mathrm{d}\log W}{\mathrm{d}W} \left\langle \nabla W, \frac{\mathrm{d}\theta}{\mathrm{d}t} \right\rangle \frac{\mathrm{d}t}{\mathrm{d}r} \\ &= \frac{\left\langle \nabla W, \frac{\mathrm{d}\theta}{\mathrm{d}t} \right\rangle}{W^2}, \end{aligned} \tag{79}$$

where we have reused $\frac{\mathrm{d}r}{\mathrm{d}t} = W$. Applying this identity to the normalized greedy and vanilla flows gives

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}r} \log \frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})} &= \frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J_{x^*}(\vartheta_r^{(i)}) \right\rangle}{\rho(\vartheta_r^{(i)}) W^2(\vartheta_r^{(i)})} - \frac{\left\langle \nabla W(\vartheta_r^{(e)}), \nabla J(\vartheta_r^{(e)}) \right\rangle}{W^2(\vartheta_r^{(e)})} \\ &= \frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J_{x^*}(\vartheta_r^{(i)}) \right\rangle}{\rho(\vartheta_r^{(i)}) W^2(\vartheta_r^{(i)})} - \frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J(\vartheta_r^{(i)}) \right\rangle}{W^2(\vartheta_r^{(i)})} \\ &\quad + \frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J(\vartheta_r^{(i)}) \right\rangle}{W^2(\vartheta_r^{(i)})} - \frac{\left\langle \nabla W(\vartheta_r^{(e)}), \nabla J(\vartheta_r^{(e)}) \right\rangle}{W^2(\vartheta_r^{(e)})} \\ &= \frac{\left\langle \nabla W(\vartheta_r^{(i)}), \frac{1}{\rho(\vartheta_r^{(i)})} \nabla J_{x^*}(\vartheta_r^{(i)}) - \nabla J(\vartheta_r^{(i)}) \right\rangle}{W^2(\vartheta_r^{(i)})} \\ &\quad + \left[\frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J(\vartheta_r^{(i)}) \right\rangle}{W^2(\vartheta_r^{(i)})} - \frac{\left\langle \nabla W(\vartheta_r^{(e)}), \nabla J(\vartheta_r^{(e)}) \right\rangle}{W^2(\vartheta_r^{(e)})} \right]. \end{aligned} \tag{80}$$

For the first term, fix $x^* = x^*(\vartheta_r^{(i)})$ and consider an arbitrary $z \in \mathcal{X}$. Since x^* maximizes the projection coefficient, $\rho(\vartheta_r^{(i)}) \geq \rho_z(\vartheta_r^{(i)})$, and Assump. 3 gives

$$\begin{aligned} &\frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J_{x^*}(\vartheta_r^{(i)}) - \rho(\vartheta_r^{(i)}) \nabla J(\vartheta_r^{(i)}) - \nabla J_z(\vartheta_r^{(i)}) + \rho_z(\vartheta_r^{(i)}) \nabla J(\vartheta_r^{(i)}) \right\rangle}{W^2(\vartheta_r^{(i)})} \\ &\geq \lambda \left(\rho(\vartheta_r^{(i)}) - \rho_z(\vartheta_r^{(i)}) \right) - \epsilon_c. \end{aligned} \tag{81}$$

Taking the expectation over $z \sim d_0$ and using

$$\mathbb{E}_z \left[\rho_z(\vartheta_r^{(i)}) \right] = 1, \qquad \mathbb{E}_z \left[\nabla J_z(\vartheta_r^{(i)}) - \rho_z(\vartheta_r^{(i)}) \nabla J(\vartheta_r^{(i)}) \right] = 0, \tag{82}$$

we obtain

$$\frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J_{x^*}(\vartheta_r^{(i)}) - \rho(\vartheta_r^{(i)}) \nabla J(\vartheta_r^{(i)}) \right\rangle}{W^2(\vartheta_r^{(i)})} \geq \lambda \left(\rho(\vartheta_r^{(i)}) - 1 \right) - \epsilon_c. \tag{83}$$

By Assump. 4, $\rho(\vartheta_r^{(i)}) \geq \rho_{\min} > 1$. Since

$$\lambda(\rho_{\min} - 1) > \epsilon_c, \tag{84}$$

and the function

$$\rho \longmapsto \frac{\lambda(\rho-1)-\epsilon_c}{\rho} \tag{85}$$

is increasing on $(0,\infty)$, it follows that

$$\begin{aligned}\frac{\left\langle \nabla W(\vartheta_r^{(i)}), \frac{1}{\rho(\vartheta_r^{(i)})}\nabla J_{x^*}(\vartheta_r^{(i)}) - \nabla J(\vartheta_r^{(i)})\right\rangle}{W^2(\vartheta_r^{(i)})} &\geq \frac{\lambda\left(\rho(\vartheta_r^{(i)})-1\right)-\epsilon_c}{\rho(\vartheta_r^{(i)})} \\ &\geq \frac{\lambda(\rho_{\min}-1)-\epsilon_c}{\rho_{\min}}.\end{aligned} \tag{86}$$

For the second term, Assump. 2 directly gives

$$\frac{\left\langle \nabla W(\vartheta_r^{(i)}), \nabla J(\vartheta_r^{(i)})\right\rangle}{W^2(\vartheta_r^{(i)})} - \frac{\left\langle \nabla W(\vartheta_r^{(e)}), \nabla J(\vartheta_r^{(e)})\right\rangle}{W^2(\vartheta_r^{(e)})} \geq -L\left|\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})}\right|. \tag{87}$$

Combining Eq. 86 with Eq. 87 yields

$$\frac{\mathrm{d}}{\mathrm{d}r}\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})} \geq \frac{\lambda(\rho_{\min}-1)-\epsilon_c}{\rho_{\min}} - L\left|\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})}\right|. \tag{88}$$

At the initial level,

$$\vartheta_{r_0}^{(i)} = \vartheta_{r_0}^{(e)} = \theta_0, \qquad \log\frac{W(\vartheta_{r_0}^{(i)})}{W(\vartheta_{r_0}^{(e)})} = 0. \tag{89}$$

The scalar comparison equation associated with Eq. 88 starts from zero with the strictly positive drift in Eq. 84. The comparison principle therefore implies

$$\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})} \geq 0. \tag{90}$$

We may consequently remove the absolute value in Eq. 88 and obtain

$$\frac{\mathrm{d}}{\mathrm{d}r}\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})} \geq \frac{\lambda(\rho_{\min}-1)-\epsilon_c}{\rho_{\min}} - L\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})}. \tag{91}$$

For $L>0$, Grönwall's inequality gives

$$\begin{aligned}\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})} &\geq \frac{\lambda(\rho_{\min}-1)-\epsilon_c}{L\rho_{\min}}\left(1-e^{-L(r-r_0)}\right) \\ &> 0.\end{aligned} \tag{92}$$

When $L=0$, the same comparison gives

$$\begin{aligned}\log\frac{W(\vartheta_r^{(i)})}{W(\vartheta_r^{(e)})} &\geq \frac{\lambda(\rho_{\min}-1)-\epsilon_c}{\rho_{\min}}(r-r_0) \\ &> 0.\end{aligned} \tag{93}$$

Therefore,

$$W(\vartheta_r^{(i)}) > W(\vartheta_r^{(e)}). \tag{94}$$

The Global Dominance. Substituting Eq. 94 into Eq. 78 gives, for every jointly reachable $r > r_0$,

$$\begin{aligned}\mathsf{t}^{(e)}(r) - \mathsf{t}^{(i)}(r) &= \int_{r_0}^{r} \left(\frac{1}{W(\vartheta_{r'}^{(e)})} - \frac{1}{W(\vartheta_{r'}^{(i)})} \right) \mathrm{d}r' \\ &\geq 0.\end{aligned} \tag{95}$$

Thus normalized greedy reaches every jointly reachable objective level no later than vanilla. Moreover, the interval of jointly reachable levels cannot terminate at any level $r < J^*$ that vanilla subsequently reaches. Indeed, Assump. 5 gives $W(\vartheta_r^{(i)}) > 0$, while Assump. 1 locally extends the normalized-greedy trajectory through strictly larger objective levels. Since $\mathsf{t}^{(i)}(r) \leq \mathsf{t}^{(e)}(r) < T$, this extension occurs within the prescribed horizon, contradicting maximality of the jointly reachable interval. Consequently, every level reached by vanilla before J^* is also reached by normalized greedy.

We now distinguish the possible behaviors within $[0, T)$. If neither flow reaches J^* in this interval, then, for every $t < T$, the level $r = J(\theta_t^{(e)}) < J^*$ is jointly reachable. Since $\mathsf{t}^{(e)}(r) = t$ and $\mathsf{t}^{(i)}(r) \leq t$, monotonicity gives

$$J(\theta_t^{(i)}) \geq J(\theta^{(i)}_{\mathsf{t}^{(i)}(r)}) = r = J(\theta_t^{(e)}). \tag{96}$$

If either flow reaches J^* at some time in $[0, T)$, the same argument applies before the first such time. If normalized greedy reaches J^* first, maximality of J^* makes the desired inequality immediate thereafter. If vanilla reaches J^* first, letting $r \to J^*$ in $\mathsf{t}^{(i)}(r) \leq \mathsf{t}^{(e)}(r)$ and using continuity shows that normalized greedy reaches J^* no later; both flows then remain at the common optimum. Therefore,

$$J(\theta_t^{(i)}) \geq J(\theta_t^{(e)}), \qquad t \in [0, T). \tag{97}$$

This completes the proof.

J.2 Greedy Flow with Time Reparameterization

Define the normalized-greedy vector field by

$$g(\theta) = \frac{1}{\rho(\theta)} \nabla J_{x^*}(\theta). \tag{98}$$

The raw and normalized greedy flows therefore satisfy

$$\frac{\mathrm{d}\theta_t^{(g)}}{\mathrm{d}t} = \nabla J_{x^*}(\theta_t^{(g)}) = \rho(\theta_t^{(g)}) g(\theta_t^{(g)}), \qquad \frac{\mathrm{d}\theta_t^{(i)}}{\mathrm{d}t} = g(\theta_t^{(i)}). \tag{99}$$

Let $s(t)$ solve

$$\frac{\mathrm{d}s(t)}{\mathrm{d}t} = \rho(\theta_{s(t)}^{(i)}), \qquad s(0) = 0. \tag{100}$$

By the chain rule,

$$\frac{\mathrm{d}}{\mathrm{d}t} \theta_{s(t)}^{(i)} = g(\theta_{s(t)}^{(i)}) \frac{\mathrm{d}s(t)}{\mathrm{d}t} = \rho(\theta_{s(t)}^{(i)}) g(\theta_{s(t)}^{(i)}) = \nabla J_{x^*}(\theta_{s(t)}^{(i)}). \tag{101}$$

Hence $\theta_{s(t)}^{(i)}$ satisfies the raw-greedy differential equation with the same initialization. Uniqueness in Assump. 1 gives

$$\theta_t^{(g)} = \theta_{s(t)}^{(i)}. \tag{102}$$

Moreover, Assump. 4 gives

$$\frac{\mathrm{d}s(t)}{\mathrm{d}t} = \rho(\theta_{s(t)}^{(i)}) \geq \rho_{\min} > 1, \tag{103}$$

and therefore $s(t) \geq t$. Since $J(\theta_t^{(i)})$ is nondecreasing,

$$J(\theta_t^{(g)}) = J(\theta_{s(t)}^{(i)}) \geq J(\theta_t^{(i)}). \tag{104}$$

Combining this inequality with the normalized-greedy dominance proved above yields

$$J(\theta_t^{(g)}) \geq J(\theta_t^{(i)}) \geq J(\theta_t^{(e)}), \qquad t \in [0, T), \tag{105}$$

for every time at which the corresponding reparameterized trajectories are defined. If either greedy flow reaches J^* earlier, the conclusion extends thereafter by maximality of J^*.