

Beyond SFT-Optimality: Cold-Start Data Selection for Reinforcement Learning via Gradient Subspace Coverage

Haisheng Lv^{1,2,*}, Zikun Su^{1,3,*}, Jingbang Liu^{1,4}, Zilu Huang¹, Xinzhang Liu¹,
Wenmin Deng¹, Yu Zhao^{1,†}, Chao Wang¹, Shuangyong Song^{1,†}

¹Xingchen AGI Lab, China Telecom Artificial Intelligence Technology (Beijing) Co., Ltd

²Peking University ³Beijing University of Posts and Telecommunications

⁴École Polytechnique Fédérale de Lausanne

Abstract

Cold-start supervised fine-tuning (SFT) is a critical initialization step before reinforcement learning from verifiable rewards (RLVR). Yet existing cold-start data selection methods largely inherit criteria from supervised instruction tuning—sample quality, difficulty, perplexity, or diversity—implicitly assuming that a stronger SFT warm start is also a better starting point for RL. We argue that this assumption does not always hold in the SFT-to-RL pipeline: cold-start SFT not only reduces supervised loss but also shapes the initial update geometry that downstream reward optimization inherits.

Building on this observation, we propose RL-Proxy-Coverage. Our method estimates a low-rank, reward-relevant gradient subspace from a small set of representative RL anchor prompts and verifier rewards, and selects, as the cold-start subset, those candidate SFT examples whose gradients provide non-redundant coverage of this subspace. In controlled mathematical-reasoning experiments, RL-Proxy-Coverage achieves the highest six-benchmark average at the designated early-to-mid RL checkpoint at both model scales, with stronger statistical evidence at 7B. Our findings suggest a new view of cold-start data: it should be regarded not merely as supervised examples for improving SFT, but as an initialization resource that shapes the starting point of downstream reward optimization.

1 Introduction

Modern math reasoning models are commonly built through a two-stage pipeline: cold-start supervised fine-tuning (SFT) on curated reasoning traces is followed by reinforcement learning from verifiable rewards (RLVR), where verifier feedback further improves sampled trajectories (Guo et al.,

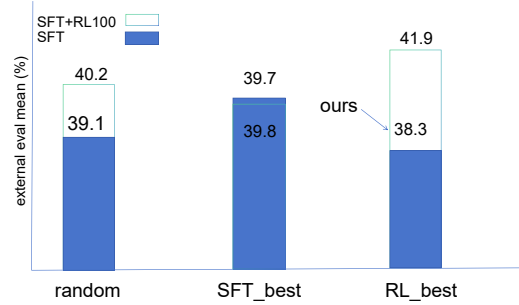


Figure 1: **SFT-best is not necessarily RL-best.** A representative SFT-to-RL rank inversion on Qwen2.5-Math-7B. Shown are six-benchmark means for three cold-start subsets before RL (filled) and after 100 RL steps (outline). The subset with the strongest SFT-only performance (SFT_best) is overtaken at RL100 by ours (RL_best), despite the latter starting from the lowest SFT-only score.

2025; Tan et al., 2025). While the choice of cold-start data has been recognized as important (Zhou et al., 2023; Muennighoff et al., 2025), existing selection methods largely inherit criteria from supervised instruction tuning—sample quality, complexity, difficulty, model confidence, or long chain-of-thought traces (Li et al., 2024a; Liu et al., 2024; Yang et al., 2026). These criteria are well suited to producing a stronger SFT checkpoint, but implicitly assume that a better SFT warm start is also a better initialization for subsequent RL.

This assumption is not reliable in our controlled experiments. Figure 1 shows a representative inversion on Qwen2.5-Math-7B: a subset that is stronger after SFT becomes weaker after the same downstream RL procedure. More systematically, across eight controlled selectors at each scale, the rank correlation between SFT-only and post-RL performance is weak on 1.5B (Spearman $\rho = 0.271$) and negative on 7B ($\rho = -0.339$); our subset moves from SFT rank 7/8 and 5.5/8, respectively, to the highest post-RL rank at both scales. Thus,

*Equal contribution.

[†]Corresponding authors. Correspondence: zhaoy11@chinatelecom.cn, songshy@chinatelecom.cn.

supervised quality alone is not a stable proxy for downstream RL utility in the settings we study. A local first-order view offers a design intuition: a sufficiently small SFT step aligned with a fixed RL gradient decreases the local RL loss to first order. This observation is only a local motivation, not a guarantee for multi-step RL under policy, sampling, or optimizer drift. It nevertheless suggests a different question: *how should we choose a cold-start SFT subset specifically to improve the sample efficiency of subsequent RL?*

We propose RL-Proxy-Coverage, an offline data selection method designed for RL initialization. Using a compact set of representative RL anchor prompts together with verifier rewards, RL-Proxy-Coverage estimates a low-rank, reward-relevant gradient subspace and selects candidate SFT examples whose gradients provide non-redundant coverage of this subspace. Rather than collapsing reward-aware gradients into a single mean direction, the subspace retains multiple recurring update modes that the cold start can cover. Our approach connects to gradient- and subspace-based data selection (Xia et al., 2024; Zhao et al., 2025b) but differs in what the target geometry represents: prior methods derive it from supervised loss or validation gradients, whereas we derive it from proxy RL gradients that characterize the local geometry of downstream reward optimization.

In controlled experiments on Qwen2.5-Math-1.5B and -7B, RL-Proxy-Coverage attains the highest six-benchmark mean at the early-to-mid RL checkpoint at both scales while keeping the cold-start budget, SFT recipe, RL recipe, and evaluation protocol fixed. Complete mechanism ablations at both scales isolate the roles of reward-relevant target geometry, multi-direction coverage, subspace alignment, and the gradient representation itself: replacing the RL-proxy target with an SFT-gradient subspace or collapsing it to a single mean direction consistently reduces post-RL performance. A lower-cost 1k-anchor variant further remains favorable when method-specific offline selection cost and downstream RL GPU occupancy are accounted for jointly. In summary, our contributions are:

- We reformulate cold-start data selection as an initialization problem for downstream RL sample efficiency, and provide systematic evidence that SFT-only rankings need not predict post-RL rankings.
- We instantiate this view as RL-Proxy-Coverage,

which extracts a low-rank reward-relevant subspace from RL-proxy anchors and selects SFT examples that provide non-redundant coverage of it.

- We provide controlled experiments at two model scales, paired bootstrap analyses, complete mechanism ablations, and compute-aware comparisons that isolate the roles of RL-proxy target geometry and multi-direction coverage.

2 Related Work

Mathematical reasoning and RL post-training.

Reasoning ability in large language models is typically improved through chain-of-thought prompting, self-consistency, bootstrapped reasoning, and verifier supervision (Wei et al., 2022; Wang et al., 2023; Zelikman et al., 2022; Cobbe et al., 2021), and is systematically evaluated on benchmarks such as MATH, OlympiadBench, and MATH-Perturb (Hendrycks et al., 2021; He et al., 2024a; Huang et al., 2025). Recent open math models and RLVR systems further show that post-training with verifiers or reward functions on tasks with verifiable answers can substantially improve performance (Shao et al., 2024; Yang et al., 2024; Guo et al., 2025; Tan et al., 2025). General-purpose model families such as TeleChat likewise combine supervised fine-tuning with reinforcement-learning-based post-training (He et al., 2024b; Wang et al., 2024, 2025b). These works mainly focus on training pipelines, RL recipes, or final performance; our work is complementary, studying how to choose cold-start SFT data *before* entering RL so as to improve early RL sample efficiency.

Reasoning data curation and SFT data selection.

The importance of carefully curated data for alignment and reasoning has been observed in many works: LIMA shows that a small set of high-quality demonstrations can induce strong instruction-following behavior (Zhou et al., 2023), while the Tele-FLM series similarly reports a “less is more” effect in SFT data construction (Li et al., 2024b). In mathematical reasoning, s1 and LIMO emphasize curated reasoning data and test-time scaling (Muennighoff et al., 2025; Ye et al., 2025), and Select2Reason selects high-value long-CoT instruction data by difficulty and long reasoning trajectories (Yang et al., 2026). Structured reasoning supervision has also been explored through computation-logic graphs for complex mathematics and step-wise linguistic reasoning in other

generation tasks (Zhao et al., 2025a; Shi et al., 2026). Large-scale math instruction data and mixed reasoning data also remain valuable for training math models (Toshniwal et al., 2024, 2025; Yu et al., 2024; Yue et al., 2024; Luo et al., 2025). In more general instruction tuning, AlpaGasus, Superfiltering, and DEITA filter data by model scoring, weak-to-strong filtering, and complexity/quality/diversity (Chen et al., 2024b; Li et al., 2024a; Liu et al., 2024). These methods mainly optimize SFT or instruction-following performance; we instead focus on RL efficiency after SFT—a subset that yields better SFT-only performance is not necessarily the best cold-start subset for subsequent reward optimization.

Data selection via gradients, subspaces, and diversity. Gradient representations offer a sample characterization closer to training dynamics. Classical subset selection methods such as CRAIG, GradMatch, and GLISTER use training or validation gradients to choose subsets that represent the full dataset or improve validation performance (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a,b). In LLM instruction tuning, LESS computes per-example gradients in LoRA parameter space and builds a low-dimensional gradient datastore via random projection (Xia et al., 2024); G2IS and GIST further represent downstream tasks as multiple directions or a low-rank subspace (Zhao et al., 2025b; Min et al., 2026); DPP and log-determinant objectives are commonly used to model set diversity and non-redundant coverage (Kulesza and Taskar, 2012; Wang et al., 2025a). Our work also uses LoRA-parameter gradients, random projection, and log-determinant coverage, but with a different target: prior gradient- and subspace-based methods construct the target geometry from supervised loss or validation gradients, whereas we use RL-proxy gradients to characterize the local update structure of downstream reward optimization for SFT $\rightarrow$ RL cold-start selection.

3 Problem Setup and Motivation

We study cold-start data selection in the SFT-to-RL pipeline. Given an initial math language model f_{θ_0} , a candidate supervised reasoning pool $\mathcal{D} = \{x_i\}_{i=1}^n$, and a cold-start budget k , the selector chooses a subset $S \subseteq \mathcal{D}$ with $|S| = k$. This subset drives cold-start SFT to produce a warm-start model $\theta_S = \text{SFT}(\theta_0, S)$, which then enters a

fixed downstream RL process:

$$\theta_{S,t} = \text{RL}_t(\theta_S; \mathcal{R}, \mathcal{D}_{\text{RL}}), \quad (1)$$

where $\mathcal{R}$ is the reward/verifier, $\mathcal{D}_{\text{RL}}$ the RL prompt pool, and t the RL training step.

Because cold-start SFT serves as an initialization for reward optimization rather than as the final objective, we replace SFT-centric criteria with an RL-oriented objective. For a fixed target checkpoint t in the early-to-mid RL stage, we define

$$S_{\text{RL}}^*(t) = \arg \max_{S \subseteq \mathcal{D}, |S|=k} \mathcal{E}(\theta_{S,t}), \quad t \in \mathcal{T}_{\text{early}}, \quad (2)$$

where $\mathcal{E}$ is external evaluation performance and $\mathcal{T}_{\text{early}}$ denotes the early-to-mid RL checkpoints. For each selector comparison, t is fixed before evaluation rather than optimized over or averaged across checkpoints.

First-order motivation. A local first-order view motivates why SFT-only quality need not coincide with the objective in Eq. (2). At a local parameter point θ (instantiated in our method by the shared warm-up probe; §4.1), let

$$g_S = \frac{1}{|S|} \sum_{x \in S} \nabla_{\theta} \ell_{\text{SFT}}(\theta; x) \quad (3)$$

denote the subset SFT-loss gradient. Treating cold-start SFT locally as a single small step $\theta^+ = \theta - \eta g_S$, and writing $L_{\text{RL}}(\theta) = -J_{\text{RL}}(\theta)$ and $g_{\text{RL}} = \nabla_{\theta} L_{\text{RL}}(\theta)$, a first-order Taylor expansion gives

$$L_{\text{RL}}(\theta - \eta g_S) \approx L_{\text{RL}}(\theta) - \eta \langle g_{\text{RL}}, g_S \rangle. \quad (4)$$

Thus, for a sufficiently small step and a fixed local RL gradient, $\langle g_{\text{RL}}, g_S \rangle > 0$ implies an immediate decrease in the local RL loss to first order. Two subsets with similar supervised quality can therefore induce different local effects on the downstream objective if their gradients interact differently with reward-relevant directions.

Importantly, Eq. (4) is only a local design motivation. It characterizes one small SFT step at a fixed parameter point and fixed RL gradient; it does not imply faster multi-step RL convergence, nor does it account for policy shift, changing rollout samples, adaptive optimization, or non-convex dynamics. We therefore use the first-order argument only to motivate an RL-aware geometry for cold-start selection. Whether that geometry yields better early-to-mid RL sample efficiency is an empirical question evaluated in §6 and §7.

From alignment to coverage. The local argument above motivates directional compatibility with reward optimization, but does not imply that downstream RL can be represented by a single direction. A natural baseline is to average the RL-proxy gradients into one vector and apply LESS-style similarity selection (Xia et al., 2024). Such averaging, however, necessarily collapses variation among the local update directions induced by different RL anchors. Distinct reward-bearing prompts and rollouts need not produce collinear proxy gradients, and directions away from their mean may still contain useful reward-relevant structure.

We therefore retain multiple dominant modes of the local RL-proxy geometry rather than reducing it to scalar mean alignment. Specifically, RL-Proxy-Coverage estimates a low-rank subspace from the anchor gradients and selects cold-start examples that provide non-redundant coverage within that subspace. This construction should be understood as a richer local representation, not as an assumption that the estimated subspace remains invariant throughout downstream RL. Its practical value is tested directly through the mean-alignment and SFT-subspace controls in §7.

4 Method: RL-Proxy Gradient Subspace Coverage

RL-Proxy-Coverage is an offline cold-start selector. Given a candidate SFT pool and budget k , it (i) constructs local RL-proxy gradient signals from representative anchors with verifier rewards, (ii) estimates a low-rank subspace from the resulting gradient bank, and (iii) selects examples whose SFT gradients provide non-redundant coverage of this subspace. The method changes only the selected cold-start subset; the downstream SFT and RL procedures remain unchanged. Figure 2 shows the pipeline, and Algorithm 1 gives the pseudocode.

4.1 Warm-up probe model and shared gradient space

To compare candidate-side SFT gradients and anchor-side RL-proxy gradients in a common local coordinate system, RL-Proxy-Coverage uses a single warm-up probe model θ_w . The probe is obtained by LoRA SFT on a small set of supervised reasoning examples; it serves only as the shared reference point for offline gradient extraction and is *never* used as the final warm start for downstream

Algorithm 1 RL-Proxy-Coverage: cold-start subset selection

Require: warm-up probe θ_w , random projection $P : \mathbb{R}^p \rightarrow \mathbb{R}^d$, candidates $\{x_i\}_{i=1}^n$, anchors $\{a_j\}_{j=1}^m$ with rewards, rank r , budget k

- 1: $b_i \leftarrow P \nabla_{\psi} \ell_{\text{SFT}}(\theta_w; x_i)$, $b_i \leftarrow b_i / \|b_i\|_2$ for all candidates
- 2: $h_a \leftarrow P \nabla_{\psi} L_{\text{RL}}^{\text{proxy}}(a; \theta)|_{\theta_w}$, $h_a \leftarrow h_a / \|h_a\|_2$ for anchors with non-zero proxy gradients
- 3: $H \leftarrow [h_{a_1}, \dots, h_{a_m}]^T$, $\tilde{H} \leftarrow H - \mathbf{1}_m \mu^T$ where $\mu = \frac{1}{m} H^T \mathbf{1}_m$
- 4: $V_r \leftarrow \text{top-}r \text{ right singular vectors of } \tilde{H}$
- 5: $z_i \leftarrow V_r^T b_i$ for all i
- 6: $S \leftarrow \emptyset$
- 7: **while** $|S| < k$ **do**
- 8: $i^* \leftarrow \arg \max_{i \notin S} \log \det(\varepsilon_{\det} I_r + Z_{S \cup \{i\}}^T Z_{S \cup \{i\}})$
- 9: $S \leftarrow S \cup \{i^*\}$
- 10: **end while**
- 11: **return** S

SFT/RL.

Using the same θ_w places both gradient types in the same local parameter region. To control extraction cost, we compute gradients only with respect to the LoRA parameters ψ and apply the same random projection $P : \mathbb{R}^p \rightarrow \mathbb{R}^d$ to both sides. For each candidate x_i ,

$$u_i = P \nabla_{\psi} \ell_{\text{SFT}}(\theta_w; x_i), \quad b_i = u_i / \|u_i\|_2. \quad (5)$$

Stacking all candidate embeddings yields

$$G = [b_1, \dots, b_n]^T \in \mathbb{R}^{n \times d}. \quad (6)$$

We set $d = 4096$ and sample entries of P i.i.d. from $\mathcal{N}(0, 1/d)$; all projected vectors are L_2 -normalized. This representation follows LESS-style LoRA-gradient datastores (Xia et al., 2024; Hu et al., 2022) and is supported by classical distance-preserving properties of random projections (Johnson and Lindenstrauss, 1984; Achlioptas, 2003).

4.2 RL-proxy anchors and gradients

Computing proxy RL gradients for the full RL prompt pool is expensive, so we estimate the local RL geometry from a representative anchor set. Anchors are drawn from a decontaminated DeepScaleR-style RL prompt pool (Tan et al., 2025): we embed prompts with BGE-M3 (Chen

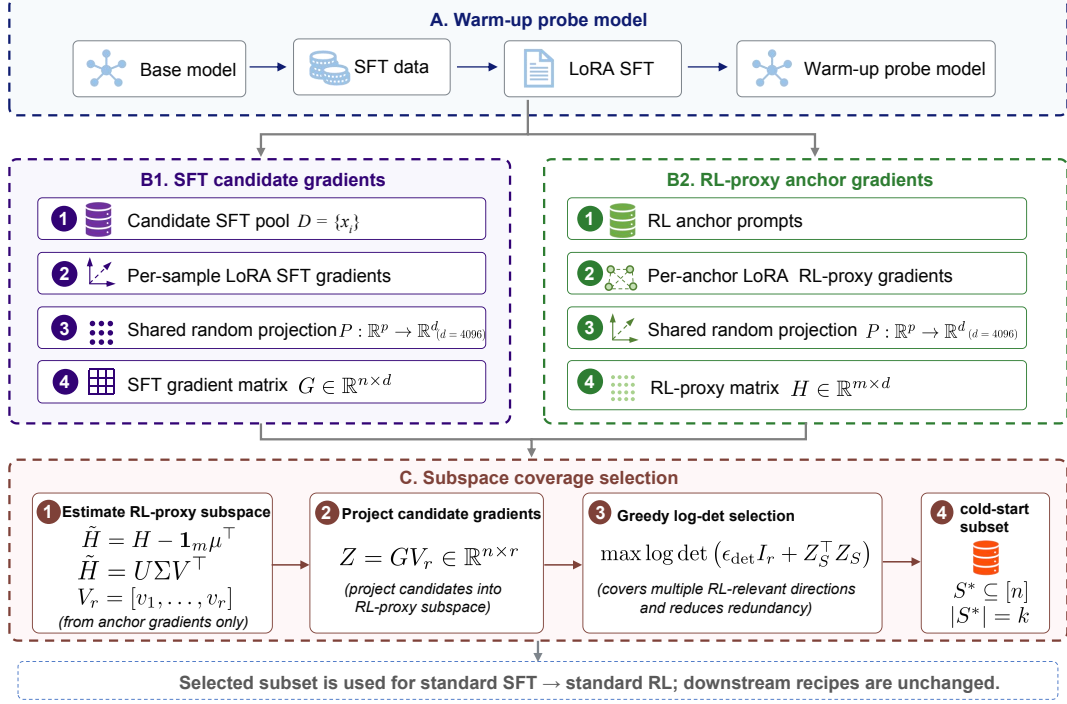


Figure 2: **Pipeline of RL-Proxy-Coverage.** From a shared warm-up probe θ_w , we (i) extract LoRA-gradient embeddings of candidate SFT examples and RL-proxy anchors with verifier rewards, (ii) build a low-rank RL-proxy subspace V_r via SVD on the anchor embeddings, and (iii) project each candidate into V_r and greedily select a budget- k subset that maximizes log-determinant coverage. The selected subset is then used for cold-start SFT before downstream RL.

et al., 2024a) and apply global k -means clustering (Lloyd, 1982). With 100 clusters and the 50 prompts closest to each centroid, the main configuration uses 5,000 anchors. The anchor count controls a substantial part of the offline cost; we therefore evaluate a lower-cost 1,000-anchor variant separately in §7.

For each anchor a , we sample K responses from θ_w and obtain rewards from a math verifier. Let μ_a and σ_a denote the mean and standard deviation of the K rewards. We use group-normalized advantages

$$\hat{A}_{a,j} = \frac{R(y_{a,j}) - \mu_a}{\sigma_a + \epsilon_{\text{adv}}}. \quad (7)$$

We then construct a PPO/GRPO-style local surrogate to extract an RL-proxy gradient, following recent math RL practice with group-relative reward normalization (Schulman et al., 2017; Shao et al., 2024). With $\ell_{a,j,t}(\theta) = \log \pi_\theta(y_{a,j,t} | a, y_{a,j,<t})$, the local likelihood ratio is

$$\rho_{a,j,t}(\theta) = \exp(\ell_{a,j,t}(\theta) - \text{sg}[\ell_{a,j,t}(\theta_w)]), \quad (8)$$

where $\text{sg}[\cdot]$ denotes stop-gradient. Old-policy log-

probabilities are fixed at θ_w , and gradients are evaluated only at $\theta = \theta_w$.

For brevity, define the clipped ratio

$$\bar{\rho}_{a,j,t}(\theta) = \text{clip}(\rho_{a,j,t}(\theta), 1 - \delta, 1 + \delta), \quad (9)$$

and the token-level surrogate

$$s_{a,j,t}(\theta) = \min \left\{ \rho_{a,j,t}(\theta) \hat{A}_{a,j}, \bar{\rho}_{a,j,t}(\theta) \hat{A}_{a,j} \right\}. \quad (10)$$

Let $\mathcal{M}_a$ denote the response-token positions participating in the actor loss, and let $L_{\text{KL}}(a; \theta)$ denote the token-level KL of π_θ against π_{θ_w} on $\mathcal{M}_a$, using the k_3 estimator (Shao et al., 2024). The resulting local surrogate is

$$L_{\text{RL}}^{\text{proxy}}(a; \theta) = -\frac{1}{|\mathcal{M}_a|} \sum_{(j,t) \in \mathcal{M}_a} s_{a,j,t}(\theta) + \beta L_{\text{KL}}(a; \theta), \quad (11)$$

with $\delta = 0.2$ and $\beta = 0.001$ (Schulman et al., 2017; Shao et al., 2024).

Importantly, this surrogate is used only as a local gradient extractor: no parameter update is

performed during proxy construction. Its LoRA gradient is projected into the shared space:

$$q_a = P \nabla_{\psi} L_{\text{RL}}^{\text{proxy}}(a; \theta) \big|_{\theta=\theta_w}, \quad h_a = q_a / \|q_a\|_2. \quad (12)$$

Anchors with zero proxy gradients are removed before subspace construction; in the 1.5B experiment, 1,604 of the 5,000 anchors yield non-zero gradients.

4.3 RL-proxy principal subspace

Stacking the non-zero RL-proxy gradient embeddings gives

$$H = [h_{a_1}, \dots, h_{a_m}]^{\top} \in \mathbb{R}^{m \times d}. \quad (13)$$

We center the anchor matrix to characterize its dominant variation directions:

$$\mu = \frac{1}{m} H^{\top} \mathbf{1}_m, \quad \tilde{H} = H - \mathbf{1}_m \mu^{\top}. \quad (14)$$

We then compute an SVD and retain the top- r right singular vectors:

$$\tilde{H} = U \Sigma V^{\top}, \quad V_r = [v_1, \dots, v_r] \in \mathbb{R}^{d \times r}. \quad (15)$$

We set $r = 64$. Thus, V_r represents the dominant variation modes in the local RL-proxy gradient bank rather than collapsing the bank to a single mean direction. As emphasized in §3, this is a local representation at θ_w ; we do not assume that the resulting subspace remains invariant throughout downstream RL.

Given V_r , we project the candidate SFT gradient matrix G onto this subspace:

$$Z = G V_r \in \mathbb{R}^{n \times r}. \quad (16)$$

We write $z_i^{\top}$ for the i -th row of Z , representing candidate x_i in the RL-proxy principal subspace.

4.4 Log-determinant coverage selection

Given the RL-proxy principal subspace, we use a log-determinant objective for fixed-budget selection. This volume-based criterion, closely related to DPP and submodular diversity objectives (Kulesza and Taskar, 2012; Krause et al., 2008), rewards subsets that provide non-redundant coverage across the retained RL-proxy directions.

For a candidate subset S , let $Z_S \in \mathbb{R}^{|S| \times r}$ denote the corresponding projected matrix, and define

$$C_S = \varepsilon_{\text{det}} I_r + Z_S^{\top} Z_S, \quad \varepsilon_{\text{det}} = 10^{-6}. \quad (17)$$

The coverage objective is

$$S^* = \arg \max_{S \subseteq [n], |S|=k} F(S), \quad F(S) = \log \det(C_S). \quad (18)$$

Since each selected example contributes the positive-semidefinite rank-one term $z_i z_i^{\top}$, the normalized objective $F(S) - F(\emptyset)$ is monotone submodular (Krause et al., 2008). Standard greedy selection therefore enjoys the usual $(1 - 1/e)$ approximation guarantee for the normalized objective: starting from $\emptyset$, we repeatedly add the candidate with the largest marginal increase in F until $|S| = k$ (Algorithm 1).

5 Experiments

5.1 Experimental Protocol

We evaluate cold-start data selection in a mathematical-reasoning RL setting. For each selector, we draw a fixed-budget subset from a shared eligible candidate pool, fine-tune the same base model on that subset, and feed the resulting checkpoint into the same verifiable-reward RL pipeline. Within each model scale, all main comparisons share the same base model, cold-start budget, SFT and RL configurations, RL prompt pool, verifier, hyperparameters, and external evaluation protocol; only the cold-start selection criterion varies.

Main experiments use Qwen2.5-Math-1.5B and -7B (Yang et al., 2024) with a cold-start budget of $k = 3000$. We evaluate all selectors at a checkpoint fixed before comparison: RL50 for 1.5B and RL100 for 7B. The 7B run uses half the rollout batch size of the 1.5B run (64 vs. 128) due to memory constraints, so RL100 at 7B and RL50 at 1.5B consume approximately the same amount of RL training data. Thus, the two scales are compared under approximately matched RL-data budgets rather than matched step counts.

Additional SFT-to-RL ranking and compute-aware analyses are reported separately in §7 and do not alter the fixed-checkpoint main comparison. Candidate-pool construction, decontamination, training, checkpointing, bootstrap, and hyperparameters are detailed in Appendices A and B.

5.2 Cold-Start Selectors

We compare five primary cold-start selectors, all of which select $k = 3000$ examples from the same eligible candidate pool:

- `sl_style`: a hard-plus-diverse selector common in reasoning-data curation, preferring harder examples while preserving semantic diversity over prompt embeddings (Muennighoff et al., 2025; Ye et al., 2025).
- `random`: uniformly random sampling, serving as the no-signal baseline.
- `ppl_low`: selects the k candidates with the lowest base-model perplexity, representing a model-confidence criterion that favors examples close to the base-model distribution.
- `rl_less_mean`: averages the non-zero RL-proxy anchor gradients into a single direction and selects SFT examples most aligned with that mean. This is a LESS-style reward-aware alignment baseline that shares the same candidate gradients and 5k RL-proxy gradient bank as RL-Proxy-Coverage.
- **RL-Proxy-Coverage (ours)**: uses the same 5k-anchor RL-proxy bank to estimate a low-rank local RL-proxy subspace and selects SFT examples that provide non-redundant coverage within that subspace.

For mechanism analysis, we additionally consider three controlled variants: `sft_subspace`, which retains log-determinant coverage but replaces the RL-proxy subspace with an SFT-gradient subspace; `align_bottom`, which uses the RL-proxy subspace but favors weakly aligned candidates as a negative control; and `adam_update`, which replaces raw gradients with one-step Adam-style update directions. Together with the five primary selectors, these form the eight-selector set used in the SFT-to-RL ranking analysis in §7. The lower-cost 1k-anchor variant is used only for the compute-aware analysis and is kept distinct from the 5k-anchor main configuration; full selector and ablation definitions are provided in Appendix C.

5.3 Evaluation Protocol

We evaluate on six external mathematical-reasoning benchmarks: AIME 2024, 2025, and 2026, MATH-500 (Hendrycks et al., 2021), MATH-P-Hard (Huang et al., 2025), and OlympiadBench (He et al., 2024a). For each AIME set, we sample $N = 64$ responses per problem at temperature $T = 0.6$ and report the mean verifier pass rate, denoted $\text{Pass}@1_{N=64}$. For the remaining benchmarks, we use deterministic decoding and report verified accuracy. The

reported six-benchmark average is the unweighted mean of these six scores.

None of the external evaluation sets participates in cold-start selection, SFT training, proxy-dev checkpoint selection, or downstream RL training.

6 Main Results

Table 1 reports the fixed-checkpoint comparison. Paired problem-level bootstrap with 10,000 replicates (Appendix A.4) shows that, on 7B, RL-Proxy-Coverage reaches 0.419 and its differences from all four primary baselines have 95% confidence intervals excluding zero. On 1.5B, RL-Proxy-Coverage achieves the highest mean (0.309); the differences from `random` and `sl_style` exclude zero, while those from `ppl_low` and `rl_less_mean` remain positive but non-significant.

The most direct reward-aware control is `rl_less_mean`, which uses the same 5k RL-proxy bank and candidate gradients but collapses them to one mean direction. It reaches 0.301/0.397 on 1.5B/7B, versus 0.309/0.419 for RL-Proxy-Coverage, supporting multi-direction coverage. `ppl_low` remains competitive on 1.5B (0.306) but reaches 0.399 on 7B, while `sl_style` obtains 0.267/0.402. Gains vary across individual benchmarks; our claim concerns aggregate early-to-mid RL performance rather than per-benchmark dominance.

7 Analysis and Ablations

Table 2 reports the complete mechanism matrix at both scales, isolating target geometry, multi-direction coverage, subspace alignment, and gradient representation; additional subset and subspace diagnostics, together with cost-accounting details, are reported in Appendices E and F.

RL-proxy geometry and coverage are complementary. `sft_subspace` keeps log-determinant coverage but replaces the RL-proxy target with an SFT-gradient subspace, reaching 0.301/0.406 on 1.5B/7B versus 0.309/0.419 for RL-Proxy-Coverage. Conversely, `rl_less_mean` keeps the RL-proxy signal but collapses it to one mean direction, reaching 0.301/0.397. Thus, neither generic gradient-space coverage nor reward-aware mean alignment alone recovers the full result.

Alignment with the RL-proxy subspace matters. `align_bottom` uses the same RL-proxy

Model	Method	Avg.	MATH-500	MATH-P-Hard	AIME26	AIME25	AIME24	Olym.
1.5B (RL50)	sl_style	0.267	0.621	0.343	0.086	0.085	0.113	0.352
	random	0.294	0.646	0.356	0.113	0.127	0.135	0.388
	ppl_low	<u>0.306</u>	<u>0.666</u>	<u>0.377</u>	0.110	0.139	<u>0.146</u>	0.400
	rl_less_mean	0.301	0.664	0.347	<u>0.119</u>	<u>0.128</u>	0.149	<u>0.399</u>
	RL-Proxy-Coverage	0.309	0.673	0.403	0.121	0.118	0.144	0.396
7B (RL100)	sl_style	0.402	0.780	0.466	0.230	0.227	<u>0.249</u>	0.462
	random	0.402	<u>0.773</u>	<u>0.492</u>	0.190	<u>0.208</u>	0.232	<u>0.520</u>
	ppl_low	0.399	<u>0.764</u>	<u>0.487</u>	0.198	<u>0.196</u>	0.230	0.519
	rl_less_mean	0.397	0.755	0.458	0.206	<u>0.208</u>	0.233	0.522
	RL-Proxy-Coverage	0.419	<u>0.773</u>	0.513	<u>0.220</u>	0.227	0.261	0.517

Table 1: External evaluation at the fixed early-to-mid RL checkpoints. Avg. is the unweighted mean of six benchmarks. Within each model block, all settings are fixed except the cold-start subset. Bold/underline denote the best/second-best values.

Model	Method	SFT	RL	Gain	vs. Ours
1.5B (RL50)	RL-Proxy-Coverage	0.255	0.309	+0.054	+0.000
	sft_subspace	0.274	0.301	+0.026	−0.008
	rl_less_mean	0.284	0.301	+0.017	−0.008
	align_bottom	0.266	0.282	+0.016	−0.027
	adam_update	0.284	0.299	+0.015	−0.010
7B (RL100)	RL-Proxy-Coverage	0.383	0.419	+0.036	+0.000
	sft_subspace	0.385	0.406	+0.021	−0.013
	rl_less_mean	0.398	0.397	−0.001	−0.022
	align_bottom	0.382	0.410	+0.028	−0.009
	adam_update	0.388	0.410	+0.022	−0.009

Table 2: Mechanism ablations at both scales. *SFT*: six-benchmark mean before RL; *RL*: RL50 (1.5B) or RL100 (7B); *Gain*: RL−SFT; *vs. Ours*: post-RL gap to RL-Proxy-Coverage.

subspace but favors weakly aligned candidates, reducing post-RL performance to 0.282/0.410 despite SFT-only scores close to RL-Proxy-Coverage. This further separates downstream RL utility from supervised performance.

Raw gradients provide a stronger selection representation. `adam_update` converts raw gradients into one-step Adam-style directions using the warm-up optimizer state. Although it improves SFT-only performance at both scales, its post-RL scores fall to 0.299/0.410. A plausible explanation is that Adam’s state-dependent coordinate-wise preconditioning alters cross-example geometry, whereas raw gradients provide an optimizer-independent local representation. This concerns the selection representation, not the downstream optimizer.

SFT-only ranking is not a reliable post-RL proxy. Across all eight controlled selectors, SFT-only and post-RL rankings have Spearman correlations of only +0.271 on 1.5B and −0.339 on 7B. RL-Proxy-Coverage moves from SFT rank 7/8 and 5.5/8, respectively, to post-RL rank 1/8 at both scales; 6/7 and 4/7 competing selectors exhibit an

Scale	Method	Steps	GPU-h	Avg.
1.5B	sl_style	50	39.01+	0.267
	random	90	31.81	0.306
	ppl_low	83	30.43	0.307
	RL-LESS-Mean-5k	50	46.57+	0.301
	Ours-1k	50	23.60	0.308
7B	sl_style	100	113.53+	0.402
	random	140	56.87	0.402
	ppl_low	133	50.18	0.392
	RL-LESS-Mean-5k	100	70.53	0.397
	Ours-1k	100	45.18	0.420

Table 3: Selector-plus-RL compute comparison. “+” denotes a conservative logged lower bound.

inversion relative to RL-Proxy-Coverage. This systematic pattern strengthens the motivation beyond the example in Figure 1.

Selector-plus-RL compute. To account for offline selection cost, we evaluate a lower-cost 1k-anchor variant (**Ours-1k**) against cheaper selectors given additional downstream RL steps. Checkpoints for this comparison were fixed before external benchmark evaluation.

Under this logged cost boundary, **Ours-1k** achieves the highest average while using the fewest

GPU-hours at both scales (Table 3). It remains favorable even when cheaper baselines receive additional RL steps, indicating that the early-RL gain is not simply purchased by extra selector compute. The comparison is GPU-hour based rather than exactly matched by generated tokens or verifier calls.

8 Conclusion

This paper reframes cold-start SFT data selection as an initialization problem for downstream RL: SFT-only performance is not a reliable proxy for the utility of a warm start after reward optimization begins. Building on this view, RL-Proxy-Coverage represents local reward-aware update geometry with a low-rank RL-proxy gradient subspace and selects cold-start examples that provide non-redundant coverage within it. On Qwen2.5-Math-7B, RL-Proxy-Coverage reaches a six-benchmark mean of 0.419 at RL100, with paired-bootstrap differences against all four primary baselines whose 95% confidence intervals exclude zero; on 1.5B it achieves the highest RL50 mean (0.309), although the margins over the strongest baselines are smaller. Across eight controlled selectors, SFT-only rankings are only weakly related to post-RL rankings, further supporting the distinction between supervised quality and downstream-RL utility.

Complete mechanism ablations at both scales show that replacing the RL-proxy target with a generic SFT-gradient subspace, collapsing the proxy bank to a single mean direction, favoring weakly aligned candidates, or using Adam-style update directions all reduce post-RL performance relative to RL-Proxy-Coverage. A lower-cost 1k-anchor variant also remains favorable under our logged selector-plus-RL GPU-hour accounting. These results support RL-Proxy-Coverage as an RL-oriented cold-start selector for improving early-to-mid RL efficiency in the controlled mathematical-reasoning setting studied here.

Limitations

Task and model scope. Our experiments are limited to verifiable mathematical reasoning with Qwen2.5-Math-1.5B and -7B. Whether the same selection principle transfers to larger model families, code RL, tool use, multi-turn agents, or open-ended preference optimization remains unknown. We also do not evaluate cross-model transfer, such as using gradients from a smaller model to select

cold-start data for a larger model.

Local theoretical scope. The first-order analysis in §3 is deliberately local. It describes the effect of one sufficiently small SFT step at a fixed parameter point and fixed RL gradient, and does not provide a convergence guarantee for multi-step RL under policy shift, changing rollout distributions, adaptive optimization, or non-convex dynamics. Likewise, the RL-proxy subspace is estimated locally at the warm-up probe; we do not assume that it remains invariant throughout downstream RL. The value of this representation is therefore established empirically rather than through a trajectory-level theoretical guarantee.

Baseline breadth. Our gradient-based comparisons are designed as controlled one-factor tests: `rl_less_mean` shares the RL-proxy gradient bank but reduces it to mean alignment, while `sft_subspace` retains multi-direction coverage but replaces the RL-proxy target with an SFT-gradient subspace. These controls directly isolate the two components central to our method, but they do not exhaust the broader family of published gradient-, coreset-, and subspace-based selection algorithms. Adapting additional methods with different objectives or validation requirements to the same SFT-to-RL pipeline would provide a broader comparison.

Compute accounting. The main RL-Proxy-Coverage configuration uses 5,000 RL anchors, and its offline cost is dominated by proxy rollouts and gradient extraction. Our compute-aware experiment shows that reducing the proxy bank to 1,000 anchors can retain favorable performance under logged selector-plus-RL GPU-hour accounting, including comparisons in which cheaper baselines receive additional RL steps. However, this analysis uses realized GPU occupancy rather than exact matching by generated tokens, verifier calls, or other hardware-independent measures, and several logged totals are conservative lower bounds. Shared costs such as the final 3k SFT stage and external evaluation are excluded because they are common across methods. More comprehensive cost accounting would further clarify the trade-off between offline selection and downstream RL compute.

Hyperparameters and anchor construction. We use fixed choices for the projection dimension, subspace rank, clustering procedure, and anchor

count. The optimal rank and number of anchors may depend on model scale, reward sparsity, RL algorithm, and the diversity of the RL prompt distribution. Although the 1k-anchor experiment suggests that the 5k main configuration is not necessary for retaining the observed benefit, adaptive selection of these quantities remains open.

Statistical and longitudinal coverage. The paired bootstrap quantifies evaluation problem-sampling uncertainty and is not a substitute for full training-seed variance. We additionally re-ran the `random` and `RL-Proxy-Coverage` 1.5B SFT-to-RL pipelines with another seed and obtained consistent conclusions (Appendix D.3), but we do not repeat every selector and ablation under multiple training seeds. Our claims are also explicitly limited to the early-to-mid RL regime. As downstream RL continues to supply substantially more reward signal, initialization effects may diminish, and we do not claim permanent or asymptotic superiority of one cold start over another.

Ethical Considerations

This work studies offline data selection for mathematical-reasoning post-training with verifiable rewards. We do not collect new human-subject data, conduct user studies, or annotate sensitive personal attributes. The supervised candidate pool, RL prompt pool, and external evaluation sets consist of mathematical reasoning problems and solutions. No external evaluation set is used for cold-start selection, SFT training, proxy-dev checkpoint selection, or downstream RL training, and the downstream RL pool is decontaminated against the proxy-dev set, the SFT subsets available at the time of its construction, and the external benchmarks (Appendix A.1).

The main ethical considerations arise from the provenance of the underlying resources and from potential downstream applications of the method. Candidate examples may carry source-specific licensing, attribution, or redistribution constraints, especially when derived from public contest or forum-style mathematical collections. Any release of selected subsets, checkpoints, or gradient-derived metadata should therefore preserve the access conditions and documentation of the original sources.

More broadly, `RL-Proxy-Coverage` is intended as a data-selection mechanism for improving the efficiency of RL post-training. Applying the same

principle to open-ended generation, user-derived data, or high-stakes domains would require additional consideration of privacy, consent, representational harms, and reward misspecification. Verifier-based optimization can also encourage exploitation of imperfections in the reward function rather than genuine improvements in reasoning quality. Consequently, benchmark gains obtained with this training pipeline should not by themselves be interpreted as evidence of deployment readiness.

Acknowledgments

This work was supported by the Innovation Studio for Beijing Outstanding Young Engineers.

Generative AI tools were used during manuscript preparation for language editing, drafting assistance, and formatting support. All technical content, claims, analyses, citations, and final wording were reviewed and approved by the authors.

References

- Dimitris Achlioptas. 2003. [Database-friendly random projections: Johnson-Lindenstrauss with binary coins](#). *Journal of Computer and System Sciences*, 66(4):671–687.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. 2024b. [AlpaGasus: Training a better alpaca with fewer data](#). In *The Twelfth International Conference on Learning Representations*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *arXiv preprint arXiv:2110.14168*.
- 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, and 175 others. 2025. [DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638.

- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024a. [OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.
- Zhongjiang He, Zihan Wang, Xinzhang Liu, Shixuan Liu, Yitong Yao, Yuyao Huang, Xuelong Li, Yongxiang Li, Zhonghao Che, Zhaoxi Zhang, Yan Wang, Xin Wang, Luwen Pu, Huinan Xu, Ruiyu Fang, Yu Zhao, Jie Zhang, Xiaomeng Huang, Zhilong Lu, and 17 others. 2024b. [TeleChat technical report](#). *arXiv preprint arXiv:2401.03804*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. 2025. [MATHperturb: Benchmarking LLMs’ math reasoning abilities against hard perturbations](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 25311–25328. PMLR.
- William B. Johnson and Joram Lindenstrauss. 1984. [Extensions of Lipschitz mappings into a Hilbert space](#). *Contemporary Mathematics*, 26:189–206.
- Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. 2021a. [GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training](#). In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5464–5474. PMLR.
- Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. 2021b. [GLISTER: Generalization based data subset selection for efficient and robust learning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118.
- Andreas Krause, Ajit Singh, and Carlos Guestrin. 2008. [Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies](#). *Journal of Machine Learning Research*, 9:235–284.
- Alex Kulesza and Ben Taskar. 2012. [Determinantal point processes for machine learning](#). *Foundations and Trends in Machine Learning*, 5(2–3):123–286.
- Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. 2024a. [Superfiltering: Weak-to-strong data filtering for fast instruction-tuning](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14255–14273, Bangkok, Thailand. Association for Computational Linguistics.
- Xiang Li, Yiqun Yao, Xin Jiang, Xuezhi Fang, Chao Wang, Xinzhang Liu, Zihan Wang, Yu Zhao, Xin Wang, Yuyao Huang, Shuangyong Song, Yongxiang Li, Zheng Zhang, Bo Zhao, Aixin Sun, Yequan Wang, Zhongjiang He, Zhongyuan Wang, Xuelong Li, and Tiejun Huang. 2024b. [52B to 1T: Lessons learned via Tele-FLM series](#). *arXiv preprint arXiv:2407.02783*.
- Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2024. [What makes good data for alignment? A comprehensive study of automatic data selection in instruction tuning](#). In *The Twelfth International Conference on Learning Representations*.
- Stuart P. Lloyd. 1982. [Least squares quantization in PCM](#). *IEEE Transactions on Information Theory*, 28(2):129–137.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. 2025. [WizardMath: Empowering mathematical reasoning for large language models via reinforced evol-instruct](#). In *The Thirteenth International Conference on Learning Representations*.
- Guanghui Min, Tianhao Huang, Ke Wan, and Chen Chen. 2026. [GIST: Targeted data selection for instruction tuning via coupled optimization geometry](#). *arXiv preprint arXiv:2602.18584*.
- Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. 2020. [Coresets for data-efficient training of machine learning models](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6950–6960. PMLR.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. [s1: Simple test-time scaling](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321, Suzhou, China. Association for Computational Linguistics.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.

- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Lingling Shi, Haoyu Jin, Ruiyu Fang, Shuangyong Song, Jinsong Su, Yongxiang Li, and Xuelong Li. 2026. [CEMT: Controllable element-oriented machine translation via structured linguistic reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 37755–37781, San Diego, California, United States. Association for Computational Linguistics.
- Sijun Tan, Michael Luo, Justin Wong, Colin Cai, Xiaoxiang Shi, William Yuan Tang, Manan Roongta, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. [DeepScaleR: Effective RL scaling of reasoning models via iterative context lengthening](#). OpenReview preprint.
- Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisanin, Alexan Ayrapetyan, and Igor Gitman. 2025. [OpenMathInstruct-2: Accelerating AI for math with massive open-source instruction data](#). In *The Thirteenth International Conference on Learning Representations*.
- Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. [OpenMathInstruct-1: A 1.8 million math instruction tuning dataset](#). *arXiv preprint arXiv:2402.10176*.
- Peiqi Wang, Yikang Shen, Zhen Guo, Matthew Stallone, Yoon Kim, Polina Golland, and Rameswar Panda. 2025a. [Diversity measurement and subset selection for instruction tuning datasets](#). In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Zihan Wang, XinZhang Liu, Shixuan Liu, Yitong Yao, Yuyao Huang, Mengxiang Li, Zhongjiang He, Yongxiang Li, Luwen Pu, Huinan Xu, Chao Wang, and Shuangyong Song. 2024. [TeleChat: An open-source bilingual large language model](#). In *Proceedings of the 10th SIGHAN Workshop on Chinese Language Processing (SIGHAN-10)*, pages 10–20, Bangkok, Thailand. Association for Computational Linguistics.
- Zihan Wang, Xinzhang Liu, Yitong Yao, Chao Wang, Yu Zhao, Zhihao Yang, Wenmin Deng, Kaipeng Jia, Jiaxin Peng, Yuyao Huang, Sishi Xiong, Zhuo Jiang, Kaidong Yu, Xiaohui Hu, Fubei Yao, Ruiyu Fang, Zhuoru Jiang, Ruiting Song, Qiyi Xie, and 19 others. 2025b. [Technical report of TeleChat2, TeleChat2.5 and T1](#). *arXiv preprint arXiv:2507.18013*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.
- Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. [LESS: Selecting influential data for targeted instruction tuning](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 54104–54132. PMLR.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. [Learning to reason under off-policy guidance](#). In *Advances in Neural Information Processing Systems*, volume 38.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. [Qwen2.5-Math Technical Report: Toward mathematical expert model via self-improvement](#). *arXiv preprint arXiv:2409.12122*.
- Cehao Yang, Xueyuan Lin, Xiaojun Wu, Chengjin Xu, Xuhui Jiang, Honghao Liu, Hui Xiong, and Jian Guo. 2026. [Select2Reason: Efficient instruction-tuning data selection for long-CoT reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 6656–6671, San Diego, California, United States. Association for Computational Linguistics.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. [LIMO: Less is more for reasoning](#). In *Second Conference on Language Modeling*.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. [MetaMath: Bootstrap your own mathematical questions for large language models](#). In *The Twelfth International Conference on Learning Representations*.
- Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhui Chen. 2024. [MAMmoTH: Building math generalist models through hybrid instruction tuning](#). In *The Twelfth International Conference on Learning Representations*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. [STaR: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488. Curran Associates, Inc.
- Deji Zhao, Donghong Han, Jia Wu, Zhongjiang He, Bo Ning, Ye Yuan, Yongxiang Li, Chao Wang, and

Shuangyong Song. 2025a. [Enhancing math reasoning ability of large language models via computation logic graphs](#). *Knowledge-Based Systems*, 325:113905.

Yang Zhao, Li Du, Xiao Ding, Yangou Ouyang, Hepeng Wang, Kai Xiong, Jinglong Gao, Zhouhao Sun, Dongliang Xu, Qing Yang, Dongchen Li, Bing Qin, and Ting Liu. 2025b. [Beyond similarity: A gradient-based graph method for instruction tuning data selection](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24391–24404, Vienna, Austria. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. [LIMA: Less is more for alignment](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 55006–55021.

A Experimental Details

This appendix provides experimental and implementation details omitted from the main text. For the fixed-checkpoint main comparisons and mechanism ablations, we follow a controlled-comparison principle: within a model scale, the base model, cold-start budget, SFT configuration, checkpoint selection, RL prompt pool, reward/verifier, RL algorithm, RL hyperparameters, and external evaluation protocol are held fixed, and only the cold-start SFT selection criterion varies. The separate compute-aware analysis intentionally reallocates downstream RL steps under a prospective cost budget and is described in §F.

A.1 Data Pools and Decontamination

We distinguish three data pools. First, the parent candidate pool is constructed from the public `Elliott/Openr1-Math-46k-8192` release associated with LUFFY (Yan et al., 2025). Starting from 45,792 raw records, we extract supervised reasoning examples, deduplicate them by normalized problem text, and retain examples whose gold answers can be reliably parsed. This produces a parent pool of 34,540 supervised reasoning examples, which serves as the common source for all cold-start selectors. The retained examples span seven recorded source categories: `olympiads` (26,164), `cn_contest` (4,006), `aops_forum` (3,405), `amc_aime` (463), `inequalities` (280), `number_theory` (109), and `olympiads_ref` (113).

Second, we use a canonical 1,000-example proxy-dev split for SFT checkpoint selection and diagnostic analysis. The split is constructed deterministically using a stable hash after excluding the SFT training examples already materialized at the time of proxy-dev construction. The final materialized SFT subsets used in the reported comparisons exclude these proxy-dev examples. We therefore refer to the remaining 33,540 parent examples as the eligible candidate pool. Each cold-start selector ultimately produces a 3,000-example training subset from this common candidate universe.

Third, downstream RL uses a decontaminated version of DeepScaleR-40K (Tan et al., 2025). Starting from 40,315 raw prompts, we apply answer-validity filtering, normalized-problem deduplication, prompt-length filtering (maximum 1024 tokens), and exact/fuzzy decontamination against the proxy-dev set, the union of SFT training

subsets already materialized at the time of RL-pool construction, and the external benchmarks (AIME 2024/2025/2026, MATH-500, MATH-P-Hard, and OlympiadBench) (Hendrycks et al., 2021; He et al., 2024a; Huang et al., 2025). For fuzzy matching, we normalize problem text and use character 5-gram Jaccard similarity with a threshold of 0.85. The resulting RL training pool contains 32,954 prompts. After construction, this pool is frozen and reused unchanged for all downstream comparisons.

For the main 5k-anchor configuration, RL-proxy anchors are drawn from this frozen RL prompt pool by embedding prompts with BGE-M3 (Chen et al., 2024a), applying global k -means with 100 clusters (Lloyd, 1982), and retaining the 50 prompts nearest each centroid. This yields 5,000 anchors. The lower-cost `Ours-1k` bank reuses the same cluster assignments and within-cluster ranking as the main 5k bank, retaining the top 10 prompts per cluster to obtain 1,000 anchors.

The main RL-proxy-based cold-start subset and later ablation variants are constructed after this RL pool and anchor bank have been fixed; we do not rebuild or re-decontaminate the RL pool after these later selections. The anchors are used only to construct RL-proxy gradients and are not part of any final cold-start SFT subset.

A.2 Training Protocol

Each selector produces a 3,000-example cold-start subset. We fine-tune the same base model on each subset to obtain a warm-start model. Within a model scale, the SFT training recipe, checkpoint-selection rule, RL prompt pool, verifier, RL algorithm, and RL hyperparameters are held fixed in the main comparison. Full training and evaluation settings are summarized in Tables 5, 6, and 7.

Cold-start SFT. All SFT runs share the same OpenRLHF + DeepSpeed ZeRO-3 configuration. The loss is next-token cross-entropy (`sft_loss`), the maximum sequence length is 8192 with sample packing, and the optimizer is AdamW with $(\beta_1, \beta_2) = (0.9, 0.95)$, learning rate 5×10^{-6} , cosine-with-min-lr schedule, warmup ratio 0.1, weight decay 0.1, global train batch size 64 (micro-batch 1), and at most 10 epochs. We save and evaluate every 46 optimizer steps (approximately one epoch) and choose the checkpoint with the highest accuracy on the fixed 1,000-example proxy-dev split. The same procedure is used for all selectors and both model scales.

Downstream RL. Starting from each selector’s SFT warm start, we run the same DeepScaleR-style verifiable math RL (Tan et al., 2025). The RL algorithm is a GRPO-style update with group-normalized verifier-reward advantages and no critic. It uses a reference-policy KL regularizer with coefficient $\beta = 0.001$, the k_3 estimator, and discount factor $\gamma = 1.0$. The actor learning rate is 1×10^{-6} with warmup ratio 0.1; each RL step uses train batch size 128, micro-batch 1, prompt maximum length 1024, 8 samples per prompt, and training decoding temperature 1.0. Due to memory constraints, 1.5B uses rollout batch size 128 and generation maximum length 7168, whereas 7B uses rollout batch size 64 and generation maximum length 6144.

Standard runs use one training episode with periodic save/evaluation every 50 steps. The fixed-checkpoint main comparison reports RL50 for 1.5B and RL100 for 7B. The compute-aware runs in §F additionally support explicit early termination at non-multiples of 50; for these runs, we evaluate the final model state produced at the specified termination step rather than requiring a periodic checkpoint at that step.

Our focus is early-to-mid RL sample efficiency, not final-performance dominance at arbitrary training lengths. The warm-up probe model in §4 is used only for offline gradient extraction and RL-proxy subspace construction and never serves as a final SFT warm start.

A.3 Evaluation Protocol

We evaluate on AIME 2024, 2025, 2026, MATH-500, MATH-P-Hard, and OlympiadBench (Hendrycks et al., 2021; He et al., 2024a; Huang et al., 2025). For AIME we sample $N = 64$ responses per problem at $T = 0.6$ and report the mean verifier pass rate over the N samples ($\text{Pass}@1_{N=64}$). For MATH-500, MATH-P-Hard, and OlympiadBench we use deterministic decoding ($T = 0$, $k = 1$) and report verified accuracy. All evaluations use vLLM as the inference backend with at most 8192 new tokens per sample; final answers are judged by a math verifier.

The overall score is the unweighted mean of the six benchmark scores:

$$\text{Avg} = \frac{1}{6} \sum_{b=1}^6 \text{Score}_b.$$

No external evaluation set participates in cold-start

selection, SFT training, proxy-dev checkpoint selection, or downstream RL training. In paired comparisons, problem IDs are identical across methods.

A.4 Bootstrap Analysis

We use paired problem-level bootstrap to estimate evaluation problem-sampling uncertainty. For each replicate we resample problem IDs with replacement *within* each benchmark, recompute each benchmark score, and then recompute the unweighted six-benchmark mean. Pairwise differences use identical resampled problem IDs across methods. All reported results use 10,000 replicates and 95% percentile intervals. This analysis captures evaluation problem-sampling uncertainty and is not equivalent to full training-seed variance.

Table 4 reports RL-Proxy-Coverage against each primary baseline at Qwen2.5-Math-1.5B (RL50) and Qwen2.5-Math-7B (RL100). On 1.5B, the 95% CIs exclude zero against `random` and `sl_style`, while the smaller positive differences against `ppl_low` and `rl_less_mean` do not. On 7B, all four intervals exclude zero.

Across both scales, all four pairwise point differences are positive. The statistical evidence is stronger at 7B, where all four 95% intervals exclude zero; we do not attribute this scale difference to a particular capacity mechanism.

B Hyperparameters and Offline Operations

Hyperparameters are grouped by stage: offline cold-start selection and RL-proxy construction (Table 5), downstream SFT (Table 6), and downstream RL plus external evaluation (Table 7). The two model scales use the same main cold-start selection configuration; downstream SFT is identical; downstream RL differs only in memory-bound rollout batch size and generation length.

Table 8 summarizes the additional offline signals used by the primary selectors. `rl_less_mean` and RL-Proxy-Coverage share the same candidate SFT gradients and RL-proxy gradients and differ in the selection objective.

C Selector and Ablation Details

C.1 Baseline Selectors

All primary baselines select the same number of examples from the same eligible candidate pool and use the same downstream SFT/RL protocol in the fixed-checkpoint comparison.

Scale	vs.	Δ	95% CI	$\Pr(\Delta > 0)$
Qwen2.5-Math-1.5B (RL50)	random	+0.015	[+0.003, +0.027]	0.993
	ppl_low	+0.003	[-0.011, +0.016]	0.656
	rl_less_mean	+0.008	[-0.005, +0.021]	0.892
	sl_style	+0.042	[+0.027, +0.059]	1.000
Qwen2.5-Math-7B (RL100)	random	+0.016	[+0.003, +0.030]	0.991
	ppl_low	+0.019	[+0.006, +0.033]	0.998
	rl_less_mean	+0.022	[+0.009, +0.035]	0.999
	sl_style	+0.016	[+0.002, +0.030]	0.988

Table 4: Paired problem-level bootstrap (10,000 replicates) for RL-Proxy-Coverage vs. each primary baseline. Δ is the six-benchmark mean difference; the last column gives the fraction of bootstrap replicates with $\Delta > 0$.

Component	Hyperparameter	Value
Candidate pool	Parent candidate size	34,540
Candidate pool	Proxy-dev size	1,000
Candidate pool	Eligible candidate size	33,540
Cold-start selection	Selection budget k	3,000
RL training pool	Decontam. RL prompts	32,954
Main RL-proxy anchors	Number of anchors	5,000
Main RL-proxy anchors	Number of clusters	100
Main RL-proxy anchors	Prompts per cluster	50
Ours-1k anchors	Number of anchors	1,000
Ours-1k anchors	Number of clusters	100
Ours-1k anchors	Prompts per cluster	10
RL-proxy rollouts	Responses per anchor K	8
RL-proxy gradients	Nonzero proxy gradients	1,604 (1.5B, main 5k)
Gradient repr.	Projection dim. d	4096
Gradient repr.	Gradient parameters	LoRA
RL-proxy subspace	Subspace rank r	64
Log-det selection	Ridge term ε_{det}	10^{-6}
Proxy RL surrogate	Likelihood-ratio clip	$\delta = 0.2$
Proxy RL surrogate	KL coefficient	$\beta = 0.001$
Proxy RL surrogate	Advantage norm.	group-normalized

Table 5: Cold-start selection and RL-proxy hyperparameters. The main configuration uses 5k anchors; Ours-1k reuses the same 100-cluster ranking and retains the top 10 prompts per cluster. The reported non-zero-gradient count is for the 1.5B main 5k bank.

Component	Hyperparameter	Value
Framework	Backend	OpenRLHF + ZeRO-3
Sequence	Max sequence length	8192 (with sample packing)
Optimizer	Type / (β_1, β_2)	AdamW / (0.9, 0.95)
Optimizer	Learning rate	5×10^{-6}
Optimizer	Weight decay (ℓ_2)	0.1
Scheduler	Type	cosine with min-lr
Scheduler	Warmup ratio	0.1
Batching	Global train batch size	64
Batching	Micro-batch size	1
Training length	Max epochs	10
Checkpointing	Save / eval every	46 steps
Ckpt. selection	Criterion	best proxy-dev accuracy
Loss	Type	next-token cross-entropy
Precision	Mixed precision	bfloat16 + FlashAttention-2
Memory	Gradient checkpointing	enabled
Reproducibility	SFT seed (main)	42

Table 6: Downstream SFT hyperparameters, shared between 1.5B and 7B. Checkpoint selection uses the fixed 1,000-example proxy-dev split.

Component	Hyperparameter	Value
Framework	Backend	OpenRLHF + Ray + vLLM
Algorithm	Type	GRPO-style (no critic)
Algorithm	Advantage estimator	group-normalized
Reward	Verifier	math-verifier
Reference policy	KL coef. / estimator	$\beta = 0.001 / k_3$
Discount	γ	1.0
Optimizer	Actor learning rate	1×10^{-6}
Scheduler	Warmup ratio	0.1
Batching	Train batch size	128
Batching	Micro train batch	1
Rollout	Batch size	128 (1.5B) / 64 (7B)
Rollout	Samples per prompt	8
Rollout	Train decoding T	1.0
Lengths	Prompt max length	1024
Lengths	Gen. max length	7168 (1.5B) / 6144 (7B)
Training length	Episodes	1
Checkpointing	Periodic save / eval	every 50 steps
Cost-aware runs	Termination	explicit early stop
Reproducibility	RL seed (main)	42
Eval (AIME)	Mode / T / k	sampled / 0.6 / 64
Eval (MATH / Olym.)	Mode / T / k	determ. / 0.0 / 1
Eval	Max new tokens	8192
Bootstrap	Replicates	10,000

Table 7: Downstream RL and external-evaluation hyperparameters. Standard runs save/evaluate periodically every 50 steps; the compute-aware comparison uses explicit early termination as detailed in §F.

Selector	Candidate gradients	RL-proxy rollouts	RL-proxy gradients	Selection step
random	–	–	–	uniform sampling
ppl_low	–	–	–	candidate-PPL ranking
sl_style	–	–	–	hard examples + embedding diversity
rl_less_mean	Y	Y	Y	mean RL-proxy direction matching
Ours	Y	Y	Y	SVD subspace + greedy log-det coverage

Table 8: Offline operations per primary selector. `rl_less_mean` and RL-Proxy-Coverage use identical gradient inputs but different selection objectives.

random. `random` uniformly samples 3,000 examples from the eligible candidate pool and serves as the no-signal cold-start baseline.

ppl_low. `ppl_low` ranks candidates by base-model perplexity and retains the lowest-PPL examples, representing an SFT-centric model-confidence criterion.

sl_style. `sl_style` is a hard-and-diverse reasoning-data selector in the style of small curated reasoning datasets (Muennighoff et al., 2025; Ye et al., 2025). It favors harder examples while preserving semantic diversity in prompt-embedding space.

rl_less_mean. `rl_less_mean` uses the same projected SFT gradients and the same non-zero RL-proxy gradients as RL-Proxy-Coverage, averages the proxy gradients into one direction, and selects candidates whose SFT gradients are most similar to that mean. It isolates mean alignment from multi-direction subspace coverage.

RL-Proxy-Coverage. RL-Proxy-Coverage estimates a low-rank local RL-proxy gradient subspace from representative anchors and selects candidates whose SFT gradients provide non-redundant log-determinant coverage within that subspace.

C.2 Ablation Variants

All ablation variants share the same cold-start budget, candidate pool, SFT/RL training protocol, and external evaluation setup and differ only in the offline selection criterion.

sft_subspace. This variant replaces the RL-proxy subspace with a PCA/SVD subspace built from candidate SFT gradients while retaining the same log-determinant coverage objective. It tests RL-proxy target geometry against generic SFT-gradient coverage.

align_bottom. This negative control uses the same RL-proxy subspace as RL-Proxy-Coverage but favors candidates with low alignment to it.

adam_update. This variant uses the warm-up optimizer state to convert per-sample raw gradients into one-step Adam-style marginal update directions and then applies the same selection procedure. It tests optimizer-state-dependent update directions against raw gradients as the selection representation.

D Additional Results

The main text focuses on RL50 for 1.5B and RL100 for 7B, consistent with our target of early-to-mid RL sample efficiency rather than final-score dominance at arbitrary training lengths.

D.1 SFT-to-RL Ranking Relationship

To test whether the SFT-to-RL inversion highlighted in Figure 1 is isolated, we consolidate the five primary selectors and three mechanism variants at each scale.

Scale	Selectors	Spearman ρ	Ours rank	Inversions
1.5B	8	+0.271	7 $\rightarrow$ 1	6/7
7B	8	-0.339	5.5 $\rightarrow$ 1	4/7

Table 9: Relationship between SFT-only and post-RL rankings. “Ours rank” reports SFT-only $\rightarrow$ post-RL rank; “Inversions” counts competing selectors ranked above RL-Proxy-Coverage before RL but below it afterward.

The relationship is weakly positive on 1.5B and negative on 7B. At both scales, RL-Proxy-Coverage moves from a relatively low SFT-only rank to the highest post-RL rank. These repeated inversions show that SFT-only ordering is not a reliable proxy for post-RL ordering among the tested selectors.

D.2 Extended 1.5B Learning Curve

Table 10 extends the RL trajectory on Qwen2.5-Math-1.5B. Due to compute constraints, we compare the two runs for which complete RL50, RL100, and final checkpoints are available: `random` and RL-Proxy-Coverage, trained from the same base model with the same RL dataset and hyperparameters. *Final* denotes the end-of-training checkpoint (global step $\approx$ 257).

Method	RL50	RL100	Final ($\approx$ 257)
<code>random</code>	0.294	0.304	0.312
RL-Proxy-Coverage	0.309	0.307	0.312

Table 10: Extended RL trajectory on Qwen2.5-Math-1.5B. Entries are the unweighted six-benchmark mean.

RL-Proxy-Coverage is ahead of `random` at RL50 and remains slightly ahead at RL100; the two runs converge to the same reported mean at the final checkpoint. This is consistent with an early-to-mid initialization benefit that can diminish as additional RL signal accumulates.

For Qwen2.5-Math-7B we report RL100 in the main text rather than a longer trajectory. Because the 7B rollout batch size is half that of 1.5B, RL100 at 7B consumes approximately the same amount of RL training data as RL50 at 1.5B. We therefore interpret these as approximately matched early-to-mid RL-data budgets rather than identical training lengths.

D.3 Seed Sensitivity at RL50 (Qwen2.5-Math-1.5B)

To assess sensitivity to pipeline randomness at the early RL checkpoint, we replicate the full SFT $\rightarrow$ RL pipeline for `random` and RL-Proxy-Coverage with a second seed. The evaluation protocol is identical to the main text. Due to compute constraints, we report $n = 2$ seeds per method.

Method	seed 42	seed 43	Mean (range)
<code>random</code>	0.294	0.254	0.274 (± 0.020)
Ours	0.309	0.291	0.300 (± 0.009)

Table 11: Seed sensitivity at RL50 on Qwen2.5-Math-1.5B. The reported spread is the half-range $(x_{\max} - x_{\min})/2$, not a sample standard deviation.

The ordering is preserved under the second seed: RL-Proxy-Coverage exceeds `random` for both seed 42 and seed 43. RL-Proxy-Coverage also has a smaller observed half-range in this two-seed check. Because $n = 2$ is insufficient for a variance estimate, we treat this only as an exploratory robustness check.

E Additional Diagnostics

This section examines whether the selected subsets differ substantially in static sample properties and whether their projected gradients exhibit the intended local RL-proxy geometry.

E.1 Static Properties of Selected Subsets

Table 12 summarizes static properties of the five primary cold-start subsets. Statistics are computed over the 3,000 examples entering SFT and joined by sample ID to parent-pool diagnostic fields. “Pass@8” is the verifier success rate of the base Qwen2.5-Math-1.5B over 8 sampled completions per candidate prompt; lower values indicate harder prompts for the base model.

`ppl_low` substantially lowers PPL, while `sl_style` selects markedly harder examples and shifts the source mixture toward AoPS. In

contrast, RL-Proxy-Coverage is close to `random` and `rl_less_mean` on these static properties, making a simple explanation based solely on PPL, difficulty, length, or a single source-bucket shift unlikely.

E.2 RL-Proxy Subspace Alignment

Table 13 reports the squared projection norm $\|z_i\|_2^2$ of each selected SFT gradient onto the $r = 64$ local RL-proxy subspace.

RL-Proxy-Coverage has the largest projection norms, whereas `align_bottom` has substantially smaller values, confirming that the latter acts as an effective low-alignment control in the constructed proxy geometry.

E.3 Redundancy and Coverage

Strong projection onto the proxy subspace does not by itself imply broad coverage. Table 14 therefore reports the average pairwise absolute cosine, effective rank of the projected covariance, and log-determinant objective.

Compared with `rl_less_mean`, RL-Proxy-Coverage combines larger projection magnitude with substantially higher log-determinant coverage and lower pairwise redundancy. This pattern is consistent with the intended mechanism of retaining multiple RL-proxy directions rather than concentrating examples around one mean direction.

E.4 Rank Sensitivity

The main experiments use subspace rank $r = 64$, 5,000 anchors, and projection dimension $d = 4096$. Table 15 varies only r in the 1.5B RL50 setup.

Among the tested values, $r = 64$ performs best. Broader or adaptive rank selection remains an open question.

F Cost Analysis

We first report a reproducible estimate of the main 5k-anchor offline pipeline on the 1.5B setup, using $N = 8$ responses per anchor, projection dimension $d = 4096$, and 4×H800 80GB. This estimate is distinct from the realized selector-plus-RL logged totals reported in Table 3.

Anchor-count scaling. The dominant anchor-side stages—proxy rollouts and RL-proxy gradient extraction—scale approximately linearly with the number of anchors m . The candidate-side gradient bank instead scales with candidate-pool size n and costs about 2 GPU-h in the reported 1.5B setup.

Method	PPL	Pass@8	Prompt toks.	Response toks.	Olym. %	CN %	AoPS %
random	0.767	0.204	83.7	4027.4	75.9	11.3	10.3
ppl_low	0.516	0.209	82.7	3489.1	75.1	12.7	8.3
sl_style	0.819	0.057	90.9	3818.4	77.4	4.2	16.2
rl_less_mean	0.737	0.198	82.6	3960.1	76.7	11.5	9.3
RL-Proxy-Coverage	0.773	0.213	87.9	3808.9	76.9	11.5	9.0

Table 12: Static properties of the five 3,000-example primary cold-start subsets. *PPL*: parent-pool assistant-token mean NLL; *Pass@8*: base-model 8-sample verifier success rate; *Prompt/Response toks.*: tokenizer counts; *Olym./CN/AoPS*: shares from the olympiads, cn_contest, and aops_forum source categories, respectively. The remaining parent-pool categories are amc_aime, inequalities, number_theory, and olympiads_ref.

Method	Mean	Median	P25	P75
random	0.160	0.154	0.131	0.182
ppl_low	0.156	0.151	0.129	0.176
sl_style	0.160	0.154	0.130	0.183
rl_less_mean	0.180	0.174	0.149	0.204
RL-Proxy-Coverage	0.205	0.198	0.174	0.227
align_bottom	0.099	0.102	0.094	0.107

Table 13: Squared projection norms $\|z_i\|_2^2$ in the local $r = 64$ RL-proxy subspace. Larger values indicate stronger alignment with this locally estimated geometry.

Method	Pairwise cos	Eff. rank	Log-det
random	0.150	39.6	99.6
ppl_low	0.147	40.8	99.3
sl_style	0.152	39.1	99.8
rl_less_mean	0.225	29.9	98.2
RL-Proxy-Coverage	0.135	44.2	121.2
align_bottom	0.122	51.0	83.4

Table 14: Redundancy and coverage diagnostics in the $r = 64$ local RL-proxy subspace. Lower pairwise |cos| and higher effective rank indicate lower directional redundancy; larger log-det indicates broader volume coverage.

Setting	Value	RL50 Avg.
Rank	$r = 32$	0.302
Rank	$r = 64$ (ours, main)	0.309
Rank	$r = 128$	0.290

Table 15: Sensitivity of RL50 mean accuracy to the subspace rank r on Qwen2.5-Math-1.5B. Only r varies.

Stage	Hardware	GPU-h
Warm-up probe	1 GPU	≈ 0.5
RL-proxy rollouts	4 GPU	≈ 46
RL-proxy gradient	4 GPU	≈ 9
SFT-side gradient	4 GPU	≈ 2
Selection (SVD + greedy)	CPU	< 0.5
Total		≈ 57

Table 16: Reproducible offline-cost estimate for the 1.5B main 5k-anchor configuration on 4×H800 80GB.

Reducing m from 5,000 to 1,000 therefore reduces the estimated anchor-side cost from roughly 55 GPU-h to roughly 11 GPU-h, yielding an approximate 14 GPU-h offline total after including the remaining stages.

The 1k configuration used in the compute-aware experiment reuses the clustering underlying the main 5k bank and retains only the top 10 ranked prompts per cluster. It is therefore a structured 1k subset of the main anchor bank rather than an independently reclustered or randomly sampled bank.

Prospective compute-aware allocation. The compute-aware comparison fixes the allocation before external benchmark evaluation. Ours-1k uses RL50 as the 1.5B reference point and RL100 as the 7B reference point. random is allocated 40 additional RL steps, giving termination steps 90 and 140. ppl_low is allocated 33 additional steps, giving termination steps 83 and 133; the seven-step difference from random accounts for the offline PPL-scoring cost in the prospective allocation.

These values are explicit early-termination points, not periodic save/evaluation checkpoints. Standard periodic save/evaluation remains every 50 steps; the cost-aware runs terminate at the specified step and evaluate the resulting final model state.

Logged selector-plus-RL accounting. Table 3 reports realized logged totals, including method-specific offline GPU stages and downstream RL GPU occupancy through the corresponding termination point. We exclude costs shared across methods: common dataset construction, final 3k SFT, CPU-only selection, and external benchmark evaluation. The “+” marker denotes a conservative logged lower bound. Under this accounting boundary, Ours-1k has the highest reported average and the lowest logged GPU-hour total at both scales.

This accounting is GPU-hour based and is not ex-

actly matched by generated tokens or verifier calls. Such hardware-independent accounting would provide a complementary efficiency comparison.

Amortization. Within a fixed model and RL prompt pool, the candidate-gradient bank, anchor rollouts, and RL-proxy gradient bank can be reused across selector and ablation analyses that require the same underlying signals. We therefore avoid assigning a fixed per-experiment amortized number, since the amortization factor depends on how many selectors or diagnostics reuse the bank.

G Reproducibility and Reporting Notes

Several details are important for reproducing the comparisons. First, all selectors should draw from the same eligible candidate pool after proxy-dev exclusion and preprocessing. Second, final warm starts should be obtained by SFT from the same base model; the warm-up probe from §4 is used only for offline gradient extraction. Third, fixed-checkpoint downstream RL comparisons should share the same prompt pool, verifier, RL algorithm, and hyperparameters. Fourth, external evaluation data should not participate in cold-start selection, SFT training, checkpoint selection, or RL training.

For `Ours-1k`, the proxy bank should be derived from the main 5k clustering by retaining the top 10 ranked prompts within each of the 100 clusters. For the compute-aware comparison, `RL83/RL90/RL133/RL140` denote explicit early-termination steps whose final model states are evaluated; they are not periodic 50-step checkpoints.

Our main claims concern early-to-mid RL sample efficiency. The extended 1.5B trajectory shows that the initialization advantage can diminish with longer RL, and we do not claim permanent final-score dominance. Likewise, the first-order argument and RL-proxy subspace are local constructions; downstream utility is established empirically rather than through a multi-step convergence guarantee.

Finally, paired bootstrap intervals quantify evaluation problem-sampling uncertainty rather than full training-seed variance. Stronger variance estimates would require repeating the complete SFT-to-RL pipeline across more independent seeds and selectors.