

Where Reasoning Branches: How Preference Pair Construction Shapes DPO for Mathematical Reasoning

Stanford CS224N Custom Project

Duy Nguyen

Department of Computer Science
Stanford University
duynguy@stanford.edu

Abstract

We originally wanted to test whether a co-evolving strategy scaffold could leave a lasting imprint on model weights during RL training for math reasoning. It did not—at 7B–8B scale the model learned to lean on the scaffold’s structure instead of internalizing its content, and our attempts to detect reasoning-critical branch points through token entropy and hidden-state probes also came up short. What we were left with was the data infrastructure we had built: rollouts, error-position identification, and correction generation. We used it to compare two ways of constructing DPO preference pairs on Qwen3-8B over MATH-500. Natural rollout pairs (correct vs. incorrect solutions to the same problem) consistently helped; correction pairs (self-repair continuations at identified error positions) consistently hurt. A matched replication on the correction side brought the total to 11 conditions; 10 of 11 went the way we expected. When we ablated what in the correction data was actually causing the drop, it turned out to be the failed-reasoning prefix, not the “wait, let me reconsider” rhetoric we initially suspected.

1 Key Information

- Mentor: Mirac Suzgun
- External collaborators: None
- Sharing project: No

2 Introduction

This project started from Dynamic Cheatsheet [1], which maintains a persistent strategy scaffold in context during inference. We asked whether co-evolving such a scaffold with RL weight updates could produce durable weight absorption—strategies that persist after the scaffold is removed. A kill-test at 3B and 7B scale (Appendix A) showed the opposite: at this scale the model learned to condition on the scaffold’s structure rather than absorb its content into weights. We then tried to identify where reasoning branches—positions where the model’s next token choice determines whether it reaches the right answer—using token entropy and hidden-state probes, but neither produced a reliable signal (42% false-positive rate on token entropy; hidden-state probes could not beat a position-only baseline).

The scaffold and probe work did not pan out, but the infrastructure we built for it—rollouts, error-position identification via DEEP-GRPO [2], and correction generation—turned out to be useful for a different question. We ended up comparing two ways of constructing preference pairs for DPO [3], and that comparison is what this paper reports. Lai et al. [4] applied DPO to math reasoning by having GPT-4o identify the first wrong step and generate a correction. Kumar et al. [5] showed that

SFT on self-corrections fails from distributional mismatch, but since DPO learns relative preferences rather than imitating targets, it was not obvious the same problem would apply.

We built preference pairs two ways on MATH-500 with Qwen3-8B. Correction-based pairs truncate an incorrect rollout at an identified error position and generate a self-repair continuation; all 3 correction conditions hurt performance (-0.8 to -3.8 pp). Rollout-based pairs take a naturally sampled correct solution and an incorrect one to the same problem, with no correction generation or external judge; all 5 rollout conditions help ($+0.6$ to $+3.2$ pp). We ran a matched-hyperparameter replication on the correction side, bringing the total to 11 conditions. Ten of 11 move in the predicted direction (sign test $p = 0.012$; Demsar [6]). Individual comparisons are underpowered and the two experiments use different baselines, so we discuss these caveats in §5.4. An ablation (§6) traces the correction failure to the failed-reasoning prefix that precedes each correction: stripping the self-repair rhetoric alone does not help, but removing the failed prefix recovers performance to baseline.

3 Related Work

Three bodies of prior work are relevant: preference data construction, error-position identification in reasoning, and the failure modes of self-correction training.

On the data side, Djuhera et al. [7] argue that the dataset *is* the algorithm—that DPO outcomes depend more on how pairs are constructed than on optimizer details. Ivison et al. [8] and Pan et al. [9] provide supporting evidence from different angles: sampling strategy and chosen-response quality, respectively. Our experiments were not designed to test this, but the correction-vs-rollout split is a concrete case where data construction flips the outcome sign.

For error-position identification, we build on Step-DPO [4], which uses GPT-4o to find the first wrong step and generate a correction. The step-level verification idea goes back to Lightman et al. [10]; Math-Shepherd [11] and OmegaPRM [12] automated it at scale. We use DEEP-GRPO’s [2] depth-biased selection instead of an external judge, which is cheaper but limits us to problems where the model already produces both correct and incorrect rollouts. GRPO [13] provides the underlying rollout infrastructure. We do not compare against Step-DPO directly (different scale, model, and task split).

Our negative results on correction pairs connect to the self-correction failure literature. Kumar et al. [5] showed that SFT on model-generated corrections fails from distributional mismatch, and Huang et al. [14] found that correction prompts bias models toward changing answers even when the original was right. We expected DPO’s contrastive objective to be less susceptible, since it does not directly imitate the chosen text. It was not—correction pairs degraded performance in all 3 of our conditions, and our ablation (§6) traces the problem to the failed-reasoning prefix rather than the correction rhetoric itself.

4 Approach

4.1 Rollout Generation and Filtering

We evaluate Qwen3-8B [15] on MATH-500 [16]. For each of the 500 problems, we generate $N = 8$ rollouts using vLLM [17] with temperature $T = 0.7$. We retain problems where at least one rollout is correct and at least one is incorrect ($1 \leq n_{\text{correct}} < N$), since constructing preference pairs requires both outcomes. This yields 141 problems (28.2%); the remaining 311 (62.2%) are solved by all rollouts and 48 (9.6%) by none.

4.2 Preference Pair Construction

We construct preference pairs from the mixed-outcome set using two data sources (Figure 1).

Correction-based pairs. We identify error positions using DEEP-GRPO’s [2] depth-biased selection:

$$Q(t) = P_{\phi}(\text{success} \mid r_t) \cdot \left(\frac{t}{T}\right)^{\gamma}, \quad \gamma = 2 \quad (1)$$

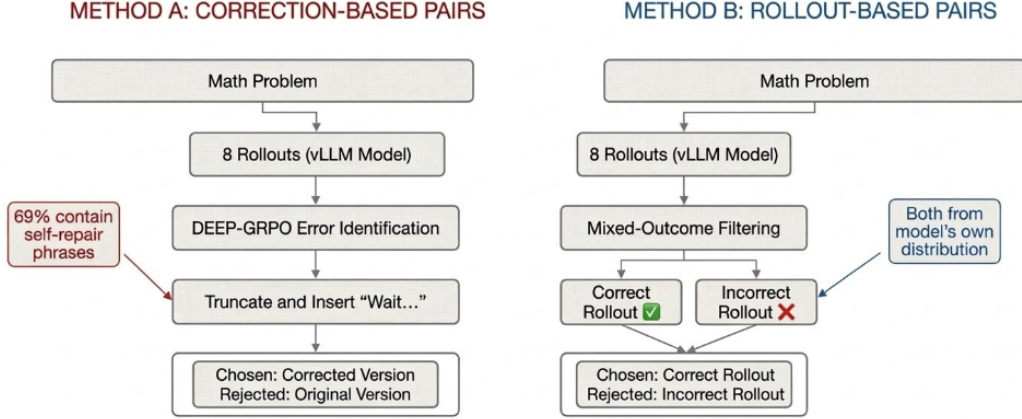


Figure 1: The two data-construction pipelines. Correction-based pairs inject out-of-distribution “Wait...” text; rollout-based pairs use only the model’s own generations.

where $P_\phi = \sigma(w \cdot r_t + b)$ is a logistic regression on normalized depth $r_t = t/T$, fit online with bootstrap initialization on the first 10 problems. Segmentation uses paragraph boundaries ($\backslash n \backslash n$) rather than fixed token chunks. At each error position, we truncate the incorrect rollout and prompt the model to generate a correction starting with “Wait, let me reconsider...” The chosen response is the corrected text; the rejected is the original incorrect continuation. This produces 189 pairs per condition across three pair-selection strategies: random subset, late divergence ($> 10\%$ depth), and early divergence ($\leq 10\%$ depth).

Rollout-based pairs. For each mixed-outcome problem, we pair a naturally sampled correct rollout with a naturally sampled incorrect one. No correction generation, no external judge—both chosen and rejected responses come from the model’s own generation distribution. This produces 150–228 pairs depending on filtering. We test two structuring variants: **full-trajectory** (problem as prompt, entire solutions as chosen/rejected) and **prefix-split** (extend the prompt with the shared prefix up to the divergence position identified during rollout comparison, snapped to paragraph boundaries, and use only the post-divergence suffixes). The prefix-split variant is motivated by an observation about DPO’s gradient (§6).

5 Experiments

5.1 Data

We use MATH-500 [16], which contains 500 problems spanning 5 difficulty levels and 7 subjects. After mixed-outcome filtering ($1 \leq n_{\text{correct}} \leq 7$ out of 8 rollouts), 141 problems (28.2%) yield usable preference pairs, with a mean accuracy of 72.9% across all rollouts.

5.2 Evaluation Method

We evaluate greedy pass@1 ($T = 0$) on all 500 problems. Each condition is compared against its base model using McNemar’s test [18], which counts problems where exactly one model is correct and tests whether those disagreements are one-sided; Demšar [6] recommends this test for paired classifier comparisons on shared evaluation sets. Within each experiment we apply Benjamini-Hochberg correction [19] to control the false discovery rate across multiple comparisons. Because individual comparisons are underpowered at this scale, we also report the sign test [6] over the full set of conditions to check directional consistency.

Table 1: Rollout-based DPO results. Base: SFT checkpoint (74.2%). All use $\beta = 0.1$. *Raw $p < 0.05$ (adjusted $p = 0.107$).

Model	Type	Pairs	Pass@1	Δ	Raw p
SFT base	—	—	74.2% (371/500)	—	—
Rollout 1	prefix-split	228	74.8% (374/500)	+0.6pp	0.761
Rollout 2	full-traj, 1ep	150	75.2% (376/500)	+1.0pp	0.511
Rollout 3	full-traj, 3ep	150	75.6% (378/500)	+1.4pp	0.349
Rollout 4	prefix-split	150	76.6% (383/500)	+2.4pp	0.036*
Rollout 5	full-traj, 1ep	228	77.4% (387/500)	+3.2pp	0.026*

Table 2: Correction-based DPO results. Base: SFT checkpoint (77.4%). $\beta = 0.2$, 3 epochs, 189 pairs. * $p < 0.05$.

Model	Pair Selection	Pass@1	Δ	p	
SFT base	—	77.4% (387/500)	—	—	
Correction 1	Random subset	73.6% (368/500)	−3.8pp	0.016*	Correction 3 vs
Correction 2	Late div. (>10%)	75.0% (375/500)	−2.4pp	0.119	
Correction 3	Early div. ($\leq 10\%$)	76.6% (383/500)	−0.8pp	0.678	

Correction 1: +3.0pp, $p = 0.036^*$.

5.3 Experimental Details

All experiments use Qwen3-8B with QLoRA [20] (LoRA [21], $r = 64$, $\alpha = 128$). The two experiments use different training pipelines:

- **Correction DPO:** starts from an SFT checkpoint (Qwen3-8B fine-tuned on 465 clean traces, 77.4% pass@1), $\beta = 0.2$, learning rate 5×10^{-6} , 3 epochs, 189 pairs per condition.
- **Rollout DPO:** starts from the same SFT checkpoint architecture but a separate merge yielding 74.2% pass@1, $\beta = 0.1$, learning rate 2×10^{-5} , 1–3 epochs, 150–228 pairs per condition.

Note that two experiments were run weeks apart as the project pivoted from correction-based to rollout-based pairs; the hyperparameter differences (β , learning rate, epochs) reflect evolving design choices rather than a controlled comparison.

Because the baselines (77.4% vs 74.2%), hyperparameters, and epoch counts differ, we cannot directly compare effect sizes across the two experiments. We use the sign test (§5.4) for directional consistency and discuss the need for a matched replication in §7.

5.4 Results

Rollout-based DPO. Table 1 shows the rollout conditions. Every condition outperforms the SFT base (74.2%), ranging from +0.6 to +3.2pp. Two reach raw $p < 0.05$ by McNemar’s test, but neither survives Benjamini-Hochberg correction (adjusted $p = 0.107$)—not surprising with only 150–228 training pairs.

Correction-based DPO. All three correction conditions (Table 2) scored below the SFT base (77.4%). Random pair selection does worst (−3.8pp, $p = 0.016$), while early-divergence loses only −0.8pp ($p = 0.678$). The gap between the two is significant (+3.0pp, $p = 0.036$), likely because earlier error positions preserve more correct reasoning before the self-repair text begins.

Directional consistency. With 500 evaluation problems and effect sizes of 1–3pp, individual McNemar tests are underpowered, so we also report directional consistency. Before the matched replication, all 3 correction conditions fell below their base and all 5 rollout conditions exceeded theirs (8/8, $p = 0.008$ by sign test; Demsar [6]). After adding 3 replication conditions, one of which crosses slightly above base, the count is 10/11 ($p = 0.012$).

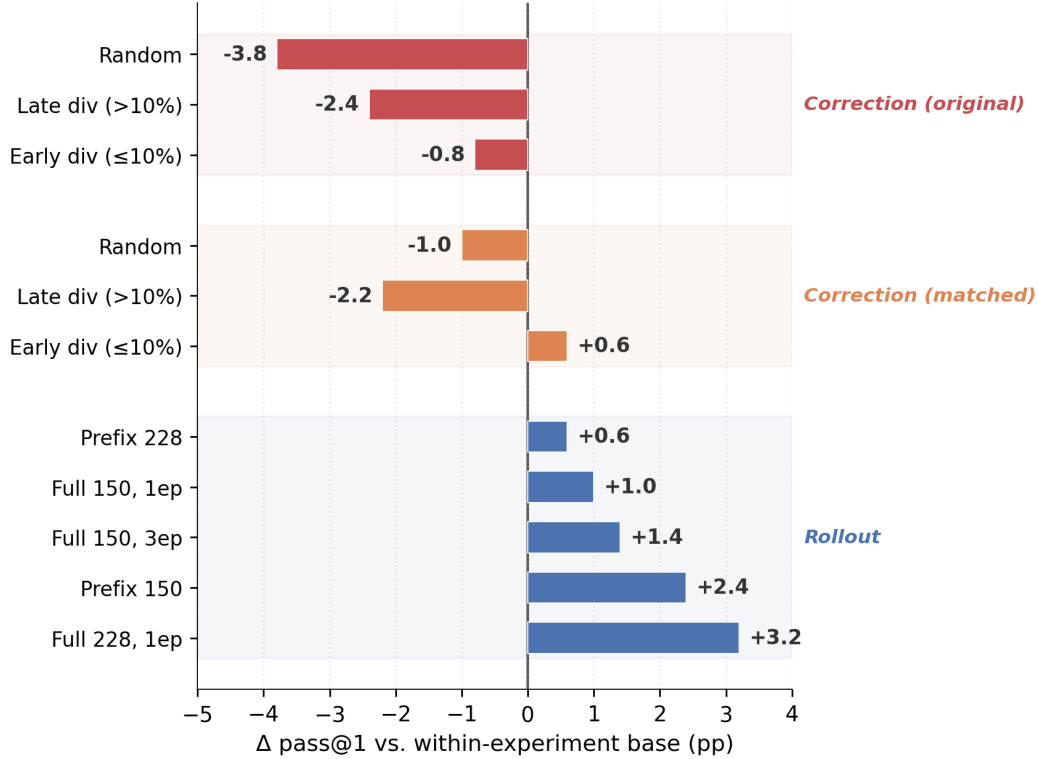


Figure 2: Pass@1 across all conditions. Left: correction-based (most below base). Right: rollout-based (all above base). Dashed lines show respective base models.

We note two caveats on the sign test. First, the two experiments use different SFT checkpoints (77.4% vs 74.2%), different β , learning rate, and epoch counts, so the 11 conditions are not drawn from the same experimental setup. Second, conditions within each experiment share the same evaluation set, inflating their non-independence.

Matched replication. To partially address the confound, we re-ran the 3 correction conditions with matched hyperparameters ($\beta = 0.1$, $\text{lr} = 2 \times 10^{-5}$, 1 epoch) starting from a fresh SFT checkpoint (76.8%). Random dropped -1.0pp ($p = 0.560$), late-divergence -2.2pp ($p = 0.144$), and early-divergence rose $+0.6\text{pp}$ ($p = 0.771$). Two of three conditions went negative; the early-divergence exception ($+0.6\text{pp}$, $p = 0.771$) is well within noise. Including the replication, the overall directional count is 10/11 ($p = 0.012$ by sign test).

Null result: Full-trajectory $\approx$ prefix-split. With 150 pairs, prefix-split leads by $+1.4\text{pp}$ ($p = 0.311$); with 228 pairs, full-trajectory leads by $+2.6\text{pp}$ ($p = 0.066$). Neither difference is significant, and the direction reverses between the two pair counts. Section 6 discusses why the two approaches produce similar results.

6 Analysis

Distributional mismatch in corrections. Our correction pipeline prepends a fixed transition phrase (“Wait, let me reconsider this step...”) before generating continuations. In the correction data, roughly 69% of generated corrections contain self-repair rhetoric (“actually,” “wait,” “let me re-evaluate”). Less than 1% of the model’s natural generations contain this kind of language. When DPO trains on correction pairs, it learns to prefer responses with self-repair markers, and we observed verbose hedging at inference even on problems where the original reasoning was correct. Kumar et al. [5] documented the same distributional mismatch for SFT; we see it here in DPO as well.

Table 3: Ablation on late-divergence correction pairs. Base: 76.0% (380/500). All use $\beta = 0.1$, 1 epoch.

Variant	Pass@1	Δ	McNemar p
Base (SFT checkpoint)	76.0% (380/500)	—	—
Neutralized rhetoric	74.0% (370/500)	−2.0pp	0.212
Neutralized + no prefix	75.8% (379/500)	−0.2pp	1.000

Ablation: rhetoric vs. prefix. To pin down what in the correction data actually causes the drop, we ran two ablations on the late-divergence condition (Table 3). First, we neutralized the self-repair rhetoric by replacing “Wait, let me reconsider. . .” with a neutral continuation marker, keeping the failed prefix intact. This did not help: pass@1 dropped −2.0pp ($p = 0.21$), nearly identical to the unmodified correction (−2.2pp in the original experiment). Second, we removed the failed prefix entirely, presenting only the post-divergence correction as the chosen response. This recovered performance to within −0.2pp of base ($p = 1.0$, discordant pairs 23 vs 24).

So the self-repair rhetoric (“wait,” “actually”) does not seem to be what causes the drop—the failed reasoning that precedes the correction is the more likely culprit. When DPO sees a chosen response that begins with incorrect reasoning followed by a course correction, it learns to associate that incorrect context with the preferred signal. Stripping the prefix removes this confound and the correction data stops hurting.

Distributional alignment in rollouts. In rollout pairs, both sides come from the same sampling process (same temperature, same model, no injected text). Both the chosen and rejected responses come from the same sampling process, so neither contains injected text or external edits. The chosen response happens to be correct, but stylistically it is indistinguishable from the model’s normal output. Pan et al. [9] argue that chosen-response quality is what drives DPO gains; rollout pairs satisfy that without the distributional artifacts we see in corrections.

Shared-prefix cancellation. In the rollout-based setup, the chosen and rejected responses share an identical prefix up to the divergence point. DPO’s gradient is $\nabla \log \pi(y_w) - \nabla \log \pi(y_l)$; shared-prefix tokens contribute the same value to both terms and cancel. SFT has no such cancellation—it reinforces all tokens including the shared prefix. In our data, Qwen3-8B traces average ~ 800 tokens and divergence is often past 90% depth, so the prefix-to-suffix ratio is around 8–10 $\times$. Under SFT, most of the gradient comes from that long shared prefix rather than the short suffix where the answers actually differ.

If this cancellation is real, manually stripping the shared prefix (prefix-split DPO) should not help much over full-trajectory DPO. In our experiments (§5.4), the two are statistically indistinguishable, which is consistent with the argument though not proof of it.

Depth and correction yield. Correction yield varies with error-position depth: early positions ($< 20\%$ depth) yield around 48% correct corrections, dropping to around 29% for late positions ($> 80\%$ depth). Early-divergence selection (−0.8pp) outperforming random (−3.8pp) in the correction experiment fits with this: earlier positions preserve more correct prefix, so the chosen response is higher quality despite the self-repair artifacts.

Reproducibility. Two separate evaluation runs of rollout DPO models produced different absolute base scores (76.0% vs 74.2%) but identical directional patterns—all rollout conditions above base in both runs. Per-run variance of ~ 1.8 pp likely reflects vLLM non-determinism and LoRA merge differences. The paired test controls for this by comparing outcomes on the same 500 problems within each run.

7 Conclusion

Correction-based DPO hurt in 3/3 of our conditions, rollout-based DPO helped in 5/5, and a matched replication brought the overall count to 10/11 ($p = 0.012$ by sign test, though conditions are not

independent and baselines differ across experiments). We went into this expecting corrections to be better under DPO than under SFT, since DPO does not directly imitate the chosen text but turned out to be wrong—the correction data degraded performance either way. An ablation (§6) narrows the cause to the failed-reasoning prefix in correction pairs: removing the self-repair rhetoric alone does not help, but stripping the entire failed prefix recovers performance to baseline.

The biggest gap in this work is that the correction and rollout experiments started from different checkpoints with different hyperparameters. The matched replication covers the correction side, but we still lack a rollout run from that same checkpoint, which means we cannot fully separate the data-source effect from training configuration. We also tested only one model (Qwen3-8B), one benchmark (MATH-500), one seed per condition, and 150–228 pairs; no individual comparison survives Benjamini-Hochberg correction.

If we had more compute, the first thing we would run is the fully matched version: correction and rollout DPO from the same checkpoint, same β , same learning rate. The ablation results raise an interesting possibility though: if corrections are generated as standalone responses rather than continuations of failed reasoning, they might work as well as rollout pairs. We did not have the compute to test this at scale. Whether rollout-based gains hold at 1K–10K pairs or transfer to other model families is an open question we did not get to test.

A Appendix: Research Trajectory

The final design emerged from several earlier directions that did not produce usable results at the 7B–8B scale we operated at; some of these might work at larger scale or with more compute.

Phase A: Scaffold co-evolution. Inspired by Dynamic Cheatsheet [1], we tested whether a self-curating strategy scaffold could co-evolve with GRPO training to preserve reasoning diversity. We used a kill-test: train with the scaffold, evaluate without it. At 7B scale, removing the scaffold collapsed performance from 0.363 to 0.079 pass@1 (Table 4). At this scale, the model learned to depend on the scaffold rather than absorb it, though larger models with stronger in-context learning might behave differently.

Table 4: Scaffold kill-test (Phase A). “Removed” = scaffold-trained model evaluated without scaffold.

Scale	Condition	pass@1	pass@8	Oracle Gap
3B	RL-only	0.513	0.900	0.387
3B	RL + scaffold	0.487	0.867	0.380
3B	Scaffold removed	0.266	0.766	0.500
7B	RL-only	0.553	0.940	0.387
7B	RL + scaffold	0.363	0.700	0.337
7B	Scaffold removed	0.079	0.433	0.354

Phase B: Token-level fork signals. Wang et al. [22] identified “fork tokens”—positions where correct and incorrect reasoning paths diverge—as driving RLVR learning. We tested whether token entropy could identify these positions, but it produced a 42% false-positive rate and 84% of flagged tokens had zero outcome variance. Token entropy measures next-token uncertainty, not whether the branch actually leads to a different answer; a richer signal (e.g., outcome variance across rollouts) might work but was outside our compute budget.

Phase C: Fork detection via hidden states. We tested five methods for single-pass fork detection (Table 5). None exceeded a position-only baseline (AUROC, area under the receiver operating characteristic curve, = 0.901). The best probe (AUROC = 0.802) was picking up on the fact that forks tend to occur at certain depths, not on what makes a position a fork; with more data or larger models the hidden-state signal might become separable, but we could not demonstrate it here.

Table 5: Five single-pass fork detection methods (Phase C). None exceeds a position-only baseline.

Method	Signal	AUROC	Status
Token entropy	Next-token H	0.58 (42% FPR)	Failed
Attention divergence	Per-head JSD	0.480	Failed
Fork probe v1	Hidden state classifier	0.551	Failed
Fork probe v2	Probe + position control	0.802 [†]	Failed
Activation change	Norm change	0.480	Failed

baseline: AUROC = 0.901 > 0.802.

Phase D: DEEP-GRPO error-position pipeline. Since single-pass detection did not work at this scale, we moved to rollout-based identification using DEEP-GRPO [2]. The pipeline successfully identified error positions and generated corrections, but training on those corrections degraded performance (see below).

Phase E: From corrections to rollouts. SFT on correction data hurt performance (Appendix C), and DPO on the same correction data also hurt (Table 2). Removing correction-derived text and pairing natural rollouts reversed the effect (Table 1), which pointed to the data source as the key variable rather than the training algorithm.

B Appendix: Pipeline Analysis

Error-position identification. We use DEEP-GRPO’s [2] depth-biased position selection on the 141 mixed-outcome problems. The logistic recoverability estimator P_ϕ is fit online: uniform sampling for the first 10 problems, refit every 10 problems thereafter. Segmentation uses `\n\n` (paragraph) boundaries.

Correction yield. Starting over from position 0 (root resample) corrects 58.2% of traces; correcting mid-trajectory at DEEP-GRPO positions corrects only 33.6% (Table 6). The depth bias ($\gamma = 2$) selects late positions (mean depth 0.709) where the model has less room to recover.

Table 6: Correction yield by method.

Method	Yield	$n_{\text{correct}}/n_{\text{total}}$	Mean Depth
Root resample (Best-of- N)	58.2%	4,876 / 8,384	0.0
First divergence (oracle)	48.2%	1,779 / 3,688	varies
Random position	37.3%	1,376 / 3,688	~ 0.50
DEEP-GRPO ($\gamma = 2$)	33.6%	1,323 / 3,936	0.709

Yield by difficulty and subject. Level 1 problems have the lowest correction yield (21.0%) because most are already solved by all 8 rollouts and only enter the mixed-outcome set when a single rollout fails—leaving little room for correction to improve on. By subject, algebra corrections succeed most often (63.2%) and geometry least (32.0%), as shown in Table 7.

Table 7: Correction yield by difficulty level (left) and subject (right).

Level	Yield	n	Subject	Yield	n
1	21.0%	176	Algebra	63.2%	280
2	55.0%	400	Count. & Prob.	41.6%	392
3	63.0%	648	Geometry	32.0%	440
4	55.7%	776	Intermed. Algebra	43.6%	1,336
5	42.5%	1,936	Number Theory	67.3%	392
			Prealgebra	50.4%	520
			Precalculus	57.3%	576

C Appendix: SFT Ablation

We first tried SFT on correction data before switching to DPO. Adding 116 correction traces to 465 clean traces (Model B in Table 8) lost 6.5pp on recovery rate and 5.0pp on pass@1 relative to clean-only SFT (Model A). A volume control with 2000 oversampled clean traces (Model C) matched the untrained base, suggesting the clean traces themselves had limited value beyond establishing the SFT checkpoint.

Table 8: SFT ablation on correction data. QLoRA $r=64$, 5 epochs.

Model	Training Data	Recovery	Pass@1
Base	None (Qwen3-8B)	0.575	0.400
A	Clean traces (465)	0.591	0.550
B	Clean + corrections (581)	0.526	0.500
C	Volume control (2000)	0.561	0.400

After this, we tried DPO on the same correction data, reasoning that a contrastive objective would not directly imitate the correction text and might avoid copying the out-of-distribution artifacts. As reported in Table 2, the correction-based DPO pairs also hurt, though less severely than SFT.

D Appendix: MATH-500 Rollout Statistics

Table 9: Rollout statistics ($N = 8$ per problem, Qwen3-8B).

Metric	Value
Total problems	500
Mean accuracy	72.9%
Mixed-outcome ($1 \leq \text{correct} \leq 7$)	141 (28.2%)
All correct (8/8)	311 (62.2%)
All incorrect (0/8)	48 (9.6%)
Total corrections generated	3,936
Unique error positions	492
Correct corrections	1,323 (33.6%)
Problems with ≥ 1 correct correction	125/141 (88.7%)
Mean pairwise diversity	0.617 ± 0.110

References

- [1] Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. *arXiv preprint arXiv:2504.07952*, 2025.
- [2] Yiran Guo, Zhongjian Qiao, Yingqi Xie, Jie Liu, Dan Ye, Ruiqing Zhang, Shuang Qiu, and Lijie Xu. Deep dense exploration for LLM reinforcement learning via pivot-driven resampling. *arXiv preprint arXiv:2602.14169*, 2026.
- [3] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *NeurIPS*, 2023.
- [4] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-DPO: Step-wise preference optimization for long-chain reasoning of LLMs. *arXiv preprint arXiv:2406.18629*, 2024.
- [5] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. *ICLR*, 2025. *arXiv:2409.12917*.
- [6] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [7] Aladin Djuhera, Farhan Ahmed, Swanand Ravindra Kadhe, Syed Zawad, Heiko Ludwig, and Holger Boche. When data is the algorithm: A systematic study and curation of preference optimization datasets. *arXiv preprint arXiv:2511.10985*, 2025.
- [8] Hamish Ivison, Yizhong Wang, Jiacheng Liu, Zeqiu Wu, Valentina Pyatkin, Nathan Lambert, Noah A. Smith, Yejin Choi, and Hannaneh Hajishirzi. Unpacking DPO and PPO: Disentangling best practices for learning from preference feedback. *arXiv preprint arXiv:2406.09279*, 2024.
- [9] Yu Pan, Zhongze Cai, Guanting Chen, Huaiyang Zhong, and Chonghuan Wang. What matters in data for DPO? *arXiv preprint arXiv:2508.18312*, 2025.
- [10] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.
- [11] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2024.

- [12] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.
- [13] 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. *arXiv preprint arXiv:2402.03300*, 2024.
- [14] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *ICLR*, 2024. arXiv:2310.01798.
- [15] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- [16] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *NeurIPS*, 2021.
- [17] 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. *SOSP*, 2023. arXiv:2309.06180.
- [18] Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.
- [19] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1):289–300, 1995.
- [20] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized language models. *NeurIPS*, 2023.
- [21] 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. *ICLR*, 2022. arXiv:2106.09685.
- [22] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. *arXiv preprint arXiv:2506.01939*, 2025.