

Beyond the Wrong Bit: Failure States and the Supervision Unit in Verifiable Reasoning

Zihan Ma^{1,2,4}, Zhuoyi Lin³, Minnan Luo^{1,2,4,†}, Xiaoxuan Guo⁵, Wenya Wang^{6,†}

¹School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, China

²National Engineering Research Center for Visual Information and Applications, China

³Institute of Advanced Intelligence and Computing (IAIC), A*STAR, Singapore

⁴Shaanxi Province Key Laboratory of Big Data Knowledge Engineering,
Xi'an Jiaotong University, Xi'an, 710049, China

⁵School of Information and Communication Engineering,
Communication University of China, Beijing, China

⁶College of Computing and Data Science, Nanyang Technological University, Singapore
mazihan880@stu.xjtu.edu.cn, minnluo@xjtu.edu.cn, wangwy@ntu.edu.sg

Abstract

Verifiable reasoning training compresses an entire response to a binary pass-or-fail label, yet a fail label alone does not reveal whether the corresponding trajectory should be discarded, penalized, or repaired. Many failed trajectories remain close to the correct solution through substantial portions of their intermediate reasoning and deviate only at a later branching point, whereas others contain no reliable prefix that can support a correct suffix. Treating both failure types as uniformly negative supervision can penalize useful reasoning states while diluting corrective updates for erroneous ones. To address this limitation, we propose PRIMA, which decomposes failed trajectories into inheritable prefixes and divergent suffixes, and rewrites repairable failures into shared-prefix preference pairs for localized reasoning optimization. We formalize the optimization cost of mixing the two failure types and show that treating them as a uniform negative class can shrink the effective gradient magnitude, motivating their separation before training. Experiments show that PRIMA converts failed trajectories from coarse-grained negative supervision into cleaner suffix-level supervision signals, yielding higher reported means in 15 of 16 model–task–training–paradigm comparisons against the strongest corresponding baselines and tying in the remaining comparison.

1 Introduction

Verifiable reasoning has emerged as an efficient training loop in recent years. Given a problem, models sample large trajectory pools, and an external verifier compresses each trajectory into a

[†]Indicates corresponding authors.

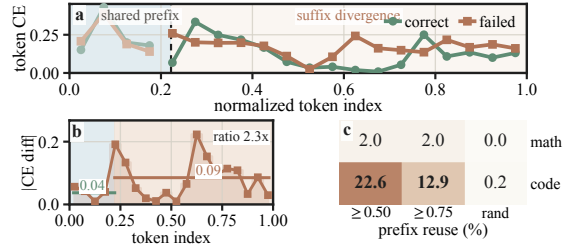


Figure 1: Motivating evidence that reasoning errors concentrate after a shared prefix. (a) Token-wise cross-entropy is similar in the early shared region. (b) The cross-entropy gap between correct and failed trajectories increases after the shared prefix. (c) Correct and failed trajectories for the same problem share more length-normalized prefix content than length-matched random controls.

binary terminal outcome. These verifier signals are then reused as search weights, rejection sampling thresholds, preference optimization labels, or process reward signals, supporting a range of methods from reinforcement learning to Direct Preference Optimization (DPO). Recent systems have shown that such external verifiable feedback can scale up reasoning training to large-scale trajectory pools (DeepSeek-AI, 2025; Guan et al., 2025).

Within this training pipeline, failed trajectories are handled through several distinct strategies. First, rejection sampling fine-tuning keeps only the passing samples and discards the failed ones. Second, preference optimization uses failed responses as negative samples, pairing them with passing responses and feeding them into the loss function of DPO and its variants (Rafailov et al., 2023). Third, reasoning repair methods, including self-refinement, multi-turn self-correction, and trajec-

tory rewriting, feed the failed trajectories back into the model in an attempt to recover the correct answer from them (Madaan et al., 2023; Shinn et al., 2023). Despite these differences in downstream usage, many terminal-label pipelines inherit the same upstream assumption: the verifier’s failure signal defines a semantically homogeneous negative class. In other words, trajectories that fail early due to fundamental reasoning errors are treated equivalently to those that fail only after substantial correct reasoning. Let a prefix denote the portion of a generated trajectory before a candidate divergence or repair point, and a suffix denote the generated sequence after that point. Existing pipelines usually do not ask whether the prefix of a failed trajectory is still a reusable reasoning context, or whether only the later suffix should be penalized or replaced.

This premise hardly holds in long-chain reasoning. Figure 1 presents three diagnostic observations on reasoning trajectories from multiple model families. Figure 1(a,b) shows that correct and failed trajectories share nearly identical token-wise cross-entropy within the early shared prefix, and that after the first textual divergence the gap widens sharply, with a suffix-region peak roughly $2.3\times$ that of the prefix region. Figure 1(c) further shows that correct and failed trajectories for the same problem share more length-normalized prefix content than length-matched random controls, reducing concern that the pattern is solely a length effect. Prefix support measures the fraction of the shorter trajectory that is shared before the first textual divergence. These observations suggest that errors are often localized after a shared prefix rather than being uniformly distributed across the entire trajectory. Yet sharing a prefix does not necessarily imply that the prefix remains reusable. Some prefixes stay close to a recoverable reasoning state and can serve as stable contexts for suffix replacement, while others have already departed from the successful-state neighborhood. A binary verifier alone cannot distinguish between them.

This exposes a structural limitation of existing verifiable reasoning pipelines: the verifier collapses failure states with fundamentally different optimization properties into the same negative label. The first class comprises *RC failures* (operationally repairable), which retain a prefix accepted by the state criterion. This prefix can be preserved as a common context for a correct suffix. The error lies mainly in the suffix, suggesting that training should

preserve the prefix while replacing the divergent suffix. The second class comprises *UC failures* (not selected as repairable), whose trajectories contain no prefix accepted by the state criterion and therefore do not enter shared-prefix suffix comparison. Although both trajectory types receive the same terminal failure label, they require different treatment during training. Mixing them within a single pool of negative samples can penalize useful prefixes and yield misaligned update directions.

In this paper, we propose **PRIMA (Prefix-Repair Identification for Minimum-pollution Alignment)**, an automated, verifier-compatible data-construction framework. Given a failed trajectory, PRIMA does not ask the verifier to reinterpret the failure, nor does it rely on surface completeness. Instead, it reads the prefix state from a frozen base model and decides whether the prefix can still be followed by a correct suffix based on three complementary signals: geometric residual, local branching response, and prefix support. In contrast, format-level cues such as answer extractability, code parsability, and termination status are logged for auditing but are not input features to the scorer or cut-point filter. Specifically, for each inheritable prefix, PRIMA retrieves a successful suffix from the same problem or from a trajectory with the same prefix, re-verifies the full response formed by prefix and suffix, and constructs a verified pair of correct and incorrect suffixes under the shared prefix. The supervision signal is thus confined to the divergent suffix region, and the trainer receives localized contrastive supervision instead of treating the whole failed response as a monolithic negative. Crucially, this construction produces an objective-agnostic supervision signal: the same verified prefix-suffix pairs can serve DPO, SFT, reranking, and downstream repair (formalized in Sections 2 and 3). The main contributions are:

- We formalize the optimization cost of mixing RC and UC failures, and provide evidence that treating them as a uniform negative class can lead to gradient contraction.
- We propose PRIMA, a data-centric framework that decomposes failed trajectories into inheritable prefixes and divergent suffixes, and constructs shared-prefix supervision signals compatible with various downstream objectives.
- Across model families, benchmarks, and training paradigms, PRIMA yields higher or tied reported mean scores than the strongest corresponding base-

lines in all 16 comparisons and provides a reusable, objective-agnostic data primitive for verifiable reasoning.

2 Disentangling Failure States and Their Optimization Cost

In this section, we start from frozen-model hidden states to formalize the heterogeneity among failed trajectories that a binary verifier masks, and we then quantify the optimization cost of mixing them. Some failures preserve useful prefixes until a late branching point, while others diverge early from successful states. Treating both as the same negative class can contract the mixed gradient when their update directions are misaligned. State diagnostics, Repair@4, downstream training, and held-out evaluation use the source pools and splits specified in their respective protocols (Section 3.2 and Appendix C.1). Complete responses are labeled pass or fail by the corresponding domain verifier.

2.1 Formalization and Empirical Evidence of Failure States

Format Boundaries Are Unreliable. Recovering reusable intermediate computation from failed trajectories first requires avoiding the mistake of treating format boundaries as reasoning states. A natural but unreliable heuristic would be to filter candidate prefixes by truncation, answer extractability, or program parsability. However, the positions at which these signals appear are usually later than the critical reasoning fork: the prefix of a truncated trajectory may already be close to the correct-solution manifold and may be continuable by a correct suffix, while a naturally ended and format-complete response may already have violated constraints in the middle. Prefix inheritability therefore cannot be reliably inferred from output form. We must return to the generation state of the frozen base model and check whether the prefix still preserves the condition needed to reach a correct suffix.

A natural first attempt is to filter failed prefixes by token-level average entropy, using entropy as a proxy for model uncertainty. However, entropy is only a scalar confidence signal and does not tell whether a prefix remains in a state from which a correct suffix is still reachable. For example, a model may continue with low entropy after it has already moved away from the successful-solution region, or it may have high entropy near a genuine reasoning

fork while the prefix is still repairable. The linear-discrimination AUC of entropy for the held-out calibration proxy labels is only 0.57. We therefore characterize candidate prefixes using three complementary quantities: a hidden-space geometric residual, a local branching response, and a discrete prefix-support score.

Prefix Representation and Whitening. Given a source model s , a problem x , and a failed trajectory $y^- = (y_1, \dots, y_T)$, let the candidate prefix at position t be

$$p_t = y_{\leq t}^-.$$

We run the frozen base model on (x, p_t) and denote by $\mathbf{h}_t$ the final-layer hidden state at the last token of the prefix. Thus, each candidate prefix is represented by the endpoint hidden state of that prefix, rather than by all token states inside the prefix.

Different source models have different hidden-state scales, anisotropy, and cross-dimensional correlations. To make geometric quantities comparable within each source model, we first normalize the hidden space of that source. Let $\mathcal{H}_s$ be the collection of final-layer endpoint hidden states extracted from candidate prefixes in the trajectory pool of source model s . We estimate the empirical mean and covariance of these hidden states as

$$\begin{aligned}\mu_s &= \mathbb{E}_{\mathbf{h} \in \mathcal{H}_s}[\mathbf{h}], \\ \Sigma_s &= \mathbb{E}_{\mathbf{h} \in \mathcal{H}_s}[(\mathbf{h} - \mu_s)(\mathbf{h} - \mu_s)^\top].\end{aligned}$$

We then construct the whitening operator

$$W_s = \Sigma_s^{-1/2},$$

which decorrelates and rescales hidden dimensions so that the source-level covariance becomes identity. The normalized prefix state is

$$\tilde{\mathbf{h}}_t = W_s(\mathbf{h}_t - \mu_s).$$

The hidden-state quantities G_t and F_t below are computed in this source-normalized hidden space. The symbolic quantity S_t is computed in token space and then length-normalized.

We first define the successful-state neighborhood used as the geometric reference. Let $\mathcal{D}_x^+$ denote the set of successful trajectories for the same problem x ; when too few same-problem successes are available, we augment it with successful trajectories from retrieval-neighbor problems in the same domain. For each trajectory in this set, we extract final-layer endpoint hidden states at candidate prefix positions and map them with the same

whitening operator W_s . To compare states at similar reasoning stages, we keep endpoints whose normalized positions are within a small window around the current position t/T . The resulting set is denoted by

$$\mathcal{N}_x^+(t) = \left\{ \tilde{\mathbf{h}}_j^+ : \frac{y^+ \in \mathcal{D}_x^+}{|j/|y^+|-t/T| \leq \delta} \right\},$$

where δ is the position-matching window. This neighborhood contains normalized endpoint hidden states from successful trajectories, not the entire sequence of token states inside each prefix.

State Quantities. The geometric residual G_t is defined as the nearest cosine distance from the current failed-prefix state to this successful-state neighborhood:

$$G_t = \min_{z \in \mathcal{N}_x^+(t)} \left(1 - \cos(\tilde{\mathbf{h}}_t, z) \right). \quad (1)$$

A smaller G_t means that the failed prefix remains close to the region visited by successful trajectories at a comparable generation stage. A larger G_t indicates that the prefix has likely moved away from the region where correct suffixes are observed.

Geometric proximity alone may miss abrupt local transitions. A prefix can remain globally close to the successful neighborhood but still undergo a local reasoning fork. To capture such local dynamics, we define a second-order hidden-state change along the normalized prefix-state sequence. Because PRIMA constructs data offline from fixed complete trajectories, this centered diagnostic is evaluated only at interior positions $2 \leq i \leq T-1$:

$$\mathcal{L}\tilde{\mathbf{h}}_i = \tilde{\mathbf{h}}_i - \frac{\tilde{\mathbf{h}}_{i-1} + \tilde{\mathbf{h}}_{i+1}}{2}.$$

We then accumulate recent second-order changes with an exponential decay:

$$F_t = \sum_{i=2}^{\min\{t, T-1\}} \exp[-\eta(t-i)] \left\| \mathcal{L}\tilde{\mathbf{h}}_i \right\|_2, \quad (2)$$

where $\eta > 0$ controls how quickly earlier changes are discounted; endpoint values are omitted, and an empty sum is zero. A large F_t indicates that a local branching event has recently occurred near the candidate prefix.

Finally, hidden-state continuity is not sufficient if the text has already lost key problem objects, constraints, or program structure. We therefore add a discrete prefix-support score. Recall that $\mathcal{D}_x^+$ denotes the set of successful trajectories for the same

problem x , augmented with retrieval-neighbor successes when same-problem successes are too few (Section 2.1). For a failed trajectory y^- and a successful trajectory $y^+ \in \mathcal{D}_x^+$, let $k(y^+, y^-)$ be the first textual divergence position between them. The length-normalized prefix support is

$$S_t = \max_{y^+ \in \mathcal{D}_x^+} \frac{\min\{t, k(y^+, y^-) - 1\}}{\min\{|y^+|, |y^-|\}}. \quad (3)$$

This score measures how much same-problem successful and failed trajectories share before their first textual divergence, normalized by the shorter trajectory length. Thus, S_t complements the hidden-space quantities G_t and F_t with a symbolic check of whether the candidate prefix still preserves shared problem structure.

Repairability Scorer and Cut-Point Selection.

We normalize and combine the three metrics above into a repairability scorer. Its calibration targets are automatically derived operational proxies independent of G_t, F_t, S_t : wrong-answer failures are the RC proxy for code, while compilation, runtime, and time-limit failures are the UC proxy; for mathematics, a finite but incorrect answer is the RC proxy and a length-capped output is the UC proxy. These proxies supervise calibration but do not define closure. Let $\phi(y^-; t) = (G_t, F_t, S_t)$. Within each source model, we apply quantile normalization and obtain $\bar{G}_t, \bar{F}_t, \bar{S}_t \in [0, 1]$. The repairability scorer is

$$r_\lambda(\phi(y^-; t)) = \sigma(\lambda_0 - \lambda_G \bar{G}_t - \lambda_F \bar{F}_t + \lambda_S \bar{S}_t), \quad (4)$$

where $\lambda_G, \lambda_F, \lambda_S \geq 0$, and σ is the sigmoid function. The signs encode the directional constraints of repairability. The weights are estimated by class-balanced logistic regression on source-balanced, problem-blocked calibration splits (Appendix C.2). We define the state-closure variable as

$$C_\lambda(y^-, t) = \mathbb{1}\{r_\lambda(\phi(y^-; t)) \geq \lambda_c\}, \quad (5)$$

where λ_c is selected on the calibration split and never on evaluation problems.

$C_\lambda(y^-, t) = 1$ means that the candidate prefix p_t passes the state check and can serve as the common context for comparing positive and negative suffixes under the same prefix. $C_\lambda = 1$ alone is still insufficient to determine the cut point: a cut that is too early loses inheritable computation, while a cut that is too late may cross a local branch. Since

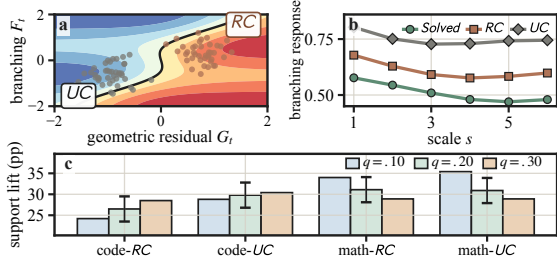


Figure 2: (a) Prefixes projected onto geometric residual G_t and branching F_t under the operational RC/UC partition. (b) Branching response across six probing scales: UC is highest, Solved is lowest, RC lies between. (c) Prefix-support lift for code and math, split by failure type across the three displayed empirical quantile settings.

C_λ already covers the geometric residual, we define a cut-point filter that adds only the structural constraints:

$$b_\lambda(y^-, t) = \mathbb{1}\{\bar{S}_t \geq s_c, \bar{F}_t \leq f_c, t < T\}, \quad (6)$$

where s_c and f_c are the lower bound on prefix support and the upper bound on local branching determined on the calibration set. The final retained position is the latest legal cut point:

$$\tau_\lambda(y^-) = \max\{t < T : C_\lambda(y^-, t) = 1, b_\lambda(y^-, t) = 1\}, \quad (7)$$

and if the legal set is empty, we set $\tau_\lambda(y^-) = \perp$. Operationally, $\tau_\lambda(y^-) \neq \perp$ means that at least one candidate prefix falls in the selected region and retains sufficient symbolic support. Section 3.2 evaluates this interpretation with Repair@M. Format-audit signals do not participate in the state decision and are recorded only offline. The failed pool is then partitioned into selected failures RC and rejected failures UC:

$$\begin{aligned} \text{RC} &= \{(x, y^-, \tau_\lambda(y^-)) \mid V(x, y^-) = 0, \\ &\quad \tau_\lambda(y^-) \neq \perp\}, \\ \text{UC} &= \{(x, y^-) \mid V(x, y^-) = 0, \\ &\quad \tau_\lambda(y^-) = \perp\}. \end{aligned} \quad (8)$$

RC and UC are disjoint and together exhaust the failed pool.

Empirical Evidence. Figure 2 projects candidate prefixes onto G_t , F_t , and S_t and shows descriptive separation under the operational RC/UC partition. Figure 3 further reports hidden-state trajectories and the distribution of G_t , showing that divergence

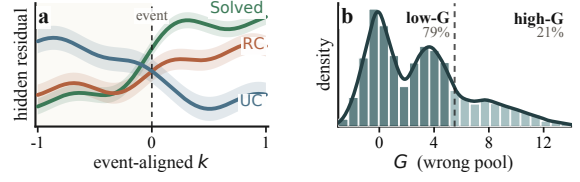


Figure 3: (a) Hidden residual aligned to the local branching event ($k = 0$); the pattern suggests that separation precedes terminal verification. (b) Under per-sample maximum-component assignment, 79% of samples belong to the low- G component and 21% to the high- G component.

appears before terminal verification. Against held-out verifier-derived proxy labels, length-stratified AUCs of G_t and F_t are 0.84 and 0.81, reducing concern that trajectory length alone drives the result. Because every trajectory in this analysis already has the same terminal fail label, the independent fixed-prefix Repair@4 test in Section 3.2 provides the direct validation of continuability.

To observe the dynamic process of state evolution, we define the diagnostic event position $\tau_{\text{evt}}(y^-) = \arg \max_{2 \leq t < T} \|\mathcal{L}\tilde{h}_t\|_2$ to align state curves. For any scalar reading m_t , the event-aligned curve is $E_m(k, s) = \mathbb{E}[m_t(y) \mid t - \tau_{\text{evt}}(y) = k, s(y) = s]$, where $s \in \{\text{Solved}, \text{RC}, \text{UC}\}$. Figure 3(a) shows that RC stays close to the Solved manifold around the branching event, while UC rapidly deviates after the event. This pattern suggests that separation emerges before terminal verification.

The geometric residual G_t around candidate cut points is bimodal (Figure 3(b)). Integrating density mass on either side of the valley gives approximately 84%/16%; this differs from the 79%/21% hard component assignment because samples overlap near the valley. Neither mixture summary defines the RC/UC partition, which also uses F_t and S_t . Format-feature splits mix the two residual modes, whereas the C_λ split aligns with the event curves. The independent Repair@4 test in Section 3.2 then evaluates whether selected prefixes are more continuable.

2.2 Optimization Cost of State Mixing

Prefix-state differentiation becomes a data-construction problem because it changes training gradients. Let D_{RC} and D_{UC} be the empirical distributions of the two sample types, and let $D_{\text{mix}} = \alpha D_{\text{UC}} + (1 - \alpha) D_{\text{RC}}$ be the mixed negative-sample distribution, where $\alpha \in [0, 1]$ is

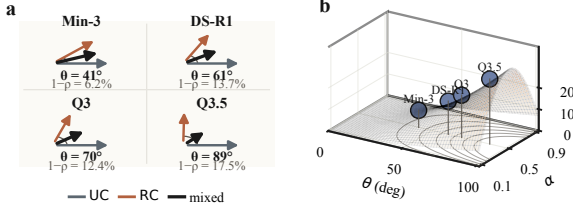


Figure 4: (a) Gradient geometry for four source models: angular separation θ between UC and RC directions ranges from 41° to 89° , while $1-\rho$ ranges from 6.2% to 17.5%. (b) Theoretical $1-\rho$ surface over θ and mixing weight α , with all four measured points overlaid.

the UC proportion. For any differentiable loss ℓ_θ , the state-conditional and mixed gradients are

$$\begin{aligned}
 g_s &= \nabla_\theta \mathbb{E}_{(x,y) \sim D_s} \ell_\theta(x,y), \quad s \in \{\text{RC}, \text{UC}\}, \\
 g_{\text{mix}} &= \alpha g_{\text{UC}} + (1-\alpha) g_{\text{RC}}, \\
 \rho &= \frac{\|g_{\text{mix}}\|}{\alpha \|g_{\text{UC}}\| + (1-\alpha) \|g_{\text{RC}}\|} \in [0, 1].
 \end{aligned} \tag{9}$$

Assuming $\alpha \|g_{\text{UC}}\| + (1-\alpha) \|g_{\text{RC}}\| > 0$, $\rho \in [0, 1]$ measures first-order gradient-norm contraction; it is not the realized step size of an adaptive optimizer. For a non-degenerate mixture with both weighted components nonzero, strict contraction occurs whenever the two gradients are not positively aligned. Under the illustrative setting $\alpha = 1/2$ and $\|g_{\text{RC}}\| = \|g_{\text{UC}}\|$, $\rho = \sqrt{(1 + \cos \theta_{\text{RCUC}})/2}$. Angles of 60° , 90° , and 105° yield norm losses of about 13%, 29%, and 39%, respectively. The empirical $\alpha \approx 0.16$ is computed directly from the τ_λ -based RC/UC partition and is not estimated from either mixture summary in Figure 3(b).

Across four reasoning models, class-mean negative-log-likelihood gradients on the last two Transformer blocks have RC/UC angles from 41° to 89° (Figure 4(a)), with measured contraction from 6.2% to 17.5%. The pointwise mean absolute deviation from the equal-weight, equal-norm geometric reference is 0.043 (Appendix B.2). This result characterizes first-order gradient-norm contraction; its downstream relevance is evaluated empirically rather than guaranteed by the bound. The observed contraction motivates state separation as a practical way to reduce gradient-norm cancellation caused by update misalignment.

2.3 Separating Prefix Preservation from Training Signals

Once the state-closure criterion partitions the failed-trajectory pool, the prefix p_τ of an RC failure can

be preserved and isolated from the negative suffix signal. Given $(x, y^-, \tau_\lambda(y^-)) \in \text{RC}$, write $y^- = (p_\tau, u^-)$, where u^- is the original erroneous suffix. If a concatenable suffix u^+ is retrieved from successful trajectories and the full response passes re-verification, we construct a preference unit (x, p_τ, u^+, u^-) with a shared prefix. The supervision is therefore restricted to the suffix comparison.

The autoregressive factorization decomposes sequence log probability into the sum of a prefix term and a suffix-conditional term:

$$\log \pi_\theta(y \mid x) = \log \pi_\theta(p_\tau \mid x) + \log \pi_\theta(u \mid x, p_\tau).$$

In preference learning, such as DPO, the positive and negative samples share the same prefix. Therefore, the explicit prefix log-probability terms cancel in the preference difference, leaving the suffix-conditional log-ratio

$$\log \frac{\pi_\theta(u^+ \mid x, p_\tau)}{\pi_{\text{ref}}(u^+ \mid x, p_\tau)} - \log \frac{\pi_\theta(u^- \mid x, p_\tau)}{\pi_{\text{ref}}(u^- \mid x, p_\tau)}.$$

This removes an explicit negative loss on prefix tokens and concentrates the contrast on the two suffixes, matching the localized supervision unit defined above.

For SFT, the repaired trajectory (x, p_τ, u^+) mines additional verified positive examples from failed data, preserving only the inheritable prefix and replacing the erroneous suffix without polluting likelihood training.

3 PRIMA: A Data-Centric Framework

3.1 The Same-Prefix Data Unit

Section 2 decomposes terminal failure into inheritable and non-inheritable states. This section translates that state distinction into data construction. Figure 5 gives an end-to-end view of the resulting workflow. PRIMA does not re-score full responses; it rewrites training units. It first checks whether a failed trajectory contains an inheritable prefix, then compresses the supervision unit into suffix-level differences under the same prefix. Only when the repaired full response passes the original verifier does the failed sample enter subsequent training.

The experimental boundary is established in Section 2. For each full-size model in the main comparison, candidate trajectories and replacement suffixes come only from that model’s fixed self-distilled pool of four trajectories per problem. SFT

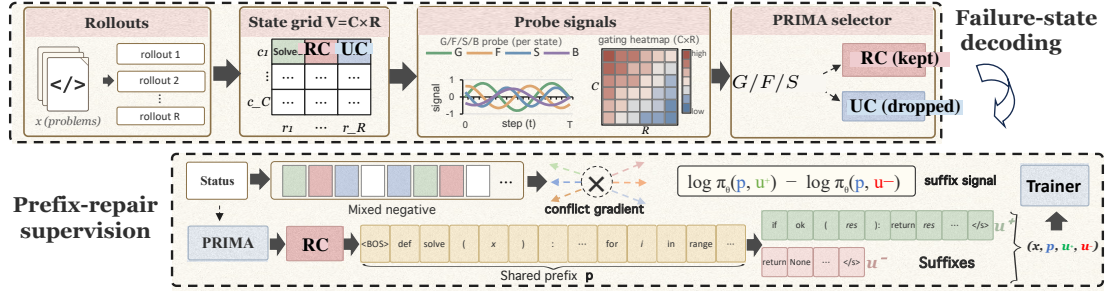


Figure 5: Failure-state decoding: Rollouts are organized into a $C \times R$ state grid; G, F, S enter the PRIMA selector, while B is logged only for format auditing. The selector partitions failures into RC (kept) and UC (dropped). Prefix-repair supervision: Shared prefixes from RC trajectories are paired with a correct suffix u^+ and a failed suffix u^- , forming contrastive training units $\{x, p, u^+, u^-\}$ whose preference term is conditional on the shared prefix.

Table 1: Fixed-prefix continuability diagnostic, Repair@4 (%).

Prefix Source	Qwen3-8B	Qwen3.5-9B	DS-R1-Llama-8B	Ministral-3-8B
Oracle correct prefix	43.2	51.8	41.5	44.0
PRIMA accepted RC	13.8	16.7	13.0	14.7
PRIMA rejected UC	5.6	7.2	5.9	5.8
$\Delta(\text{RC} - \text{UC})$	+8.2	+9.5	+7.1	+8.9

baselines filter positives; DPO baselines construct pairs; the feature selector and text reranker do not access additional trajectories. No stronger model or external teacher supplies data for these main comparisons. Training details appear in Appendix C.

For a failed trajectory that passes the repairability check in Section 2.1, $(x, y^-, \tau_\lambda(y^-))$, write

$$y^- = p_\tau \oplus u^-, \quad p_\tau = y^-_{\leq \tau_\lambda(y^-)}.$$

Here p_τ is the latest legal prefix accepted by the state discriminator, and u^- is the failed suffix after that cut point. PRIMA excludes prefixes that fail the state check and retrieves concatenable replacements from successful trajectories generated by the same source model. Donors come from the same problem or from retrieval-neighbor problems selected with the problem-representation retrieval used to construct the successful neighborhood. Only suffixes that share the required discrete prefix, can be concatenated, and pass full-response re-verification are accepted:

$$\mathcal{V}(x, p_\tau \oplus u^+) = 1, \quad \mathcal{V}(x, p_\tau \oplus u^-) = 0. \quad (10)$$

These constraints yield the same-prefix data unit set

$$\begin{aligned} \mathcal{Z}_\lambda = \{ & (x, p_\tau, u^+, u^-) \mid \tau_\lambda(y^-) \neq \perp, \\ & \mathcal{V}(x, p_\tau \oplus u^+) = 1, \\ & \mathcal{V}(x, p_\tau \oplus u^-) = 0 \}. \end{aligned} \quad (11)$$

Let $\mathcal{D}_{\text{clean}}^+$ denote the set of all verifier-passing positive examples from the shared candidate pool. For SFT, PRIMA rewrites repairable failures into additional verified positive examples

$$\mathcal{D}_\lambda^{\text{rep}} = \{(x, p_\tau \oplus u^+) \mid (x, p_\tau, u^+, u^-) \in \mathcal{Z}_\lambda\}, \quad (12)$$

and constructs the final PRIMA-SFT training set

$$\mathcal{D}_{\text{PRIMA}}^{\text{SFT}} = \mathcal{D}_{\text{clean}}^+ \cup \mathcal{D}_\lambda^{\text{rep}}. \quad (13)$$

For DPO, PRIMA preserves correct-failed suffix pairs under the same prefix:

$$\mathcal{D}_{\text{PRIMA}}^{\text{DPO}} = \mathcal{Z}_\lambda. \quad (14)$$

The two sets are trained with the standard autoregressive and standard DPO objectives, respectively. Since the positive and negative responses share the same prefix,

$$\begin{aligned} \log \pi(p_\tau \oplus u \mid x) &= \log \pi(p_\tau \mid x) \\ &+ \log \pi(u \mid x, p_\tau). \end{aligned} \quad (15)$$

The common prefix cancels in the DPO contrastive term, concentrating the parameter update on the suffix difference rather than penalizing the inheritable prefix alongside the original erroneous suffix.

Figure 5 illustrates this workflow. Code and math tasks differ at the terminal verifier. The code verifier executes the complete program against test cases, whereas the math verifier applies the task’s final-answer check. For mathematics, passing this

terminal check does not certify every intermediate reasoning step. Both domains use the same data unit: preserve the prefix, replace the suffix, and re-verify the full response.

3.2 Fixed-Prefix Continuability Diagnostic

PRIMA’s key premise is that accepted failure prefixes indeed preserve a state from which correct solutions can still be continued. To test this premise directly, we fix the prefix, resample only the suffix, and measure how often a correct completion can be reached. We formalize this as $\text{Repair}@M$: given a problem x and a fixed prefix p , we sample M suffixes

$$u_1, \dots, u_M \sim \pi(\cdot \mid x, p),$$

and check each concatenated response with the original task verifier. If at least one suffix passes verification, the prefix is scored as repairable:

$$\text{Repair}@M = \Pr[\exists m \leq M, V(x, p \oplus u_m) = 1].$$

We set $M = 4$ and report $\text{Repair}@4$ pooled across LiveCodeBench and OlympiadBench¹. This diagnostic fixes the prefix, resamples only suffixes, and applies the original verifier; it is run on the four 8B-class models with complete probing assets. Fixing the prefix isolates continuation success conditional on that prefix, so the group difference provides operational evidence of differential continuability.

All results in this section are computed on the evaluable repair subset: after fixing the prefix, suffixes can be resampled and fully judged by the original task verifier. A small number of samples for which judgment cannot be completed because of excessive length or incomplete original parsing are excluded. This treatment does not alter the definition of repairability; it only removes cases that are technically infeasible to execute.

Table 1 compares three prefix sources. Oracle prefixes come from same-problem correct solutions. RC and UC both come from same-problem failed trajectories but are respectively accepted and rejected by PRIMA. Both use the same evaluable subset and failure origin.

All four base models yield a consistent ordering,

$$\text{Oracle} > \text{RC} > \text{UC}.$$

RC versus UC provides the direct comparison. PRIMA-selected prefixes have higher $\text{Repair}@4$

¹OlympiadBench (large-scale, basic-to-competition difficulty) is used for repair diagnostics

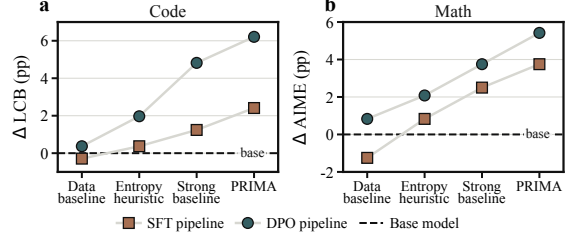


Figure 6: Downstream pass@1 improvement over non-fine-tuned models, averaged across the four full-size backbones (three-run mean). (a) LiveCodeBench; (b) AIME.

on every model, with an average gain of 8.42 percentage points and per-model gains from 7.1 to 9.5 points. The mean $\text{Repair}@4$ is 14.55 for RC and 45.13 for Oracle, indicating that selected failed prefixes remain less continuable than prefixes from native correct solutions.

3.3 Downstream Verification

To evaluate PRIMA across model families and reasoning tasks, we report downstream pass@1 improvement over non-fine-tuned models on LiveCodeBench v5/v6 and AIME 2024/2025. Each AIME year contains 30 problems; we score the years separately and report their equally weighted mean. Figure 6 gives three-run means across four full-size backbones, with per-model results in Table 7.

PRIMA maintains an advantage under both SFT and DPO training paradigms. PRIMA-SFT matches or exceeds the strongest non-PRIMA SFT baseline (quality-filtered positives) on every model-task pair, delivering relative pass@1 gains of 3.3% on the four-backbone LCB mean and 2.0% on the four-backbone AIME mean; the per-model breakdown is in Table 7. Because every method shares the same 4-trajectory candidate pool, this comparison controls for candidate-trajectory availability; PRIMA differs in how selected failures are rewritten into verifier-rechecked positive examples.

PRIMA-DPO has the highest reported mean among random pairing, shared-prefix-length pairing, entropy sorting, and both learning-based full-response scorers on every base model. The text reranker relies solely on raw text, requiring no hidden-state parsing or hand-crafted features. On LCB its mean exceeds all non-learning filtering rules; on AIME it is competitive with shared-prefix-length pairing and entropy sorting. Against the strongest non-PRIMA DPO baseline (the struc-

tured feature selector), PRIMA-DPO improves the four-backbone pass@1 mean by 3.5% on LCB and 2.6% on AIME in relative terms; the per-model breakdown is in Table 7. Across the 16 PRIMA-versus-strongest-baseline cells, PRIMA records 15 higher means and one tie; Appendix C.6 reports mean $\pm$ standard deviation. Individual AIME cells remain low-resolution because each year has only 30 problems. The fixed-pool design holds initial candidate-trajectory availability constant across methods.

4 Related Work

PRIMA relates to three strands of work: verifiable-reward and preference training (DeepSeek-AI, 2025; Kimi Team, 2025; Guan et al., 2025; Luo et al., 2025; Yuan et al., 2023; Rafailov et al., 2023), process rewards and self-correction (Lightman et al., 2024; Madaan et al., 2023; Shinn et al., 2023), and failure-state analysis. Unlike step-level reward models or refinement loops, PRIMA uses frozen-model states and no additional LLM judge. Appendix A gives an extended discussion.

5 Conclusion

We reframe failure in verifiable reasoning through prefix inheritance and characterize the gradient contraction that can arise when RC and UC failures are mixed. PRIMA separates selected prefixes before training and produces objective-agnostic, shared-prefix supervision signals. Its 15 higher reported means and one tie across the 16 model-task-training-paradigm comparisons support this data construction as a reusable primitive.

Acknowledgments

This work is supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (No. JYB2025XDXM101), the National Natural Science Foundation of China (No. 62272374), the Natural Science Foundation of Shaanxi Province (No. 2024JC-JCQN-62), the State Key Laboratory of Communication Content Cognition under Grant No. A202502, and the Key Research and Development Project in Shaanxi Province (No. 2023GXLH-024). This research is also supported by the MOE AcRF Tier 1 Seed Grant (RS37/24, #025041-00001), Singapore.

6 Limitations

We study two representative verifiable domains, code and mathematics, with four open-weight backbones. Evidence is limited to offline construction from fixed rollout pools; online GRPO/RLVR training with continuously refreshed trajectories is not evaluated. The operational proxy labels and calibrated thresholds may not transfer unchanged to open-ended, multilingual, or proprietary settings, especially when hidden states are unavailable. Repair@4 measures empirical continuability rather than proving semantic correctness of a prefix. Moreover, the mathematics verifier checks the terminal answer and does not certify every intermediate step. Individual AIME comparisons are coarse because each annual set contains only 30 problems.

7 Ethics Statement

Data and Artifacts. All datasets used in this work are publicly available. No new dataset involving external human participants was collected. Automatic verifier-derived labels are used for scorer fitting and threshold selection. Separately, three authors audited 240 stratified failed trajectories; those judgments were used only to assess agreement with the automatic proxy labels and did not enter fitting, threshold selection, training, or evaluation.

Potential Risks. This work improves the data efficiency of verifiable reasoning training by reusing information from failed trajectories. While reasoning models may be misused to generate harmful or deceptive content, our contribution targets training methodology rather than deployment and does not directly target capability expansion. Reusing failed data may reduce some computational waste in data construction.

AI Assistance. During the preparation of this manuscript, AI-based tools were used solely for language polishing and grammar checking. Some code snippets in the experimental pipeline were written with the assistance of AI-based code completion tools. All AI-generated text and code were thoroughly reviewed and verified by the authors, who take full responsibility for the correctness and integrity of the final manuscript and implementation.

References

- Thomas Anthony, Zheng Tian, and David Barber. 2017. [Thinking fast and slow with deep learning and tree search](#). *Preprint*, arXiv:1705.08439.
- Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. [A general theoretical paradigm to understand learning from human preferences](#). In *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, Proceedings of Machine Learning Research, pages 4447–4455. PMLR.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. [Constitutional AI: harmlessness from AI feedback](#). *CoRR*, abs/2212.08073.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. [Graph of thoughts: Solving elaborate problems with large language models](#). In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pages 17682–17690. AAAI Press.
- Hao Chen, Jindong Wang, Ankit Shah, Ran Tao, Hongxin Wei, Xing Xie, Masashi Sugiyama, and Bhiksha Raj. 2024a. [Understanding and mitigating the label noise in pre-training on downstream tasks](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024b. [Self-play fine-tuning converts weak language models to strong language models](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 6621–6642. PMLR / OpenReview.net.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024c. [Self-play fine-tuning converts weak language models to strong language models](#). *Preprint*, arXiv:2401.01335.
- 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. 2021a. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- 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. 2021b. [Training verifiers to solve math word problems](#). *CoRR*, abs/2110.14168.
- Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. [Stepwise verification and remediation of student reasoning errors with large language model tutors](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.
- Quy-Anh Dang and Chris Ngo. 2026. [Reinforcement learning for reasoning in small llms: What works and what doesn't](#). *Preprint*, arXiv:2503.16219.
- DeepSeek-AI. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. [RAFT: reward ranked finetuning for generative foundation model alignment](#). *Trans. Mach. Learn. Res.*, 2023.
- Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. [RLHF workflow: From reward modeling to online RLHF](#). *Trans. Mach. Learn. Res.*, 2024.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. [KTO: model alignment as prospect theoretic optimization](#). *CoRR*, abs/2402.01306.
- Leo Gao, John Schulman, and Jacob Hilton. 2023. [Scaling laws for reward model overoptimization](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, Proceedings of Machine Learning Research, pages 10835–10866. PMLR.
- Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. 2024. [CRITIC: large language models can self-correct with tool-interactive critiquing](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. [rstar-math: Small llms can master math reasoning with self-evolved deep thinking](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

- 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. 2024. [Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems](#). *Preprint*, arXiv:2402.14008.
- Alex Heyman and Joel Zylberberg. 2025. [Reasoning large language model errors arise from hallucinating critical problem features](#). *CoRR*, abs/2505.12151.
- Jiwoo Hong, Noah Lee, and James Thorne. 2024. [ORPO: monolithic preference optimization without reference model](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 11170–11189. Association for Computational Linguistics.
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025. [Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model](#). *Preprint*, arXiv:2503.24290.
- Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2024. [Large language models cannot self-correct reasoning yet](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. [Live-codebench: Holistic and contamination free evaluation of large language models for code](#). *Preprint*, arXiv:2403.07974.
- Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xingyu Chen, Xiaoyuan Liu, Zhijie Wang, Junying Chen, Benyou Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. [The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models](#). *CoRR*, abs/2503.02875.
- Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. [Process reward models that think](#). *arXiv preprint arXiv:2504.16828*.
- Kimi Team. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.
- 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. 2024. [Training language models to self-correct via reinforcement learning](#). *Preprint*, arXiv:2409.12917.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.
- Li-Cheng Lan, Andrew Bai, Minhao Cheng, Chojui Hsieh, and Tianyi Zhou. 2025. [Exploring expert failures improves LLM agent tuning](#). *CoRR*, abs/2504.13145.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, and 101 others. 2026. [Ministral 3](#). *Preprint*, arXiv:2601.08584.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang 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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.
- 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. 2024. [Improve mathematical reasoning in language models by automated process supervision](#). *Preprint*, arXiv:2406.06592.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. [Self-refine: Iterative refinement with self-feedback](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. 2021. [What matters in learning from offline human demonstrations for robot manipulation](#). *Preprint*, arXiv:2108.03298.

- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. [Simpo: Simple preference optimization with a reference-free reward](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2021. [Webgpt: Browser-assisted question-answering with human feedback](#). *CoRR*, abs/2112.09332.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. [Disentangling length from quality in direct preference optimization](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024, Findings of ACL*, pages 4998–5017. Association for Computational Linguistics.
- Guilherme Penedo, Anton Lozhkov, Hynek Křdliček, Loubna Ben Allal, Edward Beeching, Agustín Piqueres Lajarín, Quentin Gallouédec, Nathan Habib, Lewis Tunstall, and Leandro von Werra. 2025. Codeforces. <https://huggingface.co/datasets/open-r1/codeforces>.
- Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. 2024. [Recursive introspection: Teaching language model agents how to self-improve](#). *Preprint*, arXiv:2407.18219.
- Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. 2024. [From \$r\$ to \$q^*\$: Your language model is secretly a \$q\$ -function](#). *CoRR*, abs/2404.12358.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacrose, Ahmed Awadallah, and Tengyang Xie. 2024. [Direct nash optimization: Teaching language models to self-improve with general preferences](#). *CoRR*, abs/2404.03715.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *CoRR*, abs/1707.06347.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Yang Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Yexuan Shi, Mingyu Wang, Yunxiang Cao, Hongjie Lai, Junjian Lan, Xin Han, Yu Wang, Jie Geng, Zhenan Li, Zihao Xia, Xiang Chen, Chen Li, Jian Xu, Wenbo Duan, and Yuanshuo Zhu. 2025. [Aime: Towards fully-autonomous multi-agent framework](#). *Preprint*, arXiv:2507.11988.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflection: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross J. Anderson, and Yarin Gal. 2024. [AI models collapse when trained on recursively generated data](#). *Nat.*, 631(8022):755–759.
- Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, and 22 others. 2024a. [Beyond human data: Scaling self-training for problem-solving with language models](#). *Preprint*, arXiv:2312.06585.
- Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, and 22 others. 2024b. [Beyond human data: Scaling self-training for problem-solving with language models](#). *Trans. Mach. Learn. Res.*, 2024.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. [Scaling LLM test-time compute optimally can be more effective than scaling model parameters](#). *CoRR*, abs/2408.03314.
- Yutao Sun, Mingshuai Chen, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, and Jianwei Yin. 2025. [The self-improvement paradox: Can language models bootstrap reasoning capabilities without external scaffolding?](#) In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July*

- 27 - August 1, 2025, Findings of ACL, pages 6501–6512. Association for Computational Linguistics.
- Qwen Team. 2026. [Qwen3.5: Accelerating productivity with native multimodal agents](#).
- Herun Wan, Minnan Luo, Zhixiong Su, Guang Dai, and Xiang Zhao. 2025a. [On the risk of evidence pollution for malicious social text detection in the era of LLMs](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9731–9761, Vienna, Austria. Association for Computational Linguistics.
- Herun Wan, Jiaying Wu, Minnan Luo, Zhi Zeng, and Zhixiong Su. 2025b. [Truth over tricks: Measuring and mitigating shortcut learning in misinformation detection](#). *Preprint*, arXiv:2506.02350.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. [Math-shepherd: Verify and reinforce llms step-by-step without human annotations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 9426–9439. Association for Computational Linguistics.
- 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, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. 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 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. 2025a. [Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms](#). *CoRR*, abs/2506.14245.
- Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. 2025b. [Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms](#). *Preprint*, arXiv:2506.14245.
- Mengzhou Xia, Sathika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. 2024. [LESS: selecting influential data for targeted instruction tuning](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 54104–54132. PMLR / OpenReview.net.
- Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy Liang, Quoc V. Le, Tengyu Ma, and Adams Wei Yu. 2023a. [Doremi: Optimizing data mixtures speeds up language model pretraining](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Qizhe Xie. 2023b. [Self-evaluation guided beam search for reasoning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. 2024. [Iterative preference learning from human feedback: Bridging theory and practice for RLHF under kl-constraint](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 54715–54754. PMLR / OpenReview.net.
- Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. 2025a. [A minimalist approach to LLM reasoning: from rejection sampling to reinforce](#). *CoRR*, abs/2504.11343.
- Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and 1 others. 2025b. [A minimalist approach to llm reasoning: from rejection sampling to reinforce](#). *arXiv preprint arXiv:2504.11343*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Fei Yu, Anningzhe Gao, and Benyou Wang. 2024. [Ovm, outcome-supervised value models for planning in mathematical reasoning](#). In *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, Findings of ACL, pages 858–875. Association for Computational Linguistics.

- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2025. [Self-rewarding language models](#). *Preprint*, arXiv:2401.10020.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. [Scaling relationship on learning mathematical reasoning with large language models](#). *CoRR*, abs/2308.01825.
- Eric Zelikman, Georges Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah D. Goodman. 2024. [Quiet-star: Language models can teach themselves to think before speaking](#). *CoRR*, abs/2403.09629.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. [Star: Bootstrapping reasoning with reasoning](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. [Rest-mcts*: LLM self-training via process reward guided tree search](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.
- Zhi Zhang, Jiayi Shen, Congfeng Cao, Gaole Dai, Shiji Zhou, Qizhe Zhang, Shanghang Zhang, and Ekaterina Shutova. 2024b. Proactive gradient conflict mitigation in multi-task learning: A sparse training perspective. *arXiv preprint arXiv:2411.18615*.
- Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. [Processbench: Identifying process errors in mathematical reasoning](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 1009–1024. Association for Computational Linguistics.
- Wei Zhu, Jian Zhang, Lixing Yu, Kun Yue, and Zhiwen Tang. 2026. [Dissecting failure dynamics in large language model reasoning](#). *CoRR*, abs/2604.14528.

A Extended Related Work

Training with Verifiable Rewards and Preference Optimization Reinforcement learning with verifiable rewards (RLVR) trains LLMs by sampling trajectories and compressing each into a binary pass/fail outcome via an external verifier. DeepSeek-R1 (DeepSeek-AI, 2025) used GRPO (Shao et al., 2024) on top of RL training (Ouyang et al., 2022; Schulman et al., 2017) to elicit chain-of-thought reasoning (Wei et al., 2022; Yao et al., 2023; Zelikman et al., 2024, 2022; Besta et al., 2024; Wang et al., 2023); Kimi (Kimi Team, 2025) and rStar-Math (Guan et al., 2025; Luo et al., 2025) extended this with diverse rewards and self-evolved deep thinking (Chen et al., 2024b; Yuan et al., 2025; Anthony et al., 2017). Open-source studies (Hu et al., 2025; Xiong et al., 2025a; Singh et al., 2024b) and theoretical analyses (Dang and Ngo, 2026; Kumar et al., 2024; Wen et al., 2025a) further characterize this training regime. On the data side, rejection sampling fine-tuning (RFT) (Yuan et al., 2023; Dong et al., 2023; Nakano et al., 2021; Dong et al., 2024) discards failed samples, keeping only passing ones. Preference and binary-feedback objectives include DPO (Rafailov et al., 2023; Rosset et al., 2024; Xiong et al., 2024; Rafailov et al., 2024), SimPO (Meng et al., 2024), KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024), IPO (Azar et al., 2024), and length-disentangled methods (Park et al., 2024). A property central to our work is that when positive and negative samples share a common prefix, the prefix cancels in DPO’s contrastive log-ratio, concentrating the comparison on the divergent suffix. Related self-training and RLVR studies examine self-play, scaled self-training, and minimalist reinforcement-learning recipes (Chen et al., 2024c; Singh et al., 2024a; Xiong et al., 2025b; Wen et al., 2025b). At inference time, self-evaluation-guided search, outcome-value planning, and test-time-compute allocation provide complementary ways to improve reasoning (Xie et al., 2023b; Yu et al., 2024; Snell et al., 2024).

Fine-Grained Failure Signals and State-Aware Data Beyond training objectives, several lines of work exploit failure structure. Self-Refine (Madaan et al., 2023), Reflexion (Shinn et al., 2023), CRITIC (Gou et al., 2024), and RISE (Qu et al., 2024) feed failures back to the model for correction, but LLMs can struggle to self-correct reliably without external feedback (Sun et al.,

2025; Huang et al., 2024). Process reward models (PRMs) (Lightman et al., 2024) score intermediate steps; Math-Shepherd (Wang et al., 2024), ThinkPRM (Khalifa et al., 2025), and Process-Bench (Zheng et al., 2025) advance step-level evaluation, complementary to prefix-level inheritability analysis (Daheim et al., 2024). Verifier and process-guided reasoning also includes trained math verifiers, process-reward-guided tree search, and automated process supervision (Cobbe et al., 2021a; Zhang et al., 2024a; Luo et al., 2024). Broader feedback and coordination paradigms include Constitutional AI and autonomous multi-agent reasoning (Bai et al., 2022; Shi et al., 2025). Recent studies examine reasoning-error mechanisms (Heyman and Zylberberg, 2025; Zhu et al., 2026); Targeted use of failure data (Lan et al., 2025; Ji et al., 2025; Chen et al., 2024a; Xia et al., 2024) motivates state-aware filtering. Studies of gradient-interference mitigation (Zhang et al., 2024b) and data-mixture optimization (Xie et al., 2023a) provide complementary perspectives on selecting training signals. Work on reward-model overoptimization, recursively generated-data collapse, and offline-demonstration quality further illustrates how training-data and feedback quality can affect learned systems (Gao et al., 2023; Shumailov et al., 2024; Mandelkar et al., 2021; Wan et al., 2025a,b). After the rollout pool is fixed, PRIMA collects state signals through a one-time instrumented replay and reuses them offline. Its scorer is fitted with automatic failure-category proxies; a separate author audit is used only as a mechanism check.

B Supplementary Arguments for State Closure

B.1 Distinguishing Closure from Terminal Correctness

Section 2.1 defines closure as whether a candidate prefix can serve as the condition for suffix replacement, rather than whether the complete response passes the external verifier. The terminal verdict $V(x, y)$ only provides the pass state of the full trajectory; closure is given by $r_\lambda(\phi(y^-; t))$ and $b_\lambda(y^-, t)$, and determines whether the failure prefix can still support a correct suffix.

Figure 7 provides a supplementary view of this relationship. Successful trajectories occupy the terminal-correct region; selected failures RC occupy the terminal-incorrect but state-closed region;

rejected failures UC occupy the terminal-incorrect and state-unclosed region. PRIMA only rewrites RC into same-prefix training units.

Closure is not a synonym for answer extractability, code parsability, or natural termination. The feature vector used by C_λ , b_λ , and τ_λ contains only G_t , F_t , and S_t ; B_t is logged for auditing and is not a scorer input. The calibration targets are nevertheless derived from verifier failure outcomes, as detailed in Appendix C.2.

Event-aligned evidence shows that state divergence predates terminal verdicts. In Figure 3, RC maintains low geometric residuals around the event position, while UC rapidly leaves the successful neighborhood after the local branch. Verifier failure only serves as the entry criterion for the analysis pool; what truly determines the training unit is whether the failure prefix can still support a correct suffix.

B.2 Pointwise Verification of the Contraction Theory

Section 2.2 showed that mixing RC and UC can contract the first-order gradient norm. Figure 8 compares the equal-weight, equal-norm geometric reference with the measured NLL-gradient diagnostic across four base models, with a mean absolute error of 0.043.

B.3 Supplementary Formal Arguments

B.3.1 Upper Bound on the Contraction Ratio

Section 2.2 defines

$$\rho = \frac{\|g_{\text{mix}}\|}{\alpha\|g_{\text{UC}}\| + (1 - \alpha)\|g_{\text{RC}}\|} \in [0, 1]. \quad (16)$$

We assume the denominator is positive and provide the closed-form upper bound and equality condition. Let

$$a = \alpha\|g_{\text{UC}}\|, \quad b = (1 - \alpha)\|g_{\text{RC}}\|, \quad (17)$$

and denote by θ the angle between g_{UC} and g_{RC} . Direct expansion yields

$$\|g_{\text{mix}}\|^2 = a^2 + b^2 + 2ab \cos \theta, \quad (18)$$

hence

$$\rho^2 = \frac{a^2 + b^2 + 2ab \cos \theta}{(a + b)^2} = 1 - \frac{2ab(1 - \cos \theta)}{(a + b)^2}. \quad (19)$$

By the triangle inequality, $\rho \leq 1$, with $\rho = 1$ if and only if $ab(1 - \cos \theta) = 0$. In the non-degenerate

case $a, b > 0$, equality therefore requires positive alignment ($\cos \theta = 1$).

Contraction is strongest when $a = b$, yielding

$$\rho^2 = \frac{1 + \cos \theta}{2}, \quad \rho = \sqrt{\frac{1 + \cos \theta}{2}}. \quad (20)$$

This corresponds to the equal-weight, equal-norm worst case in the main text. When $a \neq b$, contraction under the same angle is weaker, since maximum cancellation occurs in the symmetric case.

B.3.2 Rationale for the Logistic Closure Score

We use the logistic form of r_λ because its sigmoid output supplies a single threshold λ_c , while the linear weights keep the roles of G_t , F_t , and S_t explicit across source models. The random forest is retained only as a nonlinear diagnostic for the ablation in Appendix D.

C Experimental Setup, Learning-Based Baselines, and Supplementary Results

C.1 Shared Candidate Pool, Models, and Isolation Rules

Unless otherwise noted, all full-size main-table methods use the same self-distillation candidate pool. For each problem, four trajectories are sampled from the corresponding base model and labeled by the original task verifier. SFT baselines filter positives; DPO baselines construct pairs. PRIMA locates selected prefixes and draws suffix donors only from successful trajectories generated by that same source model. Each base model independently constructs its own pool and training set.

Table 2 provides sampling and verifier settings. Code and math tasks share sampling parameters; the code domain uses an external executor to verify complete programs, with a single-run timeout of 5 seconds. Code-domain training trajectories come from Codeforces/OpenR1 (Penedo et al., 2025), with final evaluation on LiveCodeBench (Jain et al., 2024). Math-domain training trajectories come from GSM8K (Cobbe et al., 2021b) and an OlympiadBench (He et al., 2024) subset filtered for AIME similarity, with final evaluation on AIME 2024/2025. Table 3 summarizes the isolation rules.

Training Problem Volume The code-domain training source pool draws 2000 problems from Codeforces/OpenR1. The math-domain training source pool consists of 1000 problems from GSM8K and 1000 problems from OlympiadBench,

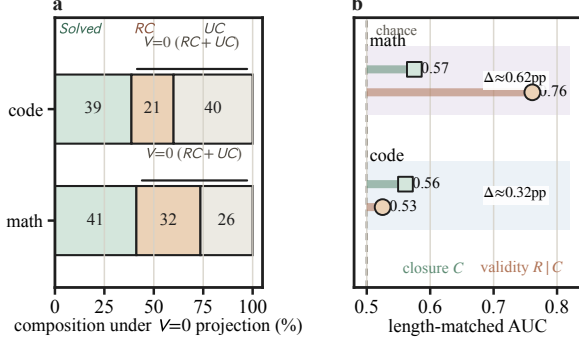


Figure 7: Closure states vs. terminal correctness. (a) Three-class partition under fixed terminal verdict. (b) Closure vs. terminal signals across domains.

Table 2: Candidate pool and verifier settings.

Item	Value
Trajectories per problem	4
Temperature	0.6
Top- p	0.95
Max new tokens	24,000
Stop condition	None
Code execution timeout	5 s

Table 3: Training, calibration, and evaluation isolation rules.

Stage	Isolation Rule
Candidate pool	Used only for constructing same-problem correct/failed trajectories and candidate pairs
State calibration	Used only for threshold and closure discrimination estimation; not fed back into final evaluation problems
Training	Final evaluation problems are never written into the candidate pool or training set
Evaluation	Only the external verifier is used to compute final pass@1

totaling 2000 problems. For each training problem, 4 trajectories are sampled according to the settings in Table 2 and labeled pass or fail by the respective domain verifier. Math-domain samples undergo AIME-similarity filtering before entering the final training pipeline, so AIME results in the main text and tables are always isolated from the training pool.

Scope and Token Budget This work targets reasoning analysis rather than task accuracy maximization. Generation and all prefix analyses use the maximum-new-token setting in Table 2; the same setting applies to every method in the fixed-pool comparison.

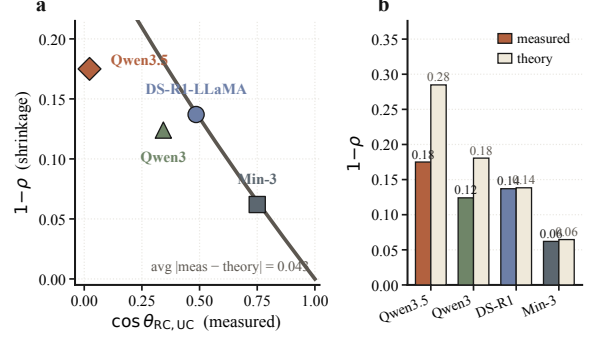


Figure 8: Contraction theory verification. (a) Equal-weight, equal-norm theory vs. measured contraction. (b) Per-model $1 - \rho$, MAE = 0.043.

C.2 State Features, Calibration, and Reproducible Implementation

Candidate prefixes are characterized by three state features. Geometric residual G_t is computed over the hidden-state neighborhood of same-problem or similar-problem successful trajectories and measures the current state’s deviation from the successful neighborhood. Local branching response F_t computes local second-order changes along the fixed complete forward-probing sequence and accumulates them with exponential decay to capture local divergence. Prefix support S_t is based on the first textual divergence position between same-problem correct and failed trajectories and characterizes shared structure at the discrete symbolic level.

Format audit B_t records parsability, termination status, and object presence but is not an input to the closure scorer. Calibration is performed separately by domain on 15,641 failed trajectories spanning code and mathematics and four source models: DeepSeek (4,176), Qwen3 (4,047), Ministral (3,999), and Qwen3.5 (3,419). Each trajectory receives a deterministic failure-category proxy derived from verifier outcome and generation metadata. For code, wrong answer (WA) is the RC proxy, while compilation error (CE), runtime error (RE), and time-limit exceeded (TLE) are UC proxies. For mathematics, a finite wrong answer is the RC proxy and a length-capped output is the UC proxy. These labels are operational calibration targets, not the definition of closure, and no human judgment enters scorer fitting.

Within each source model, G_t , F_t , and S_t undergo quantile normalization to obtain $\bar{G}_t, \bar{F}_t, \bar{S}_t \in [0, 1]$. We fit a class-balanced logistic scorer on source-balanced, problem-blocked calibration

Table 4: Author-audit majority vote against automatic proxy labels ($N = 240$).

	Auto RC	Auto UC	Total
Human repairable	105	8	113
Human non-repairable	15	112	127
Total	120	120	240

Table 5: Main experiment training hyperparameters.

Item	Value
LoRA rank	16
LoRA alpha	16
LoRA dropout	0.05
LoRA modules	q/k/v/o and gate/up/down projection layers
Learning rate	1×10^{-4}
Batch size	16
Epochs	1
DPO β	0.1
Reference model	Corresponding SFT checkpoint
Precision	bfloat16

splits. The closure threshold λ_c and cut thresholds $s_c = 0.45$ and $f_c = 0.65$ are selected on the calibration data. Held-out Repair@4 results are not used to fit or select the scorer.

Independent Author Audit To assess whether the automatic proxies match the intended mechanism, three authors independently reviewed a stratified random sample of 240 failures: four source models $\times$ two domains $\times$ two automatic labels $\times$ 15 examples. The automatic verdict and category were hidden; reviewers saw only the problem and full failed trajectory and judged whether retaining the first half while replacing the latter half could recover a solution. Majority vote agreed with the automatic label on 217 of 240 cases (90.42%). Inter-rater agreement was Fleiss’ $\kappa = 0.77$, with unanimous judgments on 82.5% of cases. This audit was conducted after label construction and did not affect features, thresholds, training, or evaluation.

Forward probing assets come from attention summaries and memory summaries collected across the four 8B-class base models. Each rollout is replayed under a thin instrumentation layer that, at every decoding step, exports a token-aligned record so that downstream features can be re-derived offline without re-running inference. Each attention-summary step records, per Transformer block, the normalized attention quality over the prompt region and over the generated region separately; the prompt-region attention entropy, used as a proxy for how diffusely the model is attend-

ing back to the input; and the top- k concentration mass of the attention distribution, used as a proxy for whether the model has locked onto a small set of supporting tokens. Each memory-summary step records the ℓ_2 norm $\|\mathbf{h}_t\|_2$ of the hidden-state residual; the memory entropy of its post-normalization activation distribution, which captures distributional spread without depending on scale; the absolute-mean (abs_mean) statistic, which captures gross activation drift away from baseline; and the singular-vector concentration of the residual matrix, which captures how rank-collapsed the local representation has become. These records allow the same trajectories to be segmented offline without re-running the generator. PRIMA derives G_t and F_t from the residual streams and S_t from trajectory text. The remaining attention and memory summaries support auxiliary diagnostics and learned baselines; they do not enter C_λ , b_λ , or τ_λ .

C.3 Training Configuration

The main experiments use a unified LoRA configuration and a single-epoch training budget. LoRA and QLoRA serve only as a means of unifying training cost and are not a contribution of this paper. Table 5 lists the core hyperparameters of the main experiments.

SFT Data Composition and Mixing The two positive-example baselines on the SFT side differ in their inclusion criteria. RFT retains only complete trajectories with $V(x, y) = 1$, giving the cleanest signal but discarding all verifier-rejected samples. Quality-filtered positives adopt a broader criterion: beyond verified correct solutions, they also retain near-correct trajectories that are well-formatted, compilable, structurally intact, and close to passing. This set has wider coverage but also introduces a small number of verifier-rejected noisy samples. PRIMA-SFT does not simply relax this admission threshold; it augments the strictly verified correct solutions with PRIMA-selected repaired positives. Specifically, PRIMA rewrites RC trajectories that pass same-prefix repair verification into $\mathcal{D}_\lambda^{\text{rep}}$, and trains on

$$\mathcal{D}_{\text{PRIMA}}^{\text{SFT}} = \mathcal{D}_{\text{clean}}^+ \cup \mathcal{D}_\lambda^{\text{rep}}. \quad (21)$$

This mixture ensures that prefixed samples do not replace complete solutions. Training on repair trajectories alone would primarily teach suffix continuations conditioned on failed prefixes, omitting

Table 6: Learning-based pair scorer configuration.

Attribute	Feature Selector	Text Reranker
Model type	GBM	Qwen3-Reranker-0.6B (Zhang et al., 2025), full fine-tuning
Input	42-dim structured features	Positive and negative raw text
Training sample size	3743 pairs (527 pos, 3216 neg)	Train 3735, val 633, cross-source test 631
Core objective	Binary classification pair scoring	Hard + 0.35 soft + 0.2 pairwise
Context length	N/A	32768
Optimizer / Learning rate	N/A	AdamW / 1.5×10^{-5}
Batch setting	N/A	Batch 4, grad_accum 8
Training epochs	N/A	1 epoch
Validation performance	5-fold OOF AUC = 0.9987	Val AUC $\approx$ 0.70, cross-source AUC $\approx$ 0.60
Inference input	Structured features	Text only
Use case	Top-K pair selection	Source-balanced top-K reranking

Table 7: Per-model downstream pass@1 (%), three-run average. Aggregated results are shown in Figure 6.

Method	LiveCodeBench				AIME			
	Qwen3 8B	Qwen3.5 9B	DS-R1 Llama-8B	Min-3 8B	Qwen3 8B	Qwen3.5 9B	DS-R1 Llama-8B	Min-3 8B
base	34.80	46.20	22.22	35.96	65.00	73.33	31.67	68.33
<i>SFT</i>								
RFT	35.38	45.03	21.35	36.26	66.67	68.33	33.33	65.00
Low-entropy filtering	35.67	45.61	22.81	36.55	66.67	70.00	35.00	70.00
Quality-filtered positives	36.26	47.66	23.39	36.84	68.33	75.00	35.00	70.00
PRIMA-SFT	37.43	48.83	24.56	38.01	70.00	75.00	36.67	71.67
<i>DPO</i>								
Random pairing	35.09	46.49	22.81	36.26	66.67	71.67	33.33	70.00
Shared-prefix length	36.55	47.66	24.56	37.72	68.33	75.00	35.00	71.67
Entropy sorting	37.13	47.37	25.15	37.43	68.33	75.00	33.33	70.00
Text reranker	38.30	48.54	26.61	38.89	68.33	71.67	33.33	68.33
Feature selector	39.77	49.71	28.65	40.35	70.00	75.00	36.67	71.67
PRIMA-DPO	41.23	50.29	30.99	41.52	71.67	76.67	38.33	73.33

trajectories from problem origin to terminal answer. Native positives provide complete trajectories satisfying the terminal verifier, while repaired positives provide intermediate-state recovery signals. For mathematics, neither source guarantees that every intermediate step is valid.

C.4 Selector and Reranker Training Details

In addition to PRIMA, this paper compares two classes of learning-based pair scorers. The feature selector trains a shallow GBM on structured features; the text reranker fine-tunes a lightweight text model directly on positive-negative sample texts. Both are built on the same pairing candidate pool but still score at the complete-response level, without constructing same-prefix training units.

The feature selector uses 42 hand-crafted features covering surface alignment, geometry and dynamics, PCA projections, and length- and health-derived quantities. The text reranker’s soft labels during training come from FRSD signals, including gradient, hidden-memory, pre/post geometry, and route-RLIS; at inference time it reads only raw text. It requires no hidden-state parsing or hand-crafted features and is a compute-efficient standalone data-

selection baseline. PRIMA has higher reported means than both learned scorers, suggesting that finer supervision units contribute beyond complete-response scoring alone.

C.5 Full Downstream Verification Results

Table 7 expands the per-model absolute scores of Figure 6. The four 8B-class base models evaluated are Qwen3-8B (Yang et al., 2025), Qwen3.5-9B (Team, 2026), DS-R1-Llama-8B (DeepSeek-AI, 2025), and Ministral-3-8B (Liu et al., 2026). All methods share the same 4-trajectory self-distilled candidate pool; SFT and DPO methods enter their respective training pipelines. The SFT side compares RFT, low-entropy filtering, quality-filtered positives, and PRIMA. The DPO side compares random pairing, shared-prefix-length, entropy sorting, text reranker, feature selector, and PRIMA. Both LCB and AIME report final verifier pass rates in percentage. Qwen3.5-4B and Qwen3.5-2B are smaller student models distilled from Qwen3.5-9B; their results appear in Appendix Figure 9.

C.6 Run-Level Uncertainty for the Main Comparison

Table 8 reports three-run mean $\pm$ standard deviation for PRIMA and the strongest fixed-pool baseline in each objective: quality-filtered positives for SFT and the feature selector for DPO. PRIMA has 15 higher means and one tie across the 16 cells. These repeated runs describe run-to-run variation; they do not make each AIME cell individually well-resolved.

C.7 Distillation to Smaller Student Models

To test whether PRIMA-DPO’s data workflow can transfer to smaller student models, this paper uses Qwen3.5-9B as the teacher and applies the same 9B-based PRIMA-DPO training data to train Qwen3.5-4B and Qwen3.5-2B. Baselines are the same-size base models, RFT, and the strongest non-PRIMA baseline, the feature selector. Figure 9 shows that PRIMA-DPO students exceed all three baselines across both sizes and both tasks. Relative to the feature selector, the 4B student gains 1.4 and 3.3 percentage points on LCB and AIME, and the 2B student gains 1.8 and 3.3 percentage points. These results show transfer to the two tested smaller student sizes and support using PRIMA-DPO data for distillation in this setting.

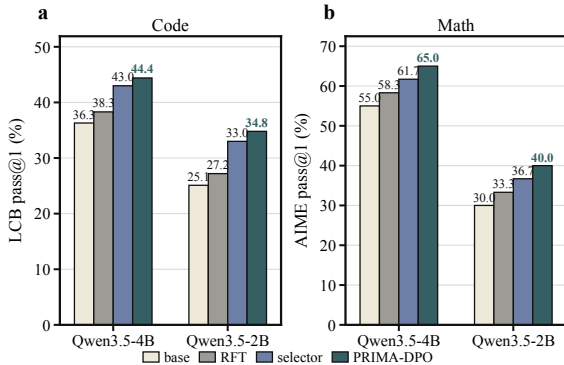


Figure 9: Distillation: Qwen3.5-9B teacher $\rightarrow$ 4B/2B students. (a) LiveCodeBench; (b) AIME.

C.8 Internal Signal Collection from the vLLM Decoding Path

Forward probing signals used in the main text are gathered by attaching hooks to the vLLM (Kwon et al., 2023) decoding path at inference time. By default, vLLM’s PagedAttention does not surface intermediate-layer residual streams or attention matrices. We register three hooks on the decoding path: intercepting hidden residuals at the output

of specified Transformer blocks; temporarily disabling the prefill-only optimization so per-layer attention weights can be collected in full; and keying every record by request id so that trajectories can be reconstructed under batch scheduling. Attention collection is only switched on during the signal-preparation phase; training and final inference continue on the standard paged path. During collection, runtime grows by roughly $1.4\times$ and memory by $1.6\times$ relative to pure decoding; once the signals are written to disk they are fully decoupled from the inference framework. These hooks collect the residual streams used to compute G_t and F_t and the auxiliary summaries used in diagnostics and learned baselines; the model consumes only text during training.

D Ablation of the Three State Projections

To test the contribution of G_t (geometric residual), F_t (branching response), and S_t (prefix support), we run a leave-one-out ablation on the random-forest auditor used as a nonlinear sanity reference for the logistic discriminator in Appendix B.3.2. Each projection is removed in turn, and the resulting ROC-AUC drop (ΔAUC) is measured by cross-validation with multiple random seeds on the pooled failed-trajectory set across four backbones, separately for math and code. We report $\Delta\text{AUC}/(\text{AUC}_{\text{full}} - 0.5)$, the fraction of above-chance separation lost.

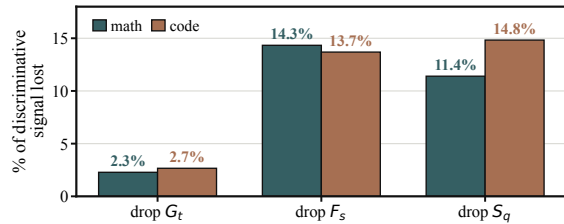


Figure 10: Leave-one-out ablation of the three state projections. Bars show $\Delta\text{AUC}/(\text{AUC}_{\text{full}} - 0.5)$ (%), i.e. the fraction of above-chance discriminative signal lost when each projection is removed, pooled across four 8B-class backbones for the math and code domains. All six entries are positive, showing that each feature group adds measurable predictive signal in this ablation.

Figure 10 reports positive AUC drops for all three removals in both domains. In this ablation, each feature group therefore adds measurable predictive signal. F_t and S_t account for most of the above-chance separation, while G_t adds a smaller share. This pattern is consistent with the three axes

Table 8: Pass@1 (%), three-run mean $\pm$ standard deviation, against the strongest fixed-pool baseline.

Task	Model	SFT baseline	PRIMA-SFT	DPO baseline	PRIMA-DPO
LCB	Qwen3-8B	36.26 $\pm$ 1.27	37.43 $\pm$ 1.27	39.77 $\pm$ 1.01	41.23 $\pm$ 1.27
LCB	Qwen3.5-9B	47.66 $\pm$ 1.01	48.83 $\pm$ 1.01	49.71 $\pm$ 1.01	50.29 $\pm$ 1.01
LCB	DS-R1-Llama-8B	23.39 $\pm$ 1.27	24.56 $\pm$ 1.27	28.65 $\pm$ 1.27	30.99 $\pm$ 1.01
LCB	Ministral-3-8B	36.84 $\pm$ 1.27	38.01 $\pm$ 1.27	40.35 $\pm$ 1.01	41.52 $\pm$ 1.01
AIME	Qwen3-8B	68.33 $\pm$ 2.89	70.00 $\pm$ 2.89	70.00 $\pm$ 4.41	71.67 $\pm$ 2.89
AIME	Qwen3.5-9B	75.00 $\pm$ 2.89	75.00 $\pm$ 2.89	75.00 $\pm$ 4.41	76.67 $\pm$ 1.67
AIME	DS-R1-Llama-8B	35.00 $\pm$ 2.89	36.67 $\pm$ 2.89	36.67 $\pm$ 4.41	38.33 $\pm$ 2.89
AIME	Ministral-3-8B	70.00 $\pm$ 2.89	71.67 $\pm$ 2.89	71.67 $\pm$ 4.41	73.33 $\pm$ 2.89

in Appendix B.1, but the leave-one-out result alone does not establish statistical independence or rule out redundancy under other models.