MILE: MILESTONE-GUIDED REINFORCEMENT LEARNING FOR MATHEMATICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning has shown strong potential for improving the mathematical reasoning capabilities of large language models (LLMs). However, existing methods often depend on sparse outcome feedback, manually designed process signals, or separately trained reward models, leaving expert reasoning trajectories underexploited as a rich source of dense supervision. We propose MILE, a milestone-guided reinforcement learning framework that models reasoning progress through intermediate states characterized by their future solvability. MILE extracts milestones from multiple expert solutions and constructs dense rewards for student reasoning transitions by measuring both progress and directional consistency. These rewards are incorporated into a two-stage GRPO optimization framework, enabling fine-grained transition-level learning followed by trajectory-level reasoning optimization. Experiments on mathematical reasoning benchmarks show that MILE consistently outperforms representative RL and distillation baselines. Further analyses validate the effectiveness of its components, scalability with expert trajectories, and improved reasoning efficiency.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) is a widely used post-training paradigm for LLM reasoning (Guo et al., 2025). By optimizing generated solutions against verifiable outcomes, RLVR has achieved strong performance on challenging reasoning tasks. However, outcome rewards provide limited supervision over intermediate reasoning, motivating richer trajectory-level guidance.

To provide richer supervision over intermediate reasoning, existing approaches broadly fall into two paradigms: Process Supervision and On-Policy Distillation (Uesato et al., 2022; Agarwal et al., 2024). *Process Supervision* evaluates intermediate reasoning states and provides dense feedback throughout the trajectory (Uesato et al., 2022; Setlur et al., 2024). Representative approaches derive supervision from Monte-Carlo completions (Wang et al., 2024), human-annotated reasoning steps (Lightman et al., 2024), verification-based reasoning (Khalifa et al., 2025; Luo et al., 2024; Zhang et al., 2025), or process reward models learned from policy rollouts and outcome labels (Cui et al., 2025a; Yuan et al., 2024). *On-Policy Distillation* uses a stronger teacher to supervise trajectories generated by the student policy (Agarwal et al., 2024). Representative approaches include distribution matching over student-generated trajectories (Agarwal et al., 2024), token-level teacher supervision (Agarwal et al., 2024), and more recent variants that incorporate contextual or near-future guidance (Ye et al., 2026; Jiang & Ferraro, 2026).

However, these approaches exhibit distinct limitations in supervising reasoning progress. Process Supervision provides dense feedback, but typically focuses on the local correctness or quality of individual steps, which may not reflect whether a step actually moves the solution forward (Chen et al., 2025; Li & Li, 2025; Setlur et al., 2024; Zhou et al., 2026). It also often relies on a separate reward model, requiring additional supervision such as step-level labels or sampled rollouts, followed by reward-model training and maintenance as the policy evolves (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024; Luo et al., 2024). On-Policy Distillation provides finer-grained guidance from a stronger teacher, but its supervision is tied to the teacher's behavior on student-generated trajectories (Agarwal et al., 2024). This can bias learning toward particular teacher-preferred reasoning paths, even when alternative paths are equally valid (Wang et al., 2023), and repeated teacher queries during policy optimization introduce substantial training cost (Agarwal et al., 2024). These limitations share a deeper cause: the supervision mainly captures local step quality or teacher agreement, rather than how much an intermediate state improves the chance of future success (Setlur et al., 2024; Li

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

& Li, 2025; Zhou et al., 2026). Effective reasoning supervision therefore requires a dense progress signal that measures meaningful advances toward the solution while allowing multiple valid reasoning paths.

Motivated by the *Subgoal Learning Model* (Catrambone, 1998), we model reasoning progress as transitions between a small set of intermediate states that indicate meaningful reductions in the remaining problem difficulty. We refer to these states as *milestones* and characterize their importance through *future solvability*: a reasoning state represents greater progress if continuing from that state provides a higher likelihood of reaching the correct solution. Unlike local step correctness or similarity to a specific reasoning trajectory, future solvability captures whether an intermediate state contributes to eventual problem solving, providing a more general criterion for evaluating reasoning progress across diverse solution paths.

Based on this formulation, we propose MILE, a milestone-guided reinforcement learning framework that derives dense supervision from reasoning milestones across expert trajectories. MILE first extracts meaningful milestones from multiple correct expert trajectories by estimating the future solvability of intermediate reasoning states. These milestones form a problem-specific reference space that captures diverse reasoning progress toward the final answer. Based on this space, MILE constructs dense rewards for student reasoning transitions, where progress reward measures movement toward states with higher future solvability and direction reward encourages transitions consistent with expert reasoning progress. Finally, we optimize the student policy with a two-stage GRPO strategy, where transition-level training learns effective local reasoning behaviors and trajectory-level training further improves end-to-end problem solving. Overall, these components enable MILE to guide reasoning toward meaningful progress while preserving flexibility across diverse solution paths.

Our contributions are summarized as follows:

- We introduce a milestone-based view of reasoning progress, where meaningful intermediate states are characterized by their impact on future solvability rather than local step correctness.
- We propose MILE, a milestone-guided reinforcement learning framework that derives dense rewards from expert reasoning milestones by modeling both reasoning progress and directional consistency.
- We show that MILE improves both reasoning accuracy and efficiency, outperforming its GRPO backbone by 5.5 points and the strongest process-supervision baseline by 3.4 points while maintaining strong performance under limited generation budgets.

2 METHOD

MILE trains a reasoning policy with dense supervision derived from expert trajectories, without requiring a separate reward model. The framework consists of three stages: (1) extracting milestones from expert trajectories based on future solvability; (2) constructing progress and direction rewards from these milestones to evaluate student reasoning transitions; and (3) optimizing the student policy with a two-stage GRPO procedure from transition-level to trajectory-level learning. Figure 1 provides an overview of the framework.

2.1 PROBLEM SETUP

Our goal is to assign a dense progress reward to every reasoning state of a reasoning trajectory. Given a problem x, the *student policy* π_ϕ is the trainable reasoning model optimized by reinforcement learning. It generates a reasoning trajectory

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_T),$$

where s_t denotes the reasoning state that contains all reasoning steps up to step t and a_t is the token sequences generated to extend s_t to s_{t+1}. Specifically, s_0 contains the problem x and the user input. We aim to assign each transition (s_t, a_t, s_{t+1}) a progress reward indicating whether the new state makes the problem easier to solve correctly. This asks about the *future solvability* of a state rather than the local correctness of a step, since a step can be logically valid yet contribute nothing toward the final answer. For each problem, we derive the progress reward from a set of expert trajectories $\mathcal{T}_x = \{\tau^{(1)}, \tau^{(2)}, \ldots, \tau^{(K)}\}$, where is generated by multiple teacher reasoning models.

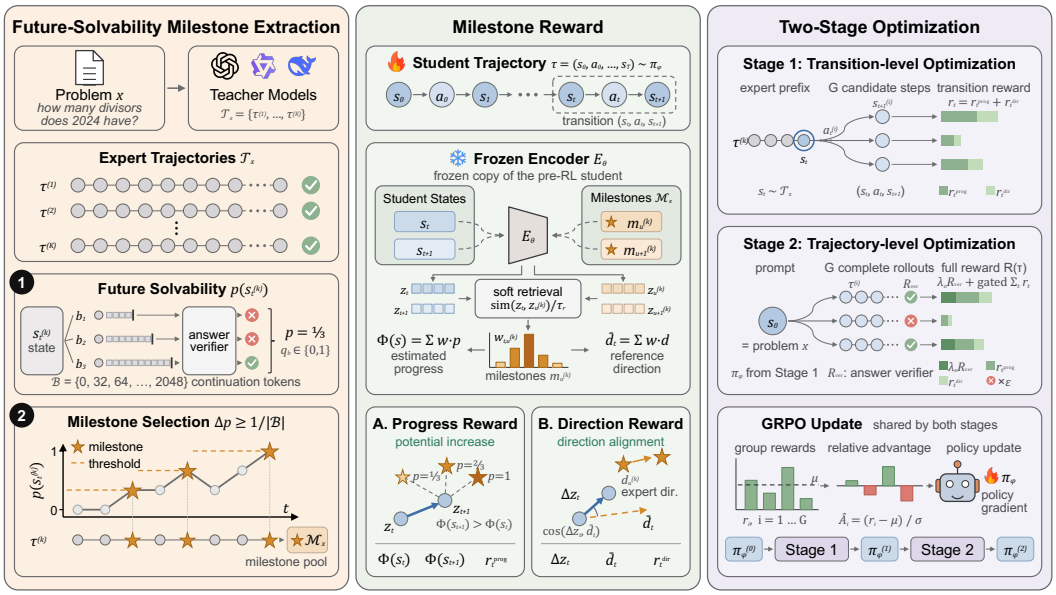


Figure 1: Overview of MILE. Left, milestone extraction: each expert state is scored by its future solvability, and states where this score jumps are kept as milestones (stars) and pooled across the trajectories of a problem. Middle, milestone reward: a frozen encoder embeds the pooled milestones and the student states, and soft retrieval yields a progress reward (gain in solvability) and a direction reward (agreement with the expert direction). Right, two-stage optimization: the student trains first on single transitions from expert prefixes and then on full rollouts, with the dense reward gated by the final-answer verifier.

2.2 Future-Solvability Milestone Extraction

Given a problem x and its set of expert solution trajectories $\mathcal{T}_x$, our goal is to identify a small set of intermediate reasoning states that mark meaningful progress toward the correct answer. We call these states *milestones*, which are used to guide the student model's learning.

Specifically, we quantify the progress of each intermediate reasoning state through its *future solvability*, i.e., how likely the problem can be solved correctly when continuing from that state. Given an expert trajectory $\tau^{(k)} \in \mathcal{T}_x$, let $s_t^{(k)}$ denote the reasoning state after its first t steps. To estimate the future solvability of $s_t^{(k)}$, we consider a set of continuation-token budgets $\mathcal{B}$. For each budget $b \in \mathcal{B}$, we independently sample three continuations from $s_t^{(k)}$, each with at most b additional tokens, and define $q_b(s_t^{(k)}) = 1$ if any continuation reaches the correct answer, and $q_b(s_t^{(k)}) = 0$ otherwise. We then define the progress score as the average solvability across different budgets:

$$p(s_t^{(k)}) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} q_b(s_t^{(k)}).$$

A higher $p(s_t^{(k)})$ therefore indicates that the current state provides a stronger basis for successfully completing the remaining reasoning.

Second, based on the progress scores, we identify a sparse set of states that represent meaningful advances in reasoning. Specifically, for each expert trajectory $\tau^{(k)}$, we select a new milestone whenever its progress score increases by at least $1/|\mathcal{B}|$ relative to the previously selected one. Let $m_u^{(k)}$ denote the position of the u-th milestone. Starting from $m_0^{(k)} = 0$, we recursively select

$$m_{u+1}^{(k)} = \min \left\{ t > m_u^{(k)} : p\left(s_t^{(k)}\right) - p\left(s_{m_u^{(k)}}^{(k)}\right) \geq \frac{1}{|\mathcal{B}|} \right\},$$

until no such state exists. This criterion retains the earliest state that achieves each meaningful increase in future solvability, yielding a compact sequence of milestones along the trajectory.

Finally, we aggregate the milestones identified from all expert trajectories into a problem-specific milestone pool:

$$\mathcal{M}_x = \bigcup_k \left\{ \left(s_{m_u^{(k)}}^{(k)}, p\left(s_{m_u^{(k)}}^{(k)}\right) \right) : u = 0, \ldots, U_k \right\},$$

where U_k denotes the index of the last selected milestone in $\tau^{(k)}$. Each milestone is paired with its progress score, while pooling across trajectories preserves diverse reasoning paths toward solving the same problem (Niu et al., 2025; Wang et al., 2023).

2.3 MILESTONE REWARD

Given the problem-specific milestone pool $\mathcal{M}_x$ constructed above, our goal is to provide dense supervision for each student transition (s_t, a_t, s_{t+1}) by measuring its alignment with teacher reasoning progress. To this end, we design two complementary rewards: a *progress reward* that measures whether the student advances toward higher-progress milestones, and a *direction reward* that measures whether the transition follows the local progression of teacher trajectories. We then combine these dense rewards with the final-answer correctness reward to obtain the overall training reward.

Progress reward. We first establish a shared representation space between the student states and teacher milestones. Specifically, we use E_θ, a frozen copy of the student encoder, to encode the current student state s_t and each teacher milestone $s^{(k)}_{m_u^{(k)}} \in \mathcal{M}_x$:

$$z_t = E_\theta(s_t), \qquad z_u^{(k)} = E_\theta\left(s^{(k)}_{m_u^{(k)}}\right).$$

We then measure the relevance of each teacher milestone to the current student state. For each milestone $s^{(k)}_{m_u^{(k)}}$, we define its weight with a softmax over cosine similarities:

$$w_{t,u}^{(k)} = \frac{\exp\left(\mathrm{sim}\left(z_t, z_u^{(k)}\right)/\tau_r\right)}{\sum_{k',u'} \exp\left(\mathrm{sim}\left(z_t, z_{u'}^{(k')}\right)/\tau_r\right)},$$

where τ_r is the retrieval temperature. This soft weighting allows the student state to receive supervision from multiple relevant milestones, potentially from different teacher trajectories. Based on these weights, we transfer the progress scores of teacher milestones to the student state. Specifically, we estimate its progress as

$$\Phi(s_t) = \sum_{k,u} w_{t,u}^{(k)} p\left(s^{(k)}_{m_u^{(k)}}\right).$$

Thus, a student state receives a higher estimated progress when it is more strongly aligned with teacher milestones of higher future solvability. We define the progress reward by measuring the improvement in estimated progress from s_t to s_{t+1}:

$$r_t^{\mathrm{prog}} = \frac{\Phi(s_{t+1}) - \Phi(s_t)}{(1 + \mathrm{Tok}(a_t))^\gamma},$$

where $\mathrm{Tok}(a_t)$ denotes the number of tokens in a_t, and $\gamma > 0$ controls the strength of length normalization. This normalization favors transitions that achieve greater progress with fewer reasoning tokens. Consequently, r_t^{prog} rewards efficient transitions toward states with higher estimated future solvability.

Direction reward. While the progress reward evaluates how much progress a transition makes, it does not capture whether the student advances in a direction consistent with teacher reasoning. We therefore introduce a direction reward that compares the student's transition with the local progression directions of relevant teacher trajectories.

We first characterize the local reasoning direction of each teacher trajectory using two consecutive milestones. For each non-terminal milestone, we define the normalized direction to its next milestone as

$$d_u^{(k)} = \frac{z_{u+1}^{(k)} - z_u^{(k)}}{\left\| z_{u+1}^{(k)} - z_u^{(k)} \right\|_2}, \qquad u < U_k.$$

Next, we estimate an teacher reference direction tailored to the current student state s_t. Using the same milestone weights $w_{t,u}^{(k)}$ defined above, we aggregate the directions associated with relevant teacher milestones:

$$\bar{d}_t = \frac{\sum_k \sum_{u<U_k} w_{t,u}^{(k)} d_u^{(k)}}{\sum_k \sum_{u<U_k} w_{t,u}^{(k)}}.$$

The terminal milestone of each expert trajectory is excluded since it has no subsequent milestone. In this way, expert directions originating from milestones more relevant to s_t contribute more strongly to $\bar{d}_t$. Finally, we compare the actual student transition direction with this expert reference direction:

$$r_t^{\mathrm{dir}} = \cos\big(z_{t+1} - z_t, \bar{d}_t\big)\,,$$

A larger r_t^{dir} indicates that the student transition follows a direction more consistent with the progression of relevant expert trajectories.

Overall reward. Finally, we combine the progress and direction rewards with a final-answer verifier to construct the trajectory-level training reward:

$$R(\tau) = \lambda_o R_{\mathrm{ver}}(\tau) + [\epsilon + (1-\epsilon)R_{\mathrm{ver}}(\tau)] \sum_t \left(\lambda_p r_t^{\mathrm{prog}} + \lambda_d r_t^{\mathrm{dir}}\right).$$

Here, $R_{\mathrm{ver}}(\tau) \in \{0, 1\}$ indicates whether the trajectory τ produces the correct final answer. The coefficients λ_o, λ_p, and λ_d control the contributions of the verifier, progress, and direction rewards, respectively. The factor $\epsilon + (1-\epsilon)R_{\mathrm{ver}}(\tau)$ modulates the dense milestone rewards according to final-answer correctness: for a correct trajectory, the dense rewards are fully preserved, whereas for an incorrect trajectory, their contribution is down-weighted by ϵ.

2.4 Two-Stage Optimization

We optimize the student policy in two consecutive stages, both using GRPO. In the first stage, we perform *transition-level optimization* by decomposing expert reasoning trajectories into local reasoning transitions. Specifically, we sample intermediate states s_t from expert trajectories and let the student generate a group of candidate next steps a_t, each leading to a new state s_{t+1} and forming a transition (s_t, a_t, s_{t+1}). Each transition is evaluated using the progress and direction rewards introduced in Section 2.3, and the resulting rewards are used by GRPO to optimize the student policy toward effective local reasoning transitions. By learning from individual transitions rather than complete trajectories, this stage provides fine-grained supervision and reduces the impact of compounding reasoning errors during early training.

In the second stage, we perform *trajectory-level optimization* by continuing training from the policy obtained in the first stage. Starting from the original problem prompts, the student generates complete reasoning trajectories, each of which is evaluated using the full reward $R(\tau)$. GRPO then optimizes the policy based on these trajectory-level rewards, enabling the student to integrate the local reasoning capability acquired in the first stage into end-to-end problem solving.

3 Experiments

In this section, we evaluate MILE on mathematical reasoning benchmarks against representative RL and distillation baselines. We further analyze its key components, scaling with expert trajectories, and reasoning efficiency under different token budgets.

3.1 Experimental Setup

Training data. We build our training data from Big-Math (Albalak et al., 2025) and OpenR1-Math-220k, removing all problems that overlap with the evaluation benchmarks. For each problem, four teacher models, *Llama 4 Scout*, *Qwen3.5-122B-A10B*, *Qwen2.5-72B-Instruct* (Yang et al., 2024), and *gpt-oss-120b* (Agarwal et al., 2025), each generate two solutions using different prompts, yielding eight candidate expert trajectories. GPT-5.4 segments each trajectory into a consistent sequence of reasoning steps. We retain only problems for which all eight trajectories reach the reference answer, resulting in 22,531 problems and 180,248 verified expert trajectories. Following Section 2.2, we extract milestones by estimating the future solvability of each expert state using its corresponding teacher model with continuation-token budgets $\mathcal{B} = \{0, 32, 64, 128, 256, 512, 1024, 2048\}$, where $b = 0$ requires answering directly from the current state. Milestones are selected with a progress threshold of $1/|\mathcal{B}| = 1/8$ and pooled across the eight trajectories to form $\mathcal{M}_x$. Appendix A provides an example of the resulting training data.

Benchmarks. We evaluate the models on five mathematical reasoning benchmarks: MATH500 (Hendrycks et al., 2021; Lightman et al., 2024), AMC23, AIME25, AIME26, and OlymMATH-100 (Sun et al., 2026). We report Pass@1 over 16 random seeds as mean $\pm$ standard

Table 1: Main results on five mathematical reasoning benchmarks. Results are reported as mean ± standard deviation across 16 random seeds. Bold and underlined values indicate the best and second-best results, respectively. Additional experiments with a different backbone model are presented in Appendix D.3.

Method	MATH500	AMC23	AIME25	AIME26	OlymMATH	Overall
Initialization						
Base model	$82.3_{\pm 0.6}$	$87.5_{\pm 0.6}$	$64.3_{\pm 1.4}$	$84.4_{\pm 1.3}$	$35.7_{\pm 1.1}$	$70.84_{\pm 1.0}$
SFT-$K^\star$ (Ouyang et al., 2022)	$87.6_{\pm 0.5}$	$91.1_{\pm 0.4}$	$73.7_{\pm 1.2}$	$87.6_{\pm 1.0}$	$41.9_{\pm 1.1}$	$76.38_{\pm 0.8}$
Reinforcement learning baselines						
GRPO (Shao et al., 2024)	$91.8_{\pm 0.5}$	$94.5_{\pm 0.4}$	$83.3_{\pm 0.9}$	$90.0_{\pm 0.7}$	$45.7_{\pm 1.2}$	$81.07_{\pm 0.7}$
Short-RL (Yuan et al., 2025)	$90.2_{\pm 0.6}$	$93.9_{\pm 0.4}$	$80.4_{\pm 0.9}$	$89.1_{\pm 0.9}$	$45.0_{\pm 1.1}$	$79.72_{\pm 0.8}$
OREAL (Lyu et al., 2025)	$94.4_{\pm 0.4}$	$96.6_{\pm 0.3}$	$85.4_{\pm 0.8}$	$90.0_{\pm 0.8}$	$47.9_{\pm 1.0}$	$82.85_{\pm 0.7}$
PRIME (Cui et al., 2025a)	$92.7_{\pm 0.5}$	$95.7_{\pm 0.3}$	$84.8_{\pm 0.8}$	$90.8_{\pm 0.8}$	$47.8_{\pm 1.0}$	$82.36_{\pm 0.7}$
PURE (Cheng et al., 2025)	$95.0_{\pm 0.4}$	$\underline{96.9_{\pm 0.3}}$	$86.7_{\pm 0.7}$	$88.8_{\pm 0.8}$	$48.3_{\pm 1.0}$	$83.12_{\pm 0.6}$
ROVER (He et al., 2026)	$93.0_{\pm 0.6}$	$95.6_{\pm 1.7}$	$85.8_{\pm 3.1}$	$90.8_{\pm 2.6}$	$48.3_{\pm 2.4}$	$82.70_{\pm 1.0}$
Distillation baseline						
OPD (Agarwal et al., 2024)	$\underline{95.2_{\pm 0.5}}$	$96.7_{\pm 1.5}$	$87.1_{\pm 2.7}$	$91.5_{\pm 3.0}$	$50.0_{\pm 1.9}$	$\underline{84.10_{\pm 0.9}}$
Our methods						
MILE-Core	$93.6_{\pm 0.5}$	$96.5_{\pm 0.4}$	$\underline{88.4_{\pm 0.9}}$	$\underline{91.6_{\pm 0.8}}$	$\underline{50.2_{\pm 1.1}}$	$84.06_{\pm 0.7}$
MILE-Full	$\mathbf{98.0}_{\pm 0.3}$	$\mathbf{98.6}_{\pm 0.2}$	$\mathbf{90.6}_{\pm 0.6}$	$\mathbf{93.3}_{\pm 0.6}$	$\mathbf{52.3}_{\pm 1.1}$	$\mathbf{86.56}_{\pm 0.6}$

deviation, with Overall computed as the unweighted average across the five benchmarks. For the scaling and ablation studies, we additionally report three diagnostic metrics: retrieval entropy, direction alignment, and false progress, defined in Appendix B.

Baselines. All baselines start from the same SFT checkpoint and are trained on the same problems as MILE. We compare against three representative families of methods: (i) outcome-only reinforcement learning, including GRPO (Shao et al., 2024), Short-RL (Yuan et al., 2025), OREAL (Lyu et al., 2025), and ROVER (He et al., 2026); (ii) process-aware reinforcement learning, including PRIME (Cui et al., 2025a) and PURE (Cheng et al., 2025); and (iii) on-policy distillation, represented by OPD (Agarwal et al., 2024), using Qwen3.5-122B-A10B as the teacher.

Implementation Details. We use Qwen3.5-4B as the main student model in all experiments. All methods start from the same SFT checkpoint for a fair comparison. This checkpoint is trained on the expert trajectories. We evaluate two variants of MILE. MILE-Core uses only the milestone-based dense rewards. MILE-Full further adds final-answer supervision. This provides an additional learning signal for final-task correctness. All reinforcement learning methods are implemented in `verl` (Sheng et al., 2024). We use consistent training and evaluation protocols across methods whenever applicable. Detailed training, rollout, and inference settings are provided in Appendix C.

3.2 Main Results

We compare our MILE with all baselines on five mathematical reasoning benchmarks. All RL and distillation methods share the same model checkpoint and training problems. The results are summarized in Table 1.

From the results, we observe that: 1) Our MILE-Full achieves consistent improvements across benchmarks. It obtains the highest mean Pass@1 on all five benchmarks, achieving an Overall score of 86.56% and outperforming GRPO by 5.49 percentage points. It also exceeds PURE, the strongest RL baseline in Overall, and OPD by 3.44 and 2.46 percentage points, respectively. Despite its competitive performance, OPD requires approximately 2.5× the training time of MILE-Full due to repeated teacher queries for on-policy supervision throughout policy optimization. (2) Milestone-based rewards alone provide competitive policy supervision. Using only milestone-based rewards during policy optimization, MILE-Core achieves an Overall score of 84.06%, surpassing all RL baselines and nearly matching OPD (84.10%). Its mean scores also exceed those of all compared baselines on AIME25, AIME26, and OlymMATH-100. These results support the effectiveness of transferring expert reasoning progress through future-solvability milestones without training a separate reward model. (3) Final-answer supervision complements milestone-based guidance. Compared with MILE-Core, MILE-Full improves Overall by 2.50 percentage points, with gains on every benchmark. This comparison supports combining dense guidance on reasoning progress with explicit supervision of final-answer correctness.

Table 2: Ablation study of MILE. Besides Overall performance, we report three diagnostic metrics: False Progress measures the proportion of states assigned high progress scores without actual improvements in future solvability (lower is better); Direction Alignment measures the consistency between student reasoning transitions and expert progress directions (higher is better); Retrieval Entropy measures the diversity of retrieved expert reasoning paths (higher indicates stronger multi-path utilization). Detailed definitions and computation procedures of these metrics are provided in Appendix B.

Variant	Overall	False Prog. (%)	Dir. Align.	Re. Entropy
Full model				
MILE-Full	**86.56**$_{\pm 0.6}$	**6.9**$_{\pm 0.4}$	**0.768**$_{\pm 0.012}$	0.643$_{\pm 0.014}$
Milestone construction				
Uniform selection	83.68$_{\pm 0.7}$	11.2$_{\pm 0.6}$	0.702$_{\pm 0.014}$	0.571$_{\pm 0.017}$
Dense Milestones	83.42$_{\pm 0.8}$	12.2$_{\pm 0.5}$	0.676$_{\pm 0.015}$	0.694$_{\pm 0.013}$
Mixed Quality	85.12$_{\pm 0.7}$	9.8$_{\pm 0.5}$	0.729$_{\pm 0.015}$	0.596$_{\pm 0.016}$
Reward design				
w/o progress reward	83.84$_{\pm 0.8}$	13.4$_{\pm 0.6}$	0.742$_{\pm 0.013}$	0.617$_{\pm 0.016}$
w/o direction reward	84.58$_{\pm 0.7}$	12.1$_{\pm 0.5}$	0.547$_{\pm 0.016}$	0.628$_{\pm 0.013}$
w/o answer reward (= MILE-Core)	84.06$_{\pm 0.7}$	15.8$_{\pm 0.5}$	0.724$_{\pm 0.014}$	0.635$_{\pm 0.015}$
Training schedule				
w/ encoder sync	84.72$_{\pm 0.9}$	10.6$_{\pm 0.7}$	0.715$_{\pm 0.018}$	0.661$_{\pm 0.021}$
w/o Stage 2	82.86$_{\pm 0.8}$	13.7$_{\pm 0.6}$	0.751$_{\pm 0.013}$	0.612$_{\pm 0.016}$
w/o Stage 1	84.34$_{\pm 0.8}$	10.9$_{\pm 0.6}$	0.698$_{\pm 0.014}$	0.631$_{\pm 0.015}$

3.3 Ablation Study

We conduct ablation studies to investigate the contribution of different components in MILE, including milestone construction, reward design, and training schedule. For milestone construction, we consider three variants. Uniform Selection replaces future-solvability-based milestone selection with uniformly sampled states from expert trajectories, evaluating the importance of progress-aware milestone identification. Dense Milestones removes milestone sparsification and directly uses all intermediate reasoning states in the milestone pool. Mixed Quality replaces two out of the eight expert trajectories with trajectories generated by lower-capability teachers to evaluate robustness against variations in expert quality. For reward design, we remove the progress reward, direction reward, and answer reward (corresponding to MILE-Core) individually to study the contribution of each supervision signal. For training schedule, we evaluate the effect of each optimization component by removing Stage 1 transition-level optimization, removing Stage 2 trajectory-level optimization, and synchronizing the retrieval encoder with the student encoder during training.

Milestone construction. We first investigate the effect of different milestone construction strategies. Uniform Selection degrades performance, indicating that uniformly sampled states fail to capture meaningful improvements in future solvability. Dense Milestones also underperforms, suggesting that including all intermediate states introduces redundant supervision, while sparse milestones provide cleaner guidance. Meanwhile, Mixed Quality achieves comparable performance, demonstrating the robustness of MILE to moderate variations in expert trajectory quality.

Reward design. We next analyze the contribution of different reward components. Removing the progress reward leads to the largest performance drop, highlighting the importance of future-solvability-based supervision. Removing the direction reward reduces Direction Alignment, showing that expert reasoning directions provide complementary guidance. The removal of answer reward (MILE-Core) also decreases performance, indicating that outcome-level supervision further improves trajectory optimization. Together, these results demonstrate that the three reward components provide complementary signals.

Training schedule. Finally, we evaluate the proposed training strategy. Removing either Stage 1 or Stage 2 optimization degrades performance, validating the necessity of transition-level and trajectory-level optimization. Stage 1 provides fine-grained supervision for local reasoning behaviors, while Stage 2 improves end-to-end problem solving. In addition, synchronizing the retrieval encoder with the student encoder reduces performance, suggesting that a fixed representation space is important for stable milestone retrieval and reward estimation.

3.4 Scaling with More Expert Trajectories

The key design of MILE is to aggregate milestones from multiple expert trajectories for the same problem, which raises the question of whether additional expert solutions provide benefits beyond

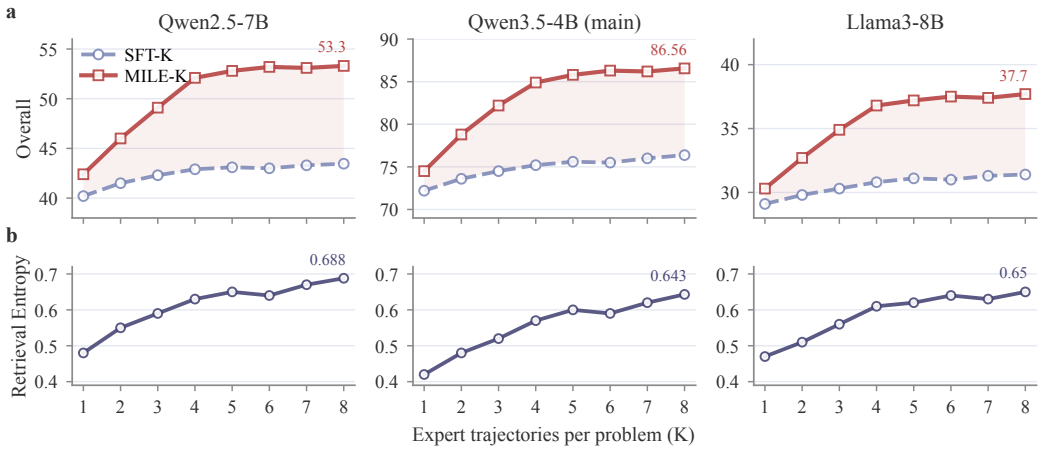


Figure 2: Scaling behavior of MILE with increasing numbers of expert trajectories.

conventional supervised learning. To study this effect, we vary the number of verified expert trajectories per problem K from 1 to 8 and construct training data using the corresponding subsets of expert trajectories. For each value of K, we train both SFT-K and MILE-K using exactly the same expert trajectories, ensuring a fair comparison. We conduct experiments on three student backbones and report the overall accuracy and retrieval entropy in Figure 2.

As shown in the figure, increasing the number of expert trajectories consistently improves the performance of MILE across different student backbones, while SFT exhibits much smaller gains. This indicates that additional expert trajectories provide valuable complementary reasoning signals when converted into milestone-based supervision, rather than merely increasing the amount of supervised data. The improvement becomes more significant as K grows, suggesting that aggregating milestones from multiple solutions enables the model to capture diverse reasoning patterns and identify more reliable progress signals. Moreover, the retrieval entropy of MILE increases with the number of expert trajectories, indicating that the model retrieves and utilizes a broader range of expert reasoning paths when more trajectories are available. This demonstrates that the milestone pool effectively preserves multi-path reasoning information and that the learned retrieval mechanism benefits from richer expert diversity.

3.5 Reasoning Efficiency and Budget Scaling

We evaluate the reasoning efficiency of MILE from two perspectives. First, we measure the average number of generated tokens per correctly solved problem under the standard setting, comparing MILE with SFT, GRPO, and Short-RL. Second, we vary the maximum generation budget from 256 to 4096 tokens and evaluate Pass@1 under different reasoning constraints. The results are summarized in Figure 3.

As shown in Figure 3(a), both MILE variants outperform GRPO while requiring fewer tokens per correct prediction, indicating that the improvements come from more effective reasoning rather than longer generations. MILE-Full further achieves the highest accuracy with comparable token consumption to Short-RL, avoiding the accuracy loss caused by overly compressed reasoning. Figure 3(b) shows that MILE consistently performs best across different generation budgets, with larger gains under tighter constraints. This suggests that milestone-based supervision enables the model to make effective reasoning progress with limited tokens. Overall, MILE improves reasoning efficiency by guiding the policy toward more productive reasoning trajectories rather than simply increasing generation length.

4 Related Work

Reward design for RLVR. RLVR is now the standard post-training recipe for reasoning, and methods within it differ mainly in the form of the training signal rather than the optimizer consuming it (Ouyang et al., 2022; Lambert et al., 2024). Existing signals can be ordered by granularity into three directions: (a) outcome-level rewards from the verified final answer, (b) step-level rewards from a trained process reward model (PRM), and (c) progress-level rewards that score a state by its

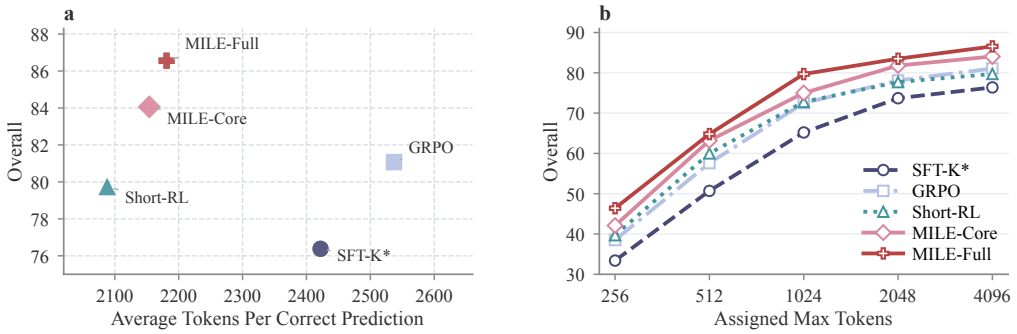


Figure 3: Reasoning efficiency and budget scaling of MILE. (a) Overall performance versus tokens per correct prediction. (b) Pass@1 under different generation budgets.

effect on eventual success. The first direction descends from answer verifiers (Cobbe et al., 2021) and matured with GRPO (Shao et al., 2024; Guo et al., 2025; Du et al., 2025); since the reward itself is fixed, later work refines only the estimator around it, through decoupled clipping (Yu et al., 2025b), debiased normalization (Liu et al., 2025), random-policy values (He et al., 2026), and best-of-N reweighting (Lyu et al., 2025), yet the ceiling appears intrinsic: RLVR sharpens rather than expands the base distribution (Yue et al., 2025; Cui et al., 2025b). The second direction instead labels each step (Uesato et al., 2022), drawing them from human annotation (Lightman et al., 2024), Monte-Carlo completions (Wang et al., 2024), tree search (Luo et al., 2024), or LLM-judge filtering (Zhang et al., 2025), with generative verifiers replacing the classifier (Khalifa et al., 2025), though transfer beyond the training distribution remains poor (Zheng et al., 2025; Song et al., 2025). The third direction rejects local correctness and scores a step by its effect on future success (Setlur et al., 2024), via Q-value ranking (Li & Li, 2025), min-form aggregation (Cheng et al., 2025), look-ahead values (Chen et al., 2025), or Monte-Carlo values in place of a critic (Kazemnejad et al., 2025), in the tradition of potential-based shaping (Ng et al., 1999); recent variants make the signal verifiable through prefix solve-rate gain (Zhou et al., 2026) or checkable sub-goals (Chen et al., 2026), while implicit PRMs read token-level rewards off an outcome model (Yuan et al., 2024; Cui et al., 2025a). However, the signal is in every case either a terminal scalar crediting all intermediate states equally, or an auxiliary estimator that must be separately trained and continually recalibrated against a moving policy.

Supervision from expert reasoning trajectories. A complementary line instead draws supervision from expert reasoning traces, which are abundant in distilled corpora but do not directly provide rewards for RL. These methods can be ordered by how tightly they bind the student to the expert's own tokens: (a) offline imitation of complete traces, (b) on-policy distillation, and (c) integration of expert traces into RLVR. The first direction fine-tunes on expert chains of thought, refined by mixing long and short chains for efficiency (Yu et al., 2025a) or by a preference objective over distilled chains (Niu et al., 2025), but trains only on states the expert visited. The second direction closes that gap by keeping the student on-policy, letting the teacher score the student's own rollouts and turning per-token teacher likelihood into a dense reward (Agarwal et al., 2024). The third direction keeps the verifiable outcome reward and folds the expert in beside it, as off-policy traces mixed into the rollout group (Yan et al., 2025), guidance restricted to critical decision points (Jiang et al., 2026b), verifiable subproblems carved from reference chains (Jiang et al., 2026a), or a dense reward recovered from demonstrations by inverse reinforcement learning (Fanconi et al., 2025). However, the expert always enters as one specific token sequence, so supervision rewards reproducing a single solution rather than the progress structure shared across the many valid paths a problem admits (Wang et al., 2023).

5 Conclusion

We introduced MILE, a milestone-guided reinforcement learning framework that derives dense supervision from multiple expert trajectories. Its core idea is to model reasoning progress through intermediate milestones characterized by future solvability, rather than relying on local step correctness or imitation of a specific reasoning trajectory. Experiments demonstrate that MILE provides effective progress-oriented supervision, improves mathematical reasoning performance, and achieves a better accuracy–efficiency trade-off. Across five benchmarks, MILE-Full improves over its GRPO backbone by 5.5 points and over the strongest process-supervision baseline by 3.4 points while remaining robust under constrained generation budgets.

AI USE STATEMENT

Generative AI tools are an original and non-standard component of the core methodology of this work, and we disclose their use as follows. Large language models are used to generate the expert reasoning trajectories that constitute MILE-Dataset (*Llama 4 Scout*, *Qwen3.5-122B-A10B*, *Qwen2.5-72B-Instruct*, and *gpt-oss-120b*), to segment raw traces into structured reasoning steps (*GPT-5.4*), and to produce the bounded continuations used for the future-solvability milestone annotation. Every generated trajectory is filtered by an automatic answer verifier, and the annotation protocol is described in full in Section 3.1 and Appendix C. The models under study are themselves language models, as stated throughout the paper.

Generative AI tools were additionally used for writing assistance, limited to grammar and phrasing polish of author-written text. They were not used for research ideation, for experimental design, or to generate any reported result. All AI-assisted content has been reviewed by the authors, and we take full responsibility for the final content of this work, including its text, claims, and released artifacts.

ETHICS STATEMENT

This work studies reinforcement learning for mathematical reasoning and does not involve human subjects, personally identifiable information, or sensitive user data. MILE-Dataset is derived from Big-Math (Albalak et al., 2025) and OpenR1-Math-220k[1], both of which are publicly released for research use; we retain only problem statements and reference final answers, and we regenerate all reasoning traces with publicly available teacher models under a single annotation protocol. To avoid contaminating the reported numbers, we remove every source problem that overlaps with the evaluation benchmarks used in Section 3.

Two residual risks are worth stating explicitly. First, the expert trajectories are model-generated, so the milestone annotations inherit whatever biases and errors the teacher models carry; we mitigate this by discarding trajectories whose final answer fails verification, but locally flawed reasoning that still reaches a correct answer can survive filtering. Second, methods that make reasoning models stronger and more token-efficient are dual-use in the ordinary sense that any capability improvement is: our contribution is confined to verifiable mathematical problem solving, and we release no model or data that targets a harmful application.

REPRODUCIBILITY STATEMENT

We have tried to make every component of MILE reproducible from the paper alone. The reward construction, milestone-extraction criterion, and two-stage GRPO schedule are specified in Section 2; the dataset pipeline, including teacher models, segmentation, filtering rules, and the continuation-budget annotation protocol, is specified in Section 3.1. The experimental protocol, backbones, benchmarks, and evaluation metrics are described in Section 3.

Numeric settings are collected in the appendix rather than scattered through the text: training hyperparameters in Table 4, reward and retrieval hyperparameters in Table 6, inference hyperparameters in Table 7, and compute accounting in Table 5. Appendix C additionally documents the non-numeric choices, including the Stage-1 local objective, the frozen encoder construction, and the milestone-annotation probe. All assumptions and complete proofs for the theoretical claims are given in Appendix E, and the diagnostic metrics reported in the ablations are defined in Appendix B. Table 3 shows the exact record format of MILE-Dataset. All reported results are Pass@1 averaged over 16 random seeds (Table 7) and reported as mean $\pm$ standard deviation. Anonymized source code for data construction, training, and evaluation is provided in the supplementary material.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. *arXiv preprint arXiv:2306.13649*, 2024. URL https://arxiv.org/abs/2306.13649.

[1] https://huggingface.co/datasets/open-r1/OpenR1-Math-220k

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models, 2025. URL https://arxiv.org/abs/2502.17387.

Richard Catrambone. The subgoal learning model: Creating better examples so that students can solve novel problems. *Journal of Experimental Psychology: General*, 127(4):355–376, 1998. doi: 10.1037/0096-3445.127.4.355.

Jianlong Chen, Daocheng Fu, Shengze Xu, Jiawei Chen, Yuan Feng, Yue Yang, Junchi Yan, Hongyuan Zha, and Renqiu Xia. Milestones over outcome: Unlocking geometric reasoning with sub-goal verifiable reward. *arXiv preprint arXiv:2601.05073*, 2026. URL https://arxiv.org/abs/2601.05073.

Wenxiang Chen, Wei He, Zhiheng Xi, Honglin Guo, Boyang Hong, Jiazheng Zhang, Nijun Li, Tao Gui, Yun Li, Qi Zhang, and Xuanjing Huang. Better process supervision with bi-directional rewarding signals. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 14471–14485, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.747. URL https://aclanthology.org/2025.findings-acl.747/.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. *arXiv preprint arXiv:2504.15275*, 2025. doi: 10.48550/arXiv.2504.15275. Accepted by NeurIPS 2025.

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025a. doi: 10.48550/arXiv.2502.01456.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025b. URL https://arxiv.org/abs/2505.22617.

Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025. URL https://arxiv.org/abs/2501.12599.

Claudio Fanconi, Nicolás Astorga, and Mihaela van der Schaar. Learning reasoning rewards from expert demonstrations with inverse reinforcement learning. *arXiv preprint arXiv:2510.01857*, 2025. URL https://arxiv.org/abs/2510.01857.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Haoran He, Yuxiao Ye, Qingpeng Cai, Chen Hu, Binxing Jiao, Daxin Jiang, and Ling Pan. Random policy valuation is enough for LLM reasoning with verifiable rewards. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ujLgLz6QQa.

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. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://arxiv.org/abs/2103.03874.

Xitai Jiang, Zihan Tang, Wenze Lin, Yang Yue, Shenzhi Wang, and Gao Huang. From reasoning chains to verifiable subproblems: Curriculum reinforcement learning enables credit assignment for LLM reasoning. *arXiv preprint arXiv:2605.22074*, 2026a. URL https://arxiv.org/abs/2605.22074.

Yuxuan Jiang and Francis Ferraro. Bridging reasoning trajectories in on-policy distillation via near-future guidance. *arXiv preprint arXiv:2606.00305*, 2026. URL https://arxiv.org/abs/2606.00305.

Zishang Jiang, Jinyi Han, Tingyun Li, Xinyi Wang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, Jiaqing Liang, and Yanghua Xiao. Selective expert guidance for effective and diverse exploration in reinforcement learning of LLMs. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/66c2e8fcd514b14be28b0e046f08f07b-Abstract-Conference.html.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://arxiv.org/abs/2410.01679.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025. URL https://arxiv.org/abs/2504.16828.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *Proceedings of the International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/26494b66ae1114d314673e25b4967288-Abstract-Conference.html.

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. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.20050.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025. URL https://arxiv.org/abs/2503.20783.

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. URL https://arxiv.org/abs/2406.06592.

Chengqi Lyu, Songyang Gao, Yuzhe Gu, Wenwei Zhang, Jianfei Gao, Kuikun Liu, Ziyi Wang, Shuaibin Li, Qian Zhao, Haian Huang, Weihan Cao, Jiangning Liu, Hongwei Liu, Junnan Liu, Songyang Zhang, Dahua Lin, and Kai Chen. Exploring the limit of outcome reward for learning mathematical reasoning. *arXiv preprint arXiv:2502.06781*, 2025. doi: 10.48550/arXiv.2502.06781.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, pp. 278–287, 1999.

Lujie Niu, Haochen Sun, Fangkun Zhao, Sheng Chen, Zimeng Bai, Jiawei Zhang, Caixia Yuan, and Xiaojie Wang. CoTD-PO: Chain-of-thought distillation with preference optimization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of*

the Association for Computational Linguistics: EMNLP 2025, pp. 19975–19986, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1087. URL https://aclanthology.org/2025.findings-emnlp.1087/.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. *arXiv preprint arXiv:2410.08146*, 2024. URL https://arxiv.org/abs/2410.08146.

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. doi: 10.48550/arXiv.2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*, 2024. doi: 10.48550/arXiv.2409.19256.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. PRMBench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*, 2025. URL https://arxiv.org/abs/2501.03124.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 17438–17457, 2026. doi: 10.18653/v1/2026.acl-long.792. URL https://aclanthology.org/2026.acl-long.792/.

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

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.acl-long.510/.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 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, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025. URL https://arxiv.org/abs/2504.14945.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. doi: 10.48550/arXiv.2412.15115. URL https://arxiv.org/abs/2412.15115.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026. URL https://arxiv.org/abs/2602.12275.

Bin Yu, Hang Yuan, Haotian Li, Xueyin Xu, Yuliang Wei, Bailing Wang, Weizhen Qi, and Kai Chen. Long-short chain-of-thought mixture supervised fine-tuning eliciting efficient reasoning in large language models. *arXiv preprint arXiv:2505.03469*, 2025a.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025b. URL https://arxiv.org/abs/2503.14476.

Danlong Yuan, Tian Xie, Shaohan Huang, Zhuocheng Gong, Huishuai Zhang, Chong Luo, Furu Wei, and Dongyan Zhao. Shorten after you're right: Lazy length penalties for reasoning RL. *arXiv preprint arXiv:2505.12284*, 2025. doi: 10.48550/arXiv.2505.12284.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024. URL https://arxiv.org/abs/2412.01981.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025. URL https://arxiv.org/abs/2504.13837.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025. URL https://arxiv.org/abs/2501.07301.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://arxiv.org/abs/2412.06559.

Yuhang Zhou, Yixin Cao, and Guangnan Ye. From correctness to utility: Gain-based prefix evaluation for LLM reasoning. *arXiv preprint arXiv:2606.07190*, 2026. URL https://arxiv.org/abs/2606.07190.

A Dataset Record Format

Table 3 shows a compact example in the same record format used by MILE-Dataset. For readability, we show one expert trajectory; full records contain multiple expert trajectories for the same problem. Red text indicates milestone positions selected by the progress-based criterion.

Table 3: A step-level MILE-Dataset demo. Bold font indicates milestone positions.

Field	Content
Problem	A rectangle has perimeter 30 and length 9. What is its width?
Reference answer	6
Trace 1, Step 1	Use the perimeter formula $2(L + W) = 30$.
Trace 1, Step 2	Substitute the length $L = 9$, giving $2(9 + W) = 30$.
Trace 1, Step 3	**Divide both sides by 2, obtaining $9 + W = 15$.**
Trace 1, Step 4	**Subtract 9 from both sides to obtain $W = 6$.**
Trace 1, Step 5	Report the final answer as $\boxed{6}$.
Milestone positions	**Trace 1, Steps 3 and 4.**

B Definitions of the Diagnostic Metrics

This appendix defines the diagnostic metrics used in the scaling and ablation experiments. These metrics are not additional training objectives; they are used only to interpret whether the learned policy follows the intended milestone retrieval behavior.

B.1 Retrieval Entropy

Retrieval entropy measures whether milestone retrieval uses multiple expert trajectories rather than collapsing to one trajectory. For each problem x with K verified expert trajectories, we use the main-text retrieval weights $w_{t,u}^{(k)}$, where k indexes the expert trajectory and u indexes the milestone within that trajectory. For every student prefix s_t produced in evaluation, the retrieval mass assigned to trajectory k is

$$c_t(k) = \sum_{u=0}^{U_k} w_{t,u}^{(k)}.$$

We then aggregate this mass over all retrieval queries for problem x:

$$C_x(k) = \sum_{t \in \mathcal{Q}_x} c_t(k),$$

where $\mathcal{Q}_x$ denotes the indices of all retrieval queries produced for problem x. The resulting trajectory-level retrieval distribution is

$$\rho_x(k) = \frac{C_x(k)}{\sum_{j=1}^{K} C_x(j)}.$$

We then compute the normalized entropy

$$\mathrm{Ent}(x; K) = \frac{-\sum_{k=1}^{K} \rho_x(k) \log \rho_x(k)}{\log K}.$$

The reported entropy for a given K is averaged over all evaluation problems:

$$\mathrm{Ent}(K) = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathrm{Ent}(x; K).$$

This metric lies in $[0, 1]$ for $K > 1$. A low value indicates that retrieval collapses to milestones from a small number of expert trajectories, whereas a high value indicates that the policy actively uses milestones from multiple expert trajectories.

B.2 Direction Alignment

Direction alignment measures whether a student transition moves through representation space in the same direction as the retrieved expert progress direction. For each generated reasoning step, let $z_t = E_\theta(s_t)$ and $z_{t+1} = E_\theta(s_{t+1})$ be the frozen-encoder representations before and after the student action. Following Section 2.3, the student displacement is

$$\Delta z_t = z_{t+1} - z_t.$$

Let $\bar{d}_t$ denote the posterior mean expert direction induced by the retrieval weights over the milestone pool, as defined in Section 2.3. The direction-alignment score for a single transition is

$$\mathrm{Align}_t = \cos(\Delta z_t, \bar{d}_t).$$

The reported direction alignment is the average over all non-empty reasoning transitions in the evaluation rollouts:

$$\mathrm{DirAlign} = \frac{1}{|\mathcal{Q}|} \sum_{(x,t) \in \mathcal{Q}} \mathrm{Align}_t,$$

where $\mathcal{Q}$ is the set of evaluated student transitions with nonzero representation displacement and nonzero retrieved expert direction. Higher values indicate stronger agreement between student updates and retrieved expert progress directions.

B.3 False Progress

False progress measures how often a model assigns high milestone-retrieval scores to prefixes that do not empirically improve future solvability. For each evaluated prefix s_i, let s_i^- denote its immediate predecessor prefix from the same reasoning trajectory. If s_i is the first prefix of a trajectory, s_i^- is the initial prefix containing only the problem prompt.

We estimate future solvability by sampling bounded continuations from a fixed evaluation policy. Let $\widehat{q}^{(B)}(x_i, s_i)$ be the fraction of sampled continuations from prefix s_i that reach the correct answer within B tokens. The empirical progress of s_i is

$$\widehat{\Delta}_i = \widehat{q}^{(B_{\text{fp}})}(x_i, s_i) - \widehat{q}^{(B_{\text{fp}})}(x_i, s_i^-),$$

where $B_{\text{fp}} = 1024$ in our experiments.

Let $h_i = \Phi(s_i)$ be the milestone-retrieval score assigned to prefix s_i, and let $\mathcal{H}_\rho(h)$ be the set of prefixes whose scores fall in the top ρ fraction:

$$\mathcal{H}_\rho(h) = \left\{ i : h_i \geq \text{Quantile}_{1-\rho}\left(\{h_j\}_{j=1}^M\right) \right\}.$$

We use $\rho = 0.1$, corresponding to the top 10% highest-scored prefixes. False progress is then defined as

$$\text{FalseProgress}(h) = 100 \times \frac{1}{|\mathcal{H}_\rho(h)|} \sum_{i \in \mathcal{H}_\rho(h)} \mathbb{1}[\widehat{\Delta}_i \leq 0].$$

A false-progress case is therefore a prefix that receives a high milestone score but does not improve the probability of reaching a correct answer under the bounded continuation protocol. Lower values are better.

C Additional Experimental Details

This appendix reports the implementation details used in the experiments and the record format of MILE-Dataset. To keep the tables readable, we place numeric hyperparameters in tables and describe non-numeric implementation choices in the surrounding text. Unless otherwise stated, all reinforcement learning methods use the same prompt format, answer verifier, rollout budget, decoding parameters, and optimization schedule.

C.1 Training Hyperparameters

The main backbone is Qwen3.5-4B. The trajectory-scaling study in Section 3.4 is run on three separate student backbones, Qwen2.5-7B-Instruct, Qwen3.5-4B, and Meta-Llama-3-8B-Instruct, and the budgeted evaluation in Section 3.5 uses Qwen3.5-4B. All policy training uses full-parameter fine-tuning with bf16 FSDP. The supervised warm start is optimized with AdamW and cosine learning-rate decay. Reinforcement learning uses GRPO, with rewards normalized by a group-relative z-score within each rollout group. The reference encoder is a frozen copy of the pre-reinforcement-learning student checkpoint, while the probe policy used for milestone annotation is described below. Experiments are run on NVIDIA H200 GPUs. Numeric training hyperparameters are summarized in Table 4.

In Stage 1, expert-prefix GRPO uses a local transition objective rather than the trajectory-level objective in Section 2.3. For an expert prefix s_t, the policy samples a group of candidate next-step continuations a_t, each defining s_{t+1}. We do not roll these candidates out to a final answer, so the verifier term $R_{\text{ver}}(\tau)$ and the final-answer gate are not used in Stage 1. The raw Stage-1 reward is

$$R^{(1)}(s_t, a_t) = \lambda_p r_t^{\text{prog}} + \lambda_d r_t^{\text{dir}},$$

and GRPO applies group-relative normalization across the candidate next steps sampled from the same expert prefix. Candidate continuations are capped at 512 new tokens and are treated as one local step: generation stops at EOS or the token cap, and if the sampled text contains multiple parsed reasoning steps, only the first generated step is used to form the transition. Stage 2 then switches to full rollouts from the original problem prompt and uses the trajectory-level verifier and answer-gated dense reward from Section 2.3.

C.2 Compute Accounting

We report compute in H200 GPU-hours, computed as wall-clock runtime multiplied by the number of allocated GPUs. Data construction includes teacher trajectory generation, trace segmentation, and bounded-continuation milestone annotation. Main-setting training includes the supervised warm start and the two GRPO stages for the primary Qwen3.5-4B run; it excludes final benchmark evaluation. Appendix D.5 additionally reports policy-training GPU-hours for the reinforcement learning baselines, alongside an outcome-only GRPO run extended to match MILE's policy-training budget, in the Qwen2.5-7B-Instruct setting.

Table 4: Numeric training hyperparameters. Values shared across methods are kept fixed unless explicitly stated.

Hyperparameter	Value
Supervised fine-tuning learning rate	2×10^{-5}
Supervised fine-tuning effective batch size	128 sequences
Supervised fine-tuning micro-batch size per GPU	1 sequence
Supervised fine-tuning gradient-accumulation steps	16
Supervised fine-tuning epochs	1
Maximum input length	4096 tokens
Maximum response length for expert-prefix GRPO	512 tokens
Maximum response length for trajectory GRPO	4096 tokens
Rollout group size	8 completions per prompt
Prompt batch size	64 prompts per GRPO update
Policy learning rate for expert-prefix GRPO	1×10^{-6}
Policy learning rate for trajectory GRPO	5×10^{-7}
Warmup ratio	0.05
Weight decay	0.01
Gradient clipping threshold	1.0
KL coefficient	0.0
Reinforcement learning epochs per stage	1
Number of GPUs	8

Table 5: Numeric compute accounting in H200 GPU-hours.

Quantity	Value
Data construction GPU-hours	1,184
Main-setting training GPU-hours	320

C.3 Milestone Annotation Probe

The probe policy used for milestone annotation is trajectory-specific rather than a separately trained verifier. For an expert trajectory generated by teacher model T, bounded continuations from prefixes of that trajectory are sampled from the same frozen teacher checkpoint T. Thus the probe architecture is one of the teacher architectures listed in Section 3.1: Llama 4 Scout, Qwen3.5-122B-A10B, Qwen2.5-72B-Instruct, or gpt-oss-120b. We do not train, fine-tune, or calibrate an additional probe model.

All probe continuations use the same continuation prompt template. The system prompt instructs the model to solve carefully and end with `Final Answer: <answer>`. The user prompt contains the problem statement, the current reasoning prefix, the required answer format, and the instruction to continue from the prefix until a final answer is produced. For budget b, the continuation is capped at b new tokens. Each nonzero budget uses three independent random seeds; the zero-token budget is evaluated without generation. A budget is marked successful if at least one of the three continuations reaches the verified correct final answer. The milestone-extraction footprint is included in the data construction GPU-hours in Table 5.

C.4 Reward and Retrieval Hyperparameters

Milestones are selected with the progress-based rule described in Section 2.2. The main setting uses $K^\star = 8$ verified expert trajectories per problem, and the scaling study evaluates every integer value from $K = 1$ to $K = 8$. Retrieval is performed by a cosine-softmax over all stored milestones for the same problem. The final reward uses the potential-shaping term, the direction-consistency term, and final-answer gating described in Section 2.3. Numeric reward and retrieval hyperparameters are summarized in Table 6.

The frozen encoder E_θ used for milestone retrieval is a decoder-only LLM encoder implemented by freezing the student checkpoint at the start of reinforcement learning. For each state, we serialize the problem statement and the current reasoning prefix with ordinary text delimiters, tokenize this text with the unchanged model tokenizer, and take the last hidden layer representation at the final non-padding token. We then L2-normalize this vector before cosine retrieval. We do not use mean pooling, add a learned separator token, or update the encoder during RL.

C.5 Inference Hyperparameters

Final evaluation reports Pass@1 averaged over 16 random seeds. The answer parser first looks for a final line in the form `Final Answer: <answer>`; if that line is absent, it falls back to the last non-empty line and canonicalizes boxed or LaTeX-formatted answers. The verifier uses

Table 6: Numeric reward and retrieval hyperparameters.

Hyperparameter	Value
Number of expert trajectories per problem in the main setting, $K^\star$	8
Values of K in the scaling study	$1, 2, \ldots, 8$
Maximum stored milestones per trajectory	8
Retrieval temperature	$\tau_r = 0.15$
Maximum retrieved milestones in the main setting	64
Potential-shaping weight	$\lambda_p = 1.0$
Progress clipping threshold	1.0
Direction-consistency weight	$\lambda_d = 0.25$
Final-answer gating residual coefficient	$\epsilon = 0.05$
Continuation budgets for milestone annotation	$0, 32, 64, 128, 256, 512, 1024, 2048$ tokens
Continuations per budget for milestone annotation	3
Nonzero probe continuations per evaluated prefix	21
Probe continuation temperature	0.6
Probe continuation top-p	0.95

normalized exact match together with symbolic equivalence checking via SymPy. Numeric inference hyperparameters are summarized in Table 7.

Table 7: Numeric inference hyperparameters for final evaluation and budgeted evaluation.

Hyperparameter	Value
Maximum new tokens for main evaluation	4096
Maximum new tokens for budgeted evaluation	$256, 512, 1024, 2048$
Number of random seeds	16
Samples per problem per seed	1

D Additional Results on Qwen2.5-7B-Instruct

Before moving to Qwen3.5-4B, we ran the full comparison and ablation suite on Qwen2.5-7B-Instruct (Yang et al., 2024). We report those results here as a second-backbone check: the base policy is substantially weaker, so absolute accuracy is far lower on every benchmark, but the ordering of methods and the direction of every ablation agree with the main setting. The experimental protocol is identical to Section 3 apart from the student checkpoint. This appendix also collects the supervision-, expert-budget-, and compute-matched controls of Appendices D.3–D.5, which were run on this backbone.

D.1 Main Comparison

Table 8 reports the comparison on Qwen2.5-7B-Instruct. MILE-Core reaches an overall accuracy of 50.08, ahead of outcome-only GRPO, Short-RL, and PRIME but still behind PURE and OREAL, so on this weaker backbone the milestone reward needs the final-answer gating to overtake the strongest process-aware baselines. MILE-Full reaches an overall accuracy of 53.30, improving over the best non-MILE baseline by 1.68 points and over the GRPO backbone by 5.50 points. On this weaker backbone MILE-Full does not uniformly dominate every column: it slightly trails OREAL on MATH500 while obtaining the best result on AMC23, AIME25, AIME26, and OlymMATH-100. This is the pattern reported in the main text for the harder competition benchmarks, where sparse final-answer rewards make intermediate credit assignment most valuable.

D.2 Ablations

Table 9 repeats the reward-design and milestone-construction ablations on Qwen2.5-7B-Instruct. Each component contributes to task performance or to one of the diagnostic measures, indicating that the gains of MILE do not come from a single heuristic but from the interaction of milestone construction, milestone reward, and answer gating.

The reward-side ablations indicate that direction consistency and answer gating address complementary failure modes. Removing the direction reward causes the largest drop in directional alignment, indicating weaker agreement between student updates and retrieved expert progress directions. Removing answer gating causes the largest increase in false progress, suggesting that models can be over-rewarded when global answer correctness is not enforced. Replacing progress-based milestone selection with uniformly selected milestones reduces overall accuracy, indicating that milestone quality matters more than fixed coverage along the trajectory. Similarly, removing milestone sparsification

Table 8: Main results on Qwen2.5-7B-Instruct (Yang et al., 2024).

Method	MATH500	AMC23	AIME25	AIME26	OlymMATH	Overall
Initialization						
Base model	$76.8_{\pm 0.7}$	$55.4_{\pm 0.9}$	$18.7_{\pm 1.1}$	$15.9_{\pm 1.3}$	$24.3_{\pm 1.0}$	$38.22_{\pm 1.1}$
SFT-$K^\star$ (Ouyang et al., 2022)	$82.4_{\pm 0.6}$	$60.7_{\pm 0.7}$	$23.6_{\pm 1.2}$	$20.8_{\pm 1.2}$	$29.8_{\pm 1.0}$	$43.46_{\pm 1.1}$
Reinforcement learning baselines						
GRPO (Shao et al., 2024)	$86.9_{\pm 0.6}$	$65.8_{\pm 0.8}$	$28.6_{\pm 1.1}$	$24.5_{\pm 1.0}$	$33.2_{\pm 1.1}$	$47.80_{\pm 0.9}$
Short-RL (Yuan et al., 2025)	$85.2_{\pm 0.7}$	$64.9_{\pm 0.8}$	$27.1_{\pm 1.0}$	$23.1_{\pm 1.2}$	$32.6_{\pm 1.0}$	$46.58_{\pm 0.9}$
OREAL (Lyu et al., 2025)	$\mathbf{94.0}_{\pm 0.4}$	$\underline{70.1_{\pm 0.8}}$	$\underline{30.8_{\pm 1.0}}$	$26.7_{\pm 1.2}$	$36.5_{\pm 1.0}$	$\underline{51.62_{\pm 1.0}}$
PRIME (Cui et al., 2025a)	$87.8_{\pm 0.6}$	$67.6_{\pm 0.7}$	$29.4_{\pm 1.0}$	$25.8_{\pm 1.2}$	$35.1_{\pm 1.0}$	$49.14_{\pm 0.9}$
PURE (Cheng et al., 2025)	$90.1_{\pm 0.5}$	$69.4_{\pm 0.8}$	$30.7_{\pm 1.0}$	$\underline{27.3_{\pm 1.1}}$	$\underline{36.9_{\pm 1.0}}$	$50.88_{\pm 0.9}$
Our methods						
MILE-Core	$88.3_{\pm 0.7}$	$68.9_{\pm 0.8}$	$30.4_{\pm 1.0}$	$27.1_{\pm 1.2}$	$35.7_{\pm 1.1}$	$50.08_{\pm 1.0}$
MILE-Full	$\underline{93.5_{\pm 0.6}}$	$\mathbf{71.9}_{\pm 0.7}$	$\mathbf{32.4}_{\pm 0.9}$	$\mathbf{29.6}_{\pm 1.1}$	$\mathbf{39.1}_{\pm 1.1}$	$\mathbf{53.30}_{\pm 0.9}$

by treating all steps as milestones increases retrieval entropy, yet still degrades task performance, so a larger or more diffuse milestone pool alone is insufficient for effective supervision. The reward is therefore most informative when the selected milestones correspond to future solvability.

Table 9: Ablations of MILE on Qwen2.5-7B-Instruct. Dir. Align. denotes the average cosine alignment between student displacement and retrieved expert direction. False Prog. (%) denotes the percentage of highly scored prefixes whose empirical progress is non-positive. Definitions of these metrics are provided in Appendix B.

Variant	Overall	False Prog. (%)	Dir. Align.	Re. Entropy
Full model				
MILE-Full	$\mathbf{53.30}_{\pm 0.9}$	$\mathbf{8.4}_{\pm 0.5}$	$\mathbf{0.741}_{\pm 0.014}$	$0.688_{\pm 0.015}$
Milestone construction				
MILE w/o progress-based selection	$49.63_{\pm 0.9}$	$13.5_{\pm 0.6}$	$0.677_{\pm 0.013}$	$0.611_{\pm 0.017}$
MILE w/o milestone sparsification	$49.29_{\pm 1.0}$	$14.8_{\pm 0.5}$	$0.652_{\pm 0.016}$	$0.743_{\pm 0.014}$
Reward design				
MILE w/o direction reward	$50.74_{\pm 0.9}$	$15.2_{\pm 0.5}$	$0.528_{\pm 0.015}$	$0.662_{\pm 0.011}$
MILE w/o answer gating ($\equiv$ MILE-Core)	$50.08_{\pm 1.0}$	$18.3_{\pm 0.4}$	$0.694_{\pm 0.015}$	$0.671_{\pm 0.016}$

D.3 Supervision-Matched Baselines

The comparison in Table 8 equalizes the SFT initialization, prompt format, answer verifier, and rollout budget, but it leaves a sharper question open: does MILE gain from *having* teacher supervision, or from *how* it converts that supervision into a reward? We therefore add two controls that receive the same teacher data as MILE and differ only in how they consume it.

The first is on-policy distillation (OPD; Agarwal et al., 2024), which trains the student to match the teacher's token-level output distribution on student-generated text. We use Qwen3.5-122B-A10B, one of the teacher models listed in Section 3.1, on the same student backbone. OPD receives strictly more teacher access than MILE: it requires the teacher's output probabilities and repeated teacher inference throughout student training, whereas MILE consumes generated text once and reuses the resulting milestone store.

The second control is a process reward model trained on exactly the annotations MILE retrieves over. For an expert prefix s, the regression target is the future-solvability score $p(s)$ of Section 2.2, computed from the same bounded continuations over the same budget set $\mathcal{B}$, and the PRM predicts this scalar directly from a prefix. During RL, the change in predicted value between consecutive student states supplies the dense progress reward in place of MILE's retrieval potential. This baseline is thus the parametric counterpart of MILE: identical labels, identical training problems, a learned value function instead of non-parametric retrieval over a milestone store.

Table 10 reports both controls. The matched PRM improves over outcome-only GRPO by only 0.74 points overall and in fact trails it slightly on AIME25, far short of the 5.50 points contributed by MILE-Full. Access to the future-solvability annotations is therefore not by itself what produces the gain; how those annotations are turned into a reward matters more than whether they are available. OPD is the stronger of the two controls at 51.06 overall, confirming that token-level teacher supervision is genuinely useful, yet MILE-Full remains ahead on every one of the five benchmarks and by 2.24 points overall, while requiring neither teacher logits nor teacher inference during RL.

Table 10: Supervision-matched baselines on Qwen2.5-7B-Instruct. OPD receives teacher logits from Qwen3.5-122B-A10B; the PRM is trained on the same future-solvability annotations that MILE retrieves over. All rows start from the same SFT-$K^\star$ checkpoint and share the prompt format, answer verifier, and rollout budget of Table 8. Accuracy is Pass@1 over 16 seeds; the two control runs were evaluated for accuracy only, so standard deviations are reported for the reference rows in Table 8 rather than repeated here.

Method	MATH500	AMC23	AIME25	AIME26	OlymMATH	Overall
Same expert data, different consumption						
GRPO (Shao et al., 2024)	86.9	65.8	28.6	24.5	33.2	47.80
PRM on MILE annotations	87.9	66.9	28.3	25.2	34.4	48.54
OPD (Agarwal et al., 2024)	91.2	69.5	30.9	27.0	36.7	51.06
Our method						
MILE-Full	**93.5**	**71.9**	**32.4**	**29.6**	**39.1**	**53.30**

D.4 Alternative Uses of the Same Expert Budget

Figure 2 compares SFT-K against MILE-K at every K, which isolates milestone-guided RL from simply fine-tuning on more trajectories. Two further alternatives use the same expert and annotation budget without the milestone reward. *Additional SFT* continues supervised fine-tuning on the same verified trajectories past the SFT-$K^\star$ warm start until the optimization budget is consumed. *Mixed-policy expert usage* interleaves expert trajectory segments with on-policy rollouts during optimization, so expert text enters the gradient directly rather than through a reward on student states.

Table 11 reports both. Additional SFT falls below outcome-only GRPO: past the warm start, further imitation of the same trajectories does not substitute for reinforcement learning, which is the same ordering the SFT-K curves show in Figure 2. Mixed-policy usage is the stronger alternative and does improve on GRPO, but still trails MILE-Full by 2.4 and 2.9 points. The same expert trajectories therefore yield more when converted into milestone supervision than when injected into the optimization directly.

Table 11: Alternative uses of the same expert budget on Qwen2.5-7B-Instruct. All rows start from the same SFT-$K^\star$ checkpoint and use the same $K^\star = 8$ verified trajectories per problem under a matched annotation and optimization budget.

Use of the same expert budget	AIME25	AIME26
Outcome-only GRPO (Shao et al., 2024)	28.6	24.5
Additional SFT	25.6	22.2
Mixed-policy expert usage	30.0	26.7
MILE-Full	**32.4**	**29.6**

D.5 Policy-Training Compute

Table 5 reports MILE's one-time offline construction cost. A separate question is whether the online gain simply reflects more optimization. To test this we continued outcome-only GRPO from the same SFT checkpoint on the same 22,531 problems until its policy-training GPU-hours matched MILE's two-stage schedule. Table 12 reports the result together with the policy-training cost of the two strongest process-supervised baselines.

Extending GRPO from 390 to 416 GPU-hours leaves both AIME scores unchanged, indicating that outcome-only optimization has largely plateaued by the end of its original schedule; the gap to MILE-Full is therefore not explained by optimization time. MILE-Full also completes its two stages in fewer policy-training GPU-hours than either PURE (470) or PRIME (500) while scoring higher on both benchmarks, so MILE's offline construction cost does not translate into a heavier online training budget. The figures in this table are measured in the Qwen2.5-7B-Instruct setting of this appendix and are not comparable to the Qwen3.5-4B main-setting figure in Table 5, which uses a smaller backbone.

Table 12: Policy-training compute on Qwen2.5-7B-Instruct, in H200 GPU-hours. Offline milestone construction is excluded and reported separately in Table 5.

Method	Policy-training GPU-hours	AIME25	AIME26
GRPO (Shao et al., 2024)	390	28.6	24.5
GRPO (compute-matched)	416	28.6	24.5
PRIME (Cui et al., 2025a)	500	29.4	25.8
PURE (Cheng et al., 2025)	470	30.7	27.3
MILE-Full	416	**32.4**	**29.6**

E THEORETICAL ANALYSIS OF THE MILESTONE REWARD

This appendix provides an idealized analysis of the milestone reward used by MILE. The goal is not to prove convergence of GRPO, but to clarify what the retrieval-based reward estimates and why multiple expert trajectories make the reward less path-imitation oriented. The analysis follows the notation of Section 2.

There are three main points. First, with a single expert trajectory, milestone retrieval has support only on one teacher path, so the induced local reward can degenerate into representation-level path following. Second, with many verified expert trajectories and sufficient coverage, the retrieval potential is a non-parametric estimator of the same future-solvability value used to annotate milestones. Third, the potential-difference part of the reward is mathematically controlled by a bounded potential; however, a strict anti-length-hacking guarantee for the complete practical reward requires additional safe-shaping conditions, since the implemented direction term is additive.

E.1 FORMAL SETUP

Fix a problem x. Let

$$\mathcal{T}_x = \{\tau^{(1)}, \dots, \tau^{(K)}\}$$

be the set of verified expert trajectories for x. The k-th trajectory is written as

$$\tau^{(k)} = \big(s_0^{(k)}, a_0^{(k)}, s_1^{(k)}, \dots, s_{T_k}^{(k)}\big),$$

where $s_t^{(k)}$ is the reasoning prefix after t expert steps.

Future-solvability annotation. Let $\mathcal{B}$ be the set of continuation-token budgets. For a prefix state s, a probe policy π_{probe} is asked to complete the solution under each budget $b \in \mathcal{B}$. To match the annotation protocol used in the dataset, allow $J \geq 1$ independent probe continuations per budget. Let

$$C_{b,j}(s) \in \{0, 1\}$$

be the indicator that the j-th bounded continuation under budget b produces a verified correct final answer. Both the main text and the dataset annotation (Section 2.2) use $J = 3$, where a budget is marked successful when at least one continuation is correct; a general J is kept only to make the dependence on the repetition rule explicit. We write this empirical budget success as

$$\widehat{q}_{b,J}(s) = 1 - \prod_{j=1}^{J} \big(1 - C_{b,j}(s)\big),$$

and the empirical future-solvability score as

$$\widehat{p}_{\mathcal{B},J}(s) = \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \widehat{q}_{b,J}(s).$$

For notational simplicity, the main paper writes this quantity as $p(s)$. In this appendix, we keep the hat when emphasizing that the quantity is estimated by finite probe continuations.

The corresponding population target induced by the same annotation protocol is

$$V_x^\star(s) = \mathbb{E}[\widehat{p}_{\mathcal{B},J}(s) \mid x, s, \pi_{\text{probe}}].$$

This target is a budgeted future-solvability value: it measures how likely the probe policy is to finish the solution correctly from prefix s under the specified budget set and repetition rule.

Representation-level value. MILE performs retrieval in the frozen initial-student representation space. Let

$$z = E_\theta(s) \in \mathbb{R}^d$$

be the frozen representation of prefix s. The representation-level oracle value is

$$v_x^\star(z) = \mathbb{E}[V_x^\star(s) \mid E_\theta(s) = z,\ x].$$

This is the value that an ideal process reward model based on the same future-solvability criterion would estimate in representation space.

Milestone selection and labels. For each expert trajectory $\tau^{(k)}$, milestones are selected using the same rule as Section 2.2. Initialize $m_0^{(k)} = 0$. Given $m_u^{(k)}$, define

$$m_{u+1}^{(k)} = \min\left\{ t > m_u^{(k)} : \widehat{p}_{\mathcal{B},J}\left(s_t^{(k)}\right) - \widehat{p}_{\mathcal{B},J}\left(s_{m_u^{(k)}}^{(k)}\right) \geq \frac{1}{|\mathcal{B}|} \right\},$$

with the sequence terminating if the set is empty. Let U_k be the number of non-initial milestone advances in trajectory k, so $u = 0, \ldots, U_k$.

The key point is that the milestone label is the actual future-solvability score:

$$p_u^{(k)} = \widehat{p}_{\mathcal{B},J}\left(s_{m_u^{(k)}}^{(k)}\right),$$

not the normalized milestone index u/U_k. The index u records milestone order, whereas $p_u^{(k)}$ records the estimated solvability of the corresponding prefix.

Define the milestone representation

$$z_u^{(k)} = E_\theta\left(s_{m_u^{(k)}}^{(k)}\right).$$

Let the flattened milestone index set be

$$\mathcal{J}_{K,x} = \{(k,u) : 1 \leq k \leq K,\ 0 \leq u \leq U_k\}, \qquad N_K = |\mathcal{J}_{K,x}|.$$

Retrieval potential. For an arbitrary representation z, MILE assigns soft retrieval weights over all milestones of the same problem:

$$w_u^{(k)}(z) = \frac{\exp\left(\mathrm{sim}(z, z_u^{(k)})/\tau_r\right)}{\sum_{(k',u')\in\mathcal{J}_{K,x}} \exp\left(\mathrm{sim}(z, z_{u'}^{(k')})/\tau_r\right)}.$$

The retrieval potential is

$$\Phi_K(z) = \sum_{(k,u)\in\mathcal{J}_{K,x}} w_u^{(k)}(z)\, p_u^{(k)}.$$

The state-level potential in the main text is $\Phi(s) = \Phi_K(E_\theta(s))$. Since the weights form a simplex and $p_u^{(k)} \in [0,1]$,

$$0 \leq \Phi_K(z) \leq 1.$$

This boundedness is the basic property behind the potential-shaping interpretation.

Direction field. For nonterminal milestones, define

$$\mathcal{I}_{K,x} = \{(k,u) : 1 \leq k \leq K,\ 0 \leq u < U_k\}.$$

The expert forward direction is

$$d_u^{(k)} = \frac{z_{u+1}^{(k)} - z_u^{(k)}}{\left\| z_{u+1}^{(k)} - z_u^{(k)} \right\|_2}.$$

For z, the nonterminal retrieval weights are renormalized as

$$\widehat{w}_u^{(k)}(z) = \frac{w_u^{(k)}(z)}{\sum_{(k',u')\in\mathcal{I}_{K,x}} w_{u'}^{(k')}(z)}, \qquad (k,u) \in \mathcal{I}_{K,x},$$

whenever the denominator is nonzero. The posterior mean direction is

$$\bar{d}_K(z) = \sum_{(k,u)\in\mathcal{I}_{K,x}} \widehat{w}_u^{(k)}(z)\, d_u^{(k)}.$$

For a student transition $s_t \to s_{t+1}$, with $z_t = E_\theta(s_t)$ and $z_{t+1} = E_\theta(s_{t+1})$, the implemented rewards are

$$r_t^{\mathrm{prog}} = \frac{\Phi_K(z_{t+1}) - \Phi_K(z_t)}{(1 + \mathrm{Tok}(a_t))^\gamma},$$

and

$$r_t^{\mathrm{dir}} = \cos\left(z_{t+1} - z_t,\ \bar{d}_K(z_t)\right).$$

E.2 Single-Trajectory Support Collapse

We first formalize the limitation of using a single expert trajectory.

Theorem E.1 (Single-path retrieval collapse)**.** *Assume $K = 1$. Write $z_u = z_u^{(1)}$ and $p_u = p_u^{(1)}$. Suppose each query point has a unique nearest milestone under the retrieval similarity. In the hard-retrieval limit $\tau_r \to 0$,*

$$\Phi_1(z) \longrightarrow p_{\nu(z)},$$

where

$$\nu(z) = \arg \max_{0 \le u \le U_1} \mathrm{sim}(z, z_u).$$

Therefore, for a transition $z \to z'$,

$$r^{\mathrm{prog}}(z, z') \longrightarrow \frac{p_{\nu(z')} - p_{\nu(z)}}{(1 + \mathrm{Tok}(a))^{\gamma}}.$$

Proof. The softmax weights concentrate on the unique maximizer of $\mathrm{sim}(z, z_u)$ as $\tau_r \to 0$. Hence all weight is assigned to the nearest milestone $z_{\nu(z)}$, and the potential converges to its label $p_{\nu(z)}$. Substituting this limit into the progress-reward definition gives the transition reward above. □

Interpretation. With one expert trajectory, positive progress reward is available only by moving toward regions assigned to later milestones of that single trajectory. Similarly, near a nonterminal milestone z_u, the direction field collapses to the single local direction

$$d_u = \frac{z_{u+1} - z_u}{\|z_{u+1} - z_u\|_2}.$$

Thus the reward need not imitate the teacher's exact tokens, but its support is restricted to one teacher path in the frozen representation space. If a correct alternative solution path is far from this path, it may receive little milestone reward despite being valid. This explains why a single expert trajectory can behave like representation-level step imitation, and motivates the multi-trajectory milestone pool.

A stronger local imitation statement follows under an identifiability margin. If, from a teacher milestone prefix, the expert successor is the only reachable successor whose representation enters the next milestone region and has maximal alignment with d_u, then the potential-plus-direction local score is maximized by the expert successor. This is not a property desired by MILE; rather, it describes the degeneration that multi-trajectory retrieval is designed to avoid.

E.3 Multi-Trajectory Consistency of the Retrieval Potential

We next show that, under standard non-parametric assumptions, the retrieval potential estimates the future-solvability value $v_x^\star$.

The main implementation uses cosine-softmax retrieval on normalized hidden-state embeddings. For unit-normalized embeddings,

$$\cos(z, z_i) = 1 - \frac{1}{2}\|z - z_i\|_2^2,$$

so

$$\exp(\cos(z, z_i)/\tau_r)$$

is equivalent, up to a query-independent multiplicative constant that cancels in the softmax, to a Gaussian kernel with bandwidth $h^2 = \tau_r$. Thus the consistency statement below can be read as an idealized local-kernel version of the implemented cosine-softmax retrieval. In experiments, τ_r is fixed; the theorem instead considers the usual asymptotic regime in which the effective bandwidth decreases as the milestone pool grows.

Assumption E.2 (Selected-milestone sampling)**.** For a fixed problem x, the selected milestone pairs

$$\{(Z_i, Y_i)\}_{i=1}^{N_K} = \left\{(z_u^{(k)}, p_u^{(k)})\right\}_{(k,u) \in \mathcal{J}_{K,x}}$$

are samples from a selected-milestone distribution μ_x^M supported on a compact set $\mathcal{Z}_x^M$. The density of μ_x^M on this support is bounded above and below with respect to a d_{eff}-dimensional volume measure. The value $v_x^\star$ is Lipschitz on $\mathcal{Z}_x^M$.

This assumption is about the distribution after applying the progress-based milestone selection rule. Milestone selection changes where samples are placed, but it does not change the meaning of their labels: each label is still the future-solvability score of that selected prefix.

Assumption E.3 (Bounded label noise)**.** The milestone labels satisfy

$$Y_i = v_x^\star(Z_i) + \xi_i + b_{\mathrm{ann}}(Z_i),$$

where

$$\mathbb{E}[\xi_i \mid Z_i] = 0, \qquad |\xi_i| \le \sigma, \qquad |b_{\mathrm{ann}}(z)| \le \alpha_{\mathrm{ann}}.$$

The term b_{ann} captures systematic mismatch between the finite annotation protocol and the idealized value being analyzed. If $v_x^\star$ is defined as the exact population target induced by the same budget set and repetition rule, then $\alpha_{\mathrm{ann}} = 0$. If instead $v_x^\star$ is interpreted as an unbounded or single-sample completion probability, then α_{ann} measures the bias introduced by finite budgets, finite repetitions, and the any-correct budget aggregation rule.

Assumption E.4 (Local-kernel bandwidth)**.** Let $h = h_{N_K}$ denote the effective retrieval bandwidth. Assume

$$h \to 0, \qquad N_K h^{d_{\mathrm{eff}}} \to \infty.$$

For uniform convergence, assume additionally the usual logarithmic strengthening

$$N_K h^{d_{\mathrm{eff}}} / \log N_K \to \infty.$$

Theorem E.5 (Consistency of the retrieval potential)**.** *Under Assumptions E.2–E.4,*

$$\sup_{z \in \mathcal{Z}_x^M} |\Phi_K(z) - v_x^\star(z)| = O(h) + O_p\left(\sqrt{\frac{\log N_K}{N_K h^{d_{\mathrm{eff}}}}}\right) + O(\alpha_{\mathrm{ann}}).$$

In particular, if $\alpha_{\mathrm{ann}} = 0$, then

$$\sup_{z \in \mathcal{Z}_x^M} |\Phi_K(z) - v_x^\star(z)| \xrightarrow{p} 0.$$

Proof. Under the local-kernel view, Φ_K is the Nadaraya–Watson estimator for the regression function

$$m_x(z) = \mathbb{E}[Y_i \mid Z_i = z].$$

By Assumption E.3,

$$m_x(z) = v_x^\star(z) + b_{\mathrm{ann}}(z).$$

The estimation error decomposes as

$$\Phi_K(z) - v_x^\star(z) = \underbrace{\big(\Phi_K(z) - m_x(z)\big)}_{\text{kernel estimation error}} + \underbrace{b_{\mathrm{ann}}(z)}_{\text{annotation bias}}.$$

The kernel estimation error has two standard components. Since $v_x^\star$ is Lipschitz, the local smoothing bias is $O(h)$. Since labels are bounded and the effective number of samples in a radius-h neighborhood is of order $N_K h^{d_{\mathrm{eff}}}$, the stochastic term is

$$O_p\left(\sqrt{\frac{\log N_K}{N_K h^{d_{\mathrm{eff}}}}}\right)$$

uniformly over the compact support, using a covering argument. Adding the annotation-bias term gives the stated bound. □

Corollary E.6 (Consistency of the progress reward)**.** *Define the oracle progress reward*

$$r_t^{\star,\mathrm{prog}} = \frac{v_x^\star(z_{t+1}) - v_x^\star(z_t)}{(1 + \mathrm{Tok}(a_t))^\gamma}.$$

Then, for any transition whose endpoints lie in $\mathcal{Z}_x^M$,

$$\left|r_t^{\mathrm{prog}} - r_t^{\star,\mathrm{prog}}\right| \le 2 \sup_{z \in \mathcal{Z}_x^M} |\Phi_K(z) - v_x^\star(z)|.$$

Therefore r_t^{prog} converges in probability to $r_t^{\star,\mathrm{prog}}$ whenever the retrieval potential converges uniformly.

Proof. Using the reward definitions,

$$\begin{aligned}\left|r_t^{\text{prog}} - r_t^{\star,\text{prog}}\right| &= \frac{\left|\left(\Phi_K(z_{t+1}) - \Phi_K(z_t)\right) - \left(v_x^\star(z_{t+1}) - v_x^\star(z_t)\right)\right|}{(1+\text{Tok}(a_t))^\gamma} \\ &\le \left|\Phi_K(z_{t+1}) - v_x^\star(z_{t+1})\right| + \left|\Phi_K(z_t) - v_x^\star(z_t)\right| \\ &\le 2 \sup_{z \in \mathcal{Z}_x^M} \left|\Phi_K(z) - v_x^\star(z)\right|,\end{aligned}$$

because $(1+\text{Tok}(a_t))^\gamma \ge 1$. □

Interpretation. The theorem explains the role of multiple expert trajectories. The selected milestone distribution μ_x^M is the support on which the non-parametric estimator can operate. With few expert trajectories, this support may cover only a small part of the solution manifold. As K increases, the pool contains milestones from more solution variants, which reduces the chance that a valid student state is far from all expert progress states. The estimator then becomes less tied to any single reference path.

E.4 Non-Parametric Interpretation of the Direction Reward

The direction term has a related interpretation as a non-parametric estimate of an expert progress vector field.

Let the nonterminal milestone triples be

$$\{(Z_i, D_i)\}_{i=1}^{M_K} = \left\{(z_u^{(k)}, d_u^{(k)})\right\}_{(k,u)\in\mathcal{I}_{K,x}}.$$

Define the conditional mean direction

$$g_x(z) = \mathbb{E}[D_i \mid Z_i = z].$$

Assumption E.7 (Direction regularity)**.** The nonterminal milestone distribution satisfies the same compact-support and bandwidth conditions as Assumptions E.2 and E.4. The conditional mean $g_x(z)$ is Lipschitz on the nonterminal support, and $\|D_i\|_2 = 1$.

Theorem E.8 (Consistency of the posterior mean direction)**.** *Under Assumption E.7,*

$$\sup_z \left\|\bar{d}_K(z) - g_x(z)\right\|_2 = O(h) + O_p\left(\sqrt{\frac{\log M_K}{M_K h^{d_{\text{eff}}}}}\right),$$

on regions where the nonterminal retrieval denominator is bounded away from zero.

Proof. The posterior mean direction is a kernel regression estimator with vector-valued bounded labels D_i. Applying the scalar kernel-regression argument componentwise gives the same bias and stochastic terms. The denominator condition ensures that the nonterminal renormalization is stable. □

Interpretation. If expert trajectories locally move in directions that increase future solvability, then $g_x(z)$ estimates an average local progress direction. In the special case where

$$D_i \approx \frac{\nabla v_x^\star(Z_i)}{\|\nabla v_x^\star(Z_i)\|_2},$$

the direction reward encourages student transitions to align with the local gradient of future solvability. The theorem does not require this gradient interpretation; it only shows that the implemented direction field estimates the average forward direction of nearby expert milestone transitions.

E.5 Multi-Path Coverage

The consistency theorem is conditional on coverage of the relevant student states by the selected milestone support. The following simple calculation illustrates why increasing the number of expert trajectories helps.

Suppose the solution manifold for a problem decomposes into C semantic modes $\mathcal{Z}_1, \ldots, \mathcal{Z}_C$. Let ρ_c be the probability that a randomly sampled verified expert trajectory contributes at least one selected

milestone in mode c. If K expert trajectories are sampled independently, then the probability that mode c is represented in the milestone pool is

$$1 - (1 - \rho_c)^K.$$

Thus the probability of missing a valid solution mode decreases exponentially in K. This does not imply that arbitrary large K is always useful: once the common modes are covered, additional trajectories may become redundant. It does explain why a multi-trajectory pool can be more path-robust than a single reference trajectory.

E.6 Replication Invariance and Trajectory Diversity

The coverage argument above explains why a multi-trajectory pool *can* be more path-robust. The following invariance property gives the converse statement: under soft retrieval, enlarging the pool with copies of a trajectory that is already present changes nothing at all. Trajectory count alone therefore cannot account for the scaling behavior reported in Section 3.4.

Replicated pools. Fix a problem x and a single verified trajectory $\tau^{(1)}$ with milestone indices $\{(1,u)\}_{u=0}^{U_1}$, embeddings $z_u^{(1)}$, and progress labels $p_u^{(1)}$. For $n \geq 1$, let the n-fold replicated pool contain n indexed copies of $\tau^{(1)}$, so that

$$z_u^{(k)} = z_u^{(1)}, \qquad p_u^{(k)} = p_u^{(1)}, \qquad k = 1, \ldots, n, \quad 0 \leq u \leq U_1.$$

This pool has n times as many entries as the single-trajectory pool, and it is exactly what a $K = n$ run would produce if the n sampled trajectories happened to be identical.

Proposition E.9 (Replication invariance)**.** *Let $\Phi_{\times n}$ and $\bar{d}_{\times n}$ denote the retrieval potential and the posterior mean direction computed on the n-fold replicated pool. Then for every representation z,*

$$\Phi_{\times n}(z) = \Phi_1(z), \qquad \bar{d}_{\times n}(z) = \bar{d}_1(z).$$

Consequently r_t^{prog}, r_t^{dir}, the trajectory reward $R(\tau)$, and the resulting GRPO objective are all unchanged.

Proof. Write $\ell_u = \exp\Big(\mathrm{sim}(z, z_u^{(1)})/\tau_r\Big)$ and $S = \sum_u \ell_u$. Every copy contributes the same logits, so the retrieval normalizer becomes nS and

$$w_u^{(k)}(z) = \frac{\ell_u}{nS} = \frac{1}{n}\, w_u^{(1)}(z), \qquad k = 1, \ldots, n.$$

Each individual weight is divided by n, but there are now n copies of every milestone, so summing over the pool recovers the original potential:

$$\Phi_{\times n}(z) = \sum_{k=1}^{n} \sum_u \frac{1}{n}\, w_u^{(1)}(z)\, p_u^{(1)} = \sum_u w_u^{(1)}(z)\, p_u^{(1)} = \Phi_1(z).$$

The nonterminal index set $\mathcal{I}_{K,x}$ is replicated n-fold as well, so the renormalized weights satisfy $\widehat{w}_u^{(k)}(z) = \frac{1}{n}\widehat{w}_u^{(1)}(z)$; together with $d_u^{(k)} = d_u^{(1)}$ the same cancellation gives

$$\bar{d}_{\times n}(z) = \sum_{k=1}^{n} \sum_{u<U_1} \frac{1}{n}\, \widehat{w}_u^{(1)}(z)\, d_u^{(1)} = \bar{d}_1(z).$$

Both milestone rewards depend on the pool only through Φ and $\bar{d}$, which completes the proof. □

Near-duplicate pools. Exact replication is an idealization; the relevant question is what happens when the K sampled trajectories are merely similar. Assume unit-norm embeddings, that the trajectories share a common milestone count U, and that after aligning milestone indices

$$\Big\| z_u^{(k)} - z_u^{(1)} \Big\|_2 \leq \varepsilon, \qquad \Big| p_u^{(k)} - p_u^{(1)} \Big| \leq \varepsilon_p \qquad \text{for all } k, u.$$

Comparing this pool with the K-fold replication of $\tau^{(1)}$ puts both on a common index set. For unit-norm embeddings, cosine similarity obeys $|\,\mathrm{sim}(z,a) - \mathrm{sim}(z,b)| \leq \|a-b\|_2$ by Cauchy–Schwarz, so the retrieval logits of the two pools differ by at most $\eta = \varepsilon/\tau_r$ entrywise. For softmax weights whose logits differ by at most η in ℓ_∞, each ratio w_i/w_i' lies in $\big[e^{-2\eta}, e^{2\eta}\big]$, hence

$$\|w - w'\|_1 \leq e^{2\eta} - 1.$$

Corollary E.10 (Stability under near-duplication)**.** *Under the above conditions, for every representation z,*

$$|\Phi_K(z) - \Phi_1(z)| \le \underbrace{e^{2\varepsilon/\tau_r} - 1}_{\text{geometry}} + \underbrace{\varepsilon_p}_{\text{labels}}.$$

Proof. Let w and w' be the retrieval weights of the two pools on the common index set. Since $p_u^{(k)} \in [0,1]$,

$$|\Phi_K(z) - \Phi_1(z)| \le \left|\sum_i (w_i - w_i')\, p_i\right| + \left|\sum_i w_i'\,(p_i - p_i')\right| \le \|w - w'\|_1 + \varepsilon_p,$$

and Proposition E.9 identifies the second pool's potential with Φ_1. □

Interpretation. The bound is uniform in z and vanishes as $\varepsilon, \varepsilon_p \to 0$: a pool of near-duplicate trajectories induces essentially the same progress reward as a single trajectory, however many trajectories it contains. Reading the bound in the other direction, if a K-trajectory pool changes the potential by at least $\delta > \varepsilon_p$ at some state, then the pool cannot be more homogeneous than

$$\varepsilon \ge \frac{\tau_r}{2}\,\log(1 + \delta - \varepsilon_p)\,.$$

This separates trajectory count from trajectory diversity: only the latter can move the reward. The monotone improvements from $K = 1$ to $K = 8$ in Figure 2 are therefore evidence that additional verified trajectories contribute milestones that are genuinely distinct in representation space, rather than additional probability mass on the same reasoning path. It also clarifies the role of the retrieval entropy reported alongside the scaling curves: entropy grows with K only when the added milestones are separated in the encoder geometry, since replicated milestones would leave the induced reward, and hence the effective retrieval distribution over distinct states, unchanged.

E.7 Length Properties of the Potential Reward

We now clarify what is and is not guaranteed by the potential-based reward.

Let

$$\delta_t = \Phi_K(z_{t+1}) - \Phi_K(z_t).$$

Because $0 \le \Phi_K(z) \le 1$, the unnormalized potential differences telescope:

$$\sum_{t=0}^{T-1} \delta_t = \Phi_K(z_T) - \Phi_K(z_0) \in [-1, 1].$$

Lemma E.11 (Telescoping bound for the potential difference)**.** *For any trajectory,*

$$\sum_{t=0}^{T-1} \delta_t \le 1.$$

If $z_T = z_0$, then

$$\sum_{t=0}^{T-1} \delta_t = 0.$$

Proof. This follows directly from telescoping and the boundedness of Φ_K. □

The implemented progress reward is length-normalized:

$$r_t^{\text{prog}} = \frac{\delta_t}{(1 + \text{Tok}(a_t))^{\gamma}}.$$

This normalization has the intended local effect: for a fixed positive progress $\delta_t > 0$, the reward is non-increasing in the number of generated tokens, and is strictly decreasing when $\gamma > 0$. Therefore, for the same semantic progress, a longer action receives no larger progress reward.

However, the length-normalized reward is not an exact telescoping sum when negative progress is possible. If $\delta_t < 0$, dividing by $(1 + \text{Tok}(a_t))^{\gamma}$ weakens the penalty for regressive steps. Thus a

strict length-independent anti-hacking theorem does not hold for arbitrary sign-changing cycles under the implemented r_t^{prog} alone.

A positive statement still holds for monotone-progress trajectories.

Proposition E.12 (Bound under monotone potential progress)**.** *If $\delta_t \geq 0$ for all t, then*

$$\sum_{t=0}^{T-1} r_t^{\text{prog}} \leq 1.$$

Proof. Since $(1 + \text{Tok}(a_t))^\gamma \geq 1$,

$$r_t^{\text{prog}} \leq \delta_t$$

whenever $\delta_t \geq 0$. Summing over t and applying Lemma E.11 gives

$$\sum_t r_t^{\text{prog}} \leq \sum_t \delta_t \leq 1. \qquad \square$$

Why a stronger statement needs safe shaping. Consider a two-step representation cycle with $\delta_1 = a > 0$ and $\delta_2 = -a$. The unnormalized potential differences sum to zero. But if the positive step is short and the negative step is very long, then

$$\frac{a}{(1 + \text{Tok}(a_1))^\gamma} - \frac{a}{(1 + \text{Tok}(a_2))^\gamma}$$

can be positive. Repeating such a cycle could accumulate positive normalized reward in principle. This is why the practical implementation also uses answer gating and a finite generation budget, and why a strict theoretical anti-length-hacking guarantee requires a safer shaping form.

One sufficient safe variant is the following:

$$r_t^{\text{safe}} = (\lambda_p + \lambda_d[\text{cost}_t]_+) \frac{[\delta_t]_+}{(1 + \text{Tok}(a_t))^\gamma} - (\lambda_p + \lambda_d)[-\delta_t]_+ - \eta \, \text{Tok}(a_t),$$

where

$$[\delta]_+ = \max(\delta, 0), \qquad [-\delta]_+ = \max(-\delta, 0), \qquad \text{cost}_t = \cos(z_{t+1} - z_t, \bar{d}_K(z_t)),$$

and $\lambda_p, \lambda_d, \eta \geq 0$.

Theorem E.13 (Anti-length-hacking under safe shaping)**.** *For the safe process reward above,*

$$\sum_{t=0}^{T-1} r_t^{\text{safe}} \leq \lambda_p + \lambda_d.$$

If $z_T = z_0$, then

$$\sum_{t=0}^{T-1} r_t^{\text{safe}} \leq 0,$$

and the inequality is strict for any nonempty cycle when $\eta > 0$.

Proof. Let $\alpha = \lambda_p + \lambda_d$. Since $[\text{cost}_t]_+ \leq 1$ and $(1 + \text{Tok}(a_t))^\gamma \geq 1$,

$$(\lambda_p + \lambda_d[\text{cost}_t]_+) \frac{[\delta_t]_+}{(1 + \text{Tok}(a_t))^\gamma} \leq \alpha[\delta_t]_+.$$

Therefore,

$$r_t^{\text{safe}} \leq \alpha[\delta_t]_+ - \alpha[-\delta_t]_+ = \alpha\delta_t.$$

Summing over t gives

$$\sum_t r_t^{\text{safe}} \leq \alpha \sum_t \delta_t = \alpha\big(\Phi_K(z_T) - \Phi_K(z_0)\big) \leq \alpha.$$

If $z_T = z_0$, the potential difference is zero, so the cumulative safe reward is at most zero. If $\eta > 0$, any nonempty cycle pays a strictly positive token cost, making the inequality strict. $\square$

This theorem should be read as a sufficient condition, not as a claim that the exact implemented reward has this strict guarantee. The implemented reward uses the simpler length-normalized progress term together with direction consistency and answer gating; the ablations empirically test these robustness mechanisms.

E.8 Connection to the Full MILE Reward

The complete trajectory reward in Section 2.3 is

$$R(\tau) = \lambda_o R_{\text{ver}}(\tau) + \big(\epsilon + (1-\epsilon)R_{\text{ver}}(\tau)\big) \sum_t \left(\lambda_p r_t^{\text{prog}} + \lambda_d r_t^{\text{dir}}\right).$$

The theory above analyzes the process part of this reward.

The verifier term R_{ver} is a trajectory-level correctness signal. The multiplier

$$g(\tau) = \epsilon + (1-\epsilon)R_{\text{ver}}(\tau)$$

satisfies $g(\tau) = 1$ for verified correct trajectories and $g(\tau) = \epsilon$ for incorrect trajectories. Thus answer gating preserves dense process rewards for globally correct solutions while strongly down-weighting dense rewards for trajectories that locally resemble expert progress but fail to solve the problem. This gating does not by itself prove optimality or rule out every possible reward-hacking behavior, but it limits the influence of local milestone similarity on incorrect rollouts.

The direction reward is also a practical robustness term. Theorems E.8 and E.13 show that it can be interpreted as an estimated expert progress direction, and that strict anti-cycle guarantees are obtained if direction reward is gated by positive potential progress. In the implemented reward, the additive direction term is used because it improves local alignment empirically; its effect is monitored through the direction alignment and false-progress diagnostics reported in the experiments.

Overall, the retrieval potential Φ_K is the mathematically controlled component of MILE. Under coverage and kernel-consistency assumptions, it estimates the future-solvability value used to annotate milestones. Multiple expert trajectories enlarge the support of this estimator across alternative solution paths, while answer gating and direction consistency serve as practical mechanisms that make the reward more robust during RL.