Internalizing Meta-Experience into Memory for Guided Reinforcement Learning in Large Language Models

Anonymous ACL submission

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as an effective approach for enhancing the reasoning capabilities of Large Language Models (LLMs). Despite its efficacy, RLVR faces a meta-learning bottleneck: it lacks mechanisms for error attribution and experience internalization intrinsic to the human learning cycle beyond practice and verification, thereby limiting fine-grained credit assignment and reusable knowledge formation. We term such reusable knowledge representations derived from past errors as meta-experience. Based on this insight, we propose **M**eta-**E**xperience **L**earning (**MEL**), a novel framework that incorporates self-distilled meta-experience into the model's parametric memory. Building upon standard RLVR, we introduce an additional design that leverages the LLM's self-verification capability to conduct contrastive analysis on paired correct and incorrect trajectories, identify the precise bifurcation points where reasoning errors arise, and summarize them into generalizable meta-experience. The meta-experience is further internalized into the LLM's parametric memory by minimizing the negative log-likelihood, which induces a language-modeled learning signal that bridges correct and incorrect reasoning trajectories and facilitates effective knowledge reuse. Experimental results demonstrate that MEL achieves consistent improvements on benchmarks, yielding 3.92%–4.73% Pass@1 gains across varying model sizes. Code will be available.

1 Introduction

Reinforcement Learning (RL) has emerged as a pivotal paradigm for enhancing the reasoning capabilities of Large Language Models (LLMs) on complex tasks, such as mathematics, programming, and logical reasoning (Shao et al., 2024; Wang et al., 2025b; Chen et al., 2025; Zeng et al., 2025; Wang et al., 2025a; Huang et al., 2026). By leveraging feedback signals from interaction with the task environment, RL enables LLMs to move beyond passive imitation learning toward goal-directed reasoning and action (Schulman et al., 2017; Ouyang et al., 2022; Wulfmeier et al., 2024). Furthermore, by replacing learned reward models with programmatically verifiable signals, Reinforcement Learning with Verifiable Rewards (RLVR) eliminates the need for expensive human annotations and mitigates reward hacking (Lambert et al., 2024), thereby enabling models to explore problem-solving strategies more effectively, contributing to its growing attention.

However, RLVR still faces a fundamental bottleneck regarding the granularity and utilization of learning signals. From a meta-learning perspective, the human learning cycle involves three critical components: practice and verification, error attribution, and experience internalization. While RLVR primarily drives policy updates through practice and verification, it overlooks the critical stages of error attribution and experience internalization, both of which are essential for fine-grained credit assignment and the formation of reusable knowledge (Wu et al., 2025; Zhang et al., 2025a). In other words, RLVR is largely limited to assessing the overall quality of entire trajectories, while struggling to reason about fine-grained knowledge at the level of intermediate steps (Xie et al., 2025). Although RL approaches (Lightman et al., 2023; Khalifa et al., 2025) employing Process Reward Models (PRMs) to provide dense learning signals attempt to mitigate this limitation, their reliance on trained proxies makes them inherently susceptible to reward hacking (Cheng et al., 2025; Guo et al., 2025), and poses a fundamental tension with the RLVR paradigm, which is centered on programmatically verifiable rewards.

Recently, a growing number of studies have explored integrating experience learning within the RLVR framework to address the above challenge (Zhang et al., 2025c,d). However, these approaches primarily utilize retrieved experience as

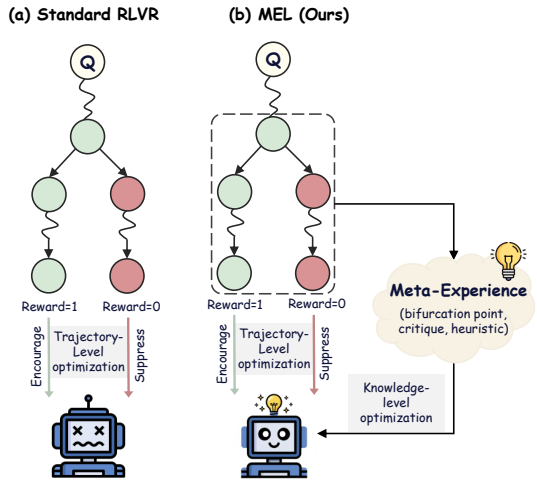

Figure 1: Paradigm comparison between standard RLVR and MEL, where MEL extends RLVR with an explicit knowledge-level learning loop.

external hints or training-time prefixes. While these strategies effectively elicit better reasoning paths during training, they introduce critical issues such as off-policy optimization instability (Zhang et al., 2025c) or severe distributional mismatch during inference when such prefixes are unavailable (Zhang et al., 2025d). Critically, despite their algorithmic differences, these methods share a fundamental limitation: they treat experience merely as external context rather than internalizing it. Consequently, the resulting learning signals remain predominantly at the *trajectory-level*, yielding superficial corrections rather than intrinsic cognitive enhancements.

Building on this insight, we introduce the concept of ***meta-experience***, elevating experience learning from *trajectory-level* instances to *knowledge-level* representations. Through an in-depth contrastive analysis of paired correct and incorrect trajectories, we pinpoint the bifurcation points underlying reasoning failures and abstract them into reusable meta-experiences. Accordingly, we propose **M**eta-**E**xperience **L**earning (**MEL**), a framework explicitly designed to enable knowledge-level internalization and reuse of meta-experiences. During training phase, MEL leverages meta-experiences to inject generalizable insights via a self-distillation mechanism, and internalizes them by minimizing the negative log-likelihood in the model's parametric memory. As shown in Figure 1, MEL differs from standard RLVR, which relies on coarse-grained outcome rewards and treats correct and incorrect trajectories independently, by explicitly connecting them via meta-experiences. Hence, this mechanism can be viewed as a language-modeled process-level reward signal, providing continuous and fine-grained guidance for improving reasoning capability. To further enhance stability and effectiveness during RLVR training, we propose empirical validation via replay, which uses meta-experiences as auxiliary in-context hints to assess their contribution to output accuracy. Meta-experiences that pass validation are integrated via negative log-likelihood minimization, while those that fail validation are excluded. In summary, our main contributions are as follows:

- We propose MEL, a novel framework that integrates self-distilled meta-experience with reinforcement learning, addressing the limitations of standard RLVR in error attribution and experience internalization by embedding these meta-experiences directly into the parametric memory of LLMs.
- Extensive experiments across three model scales (4B, 8B, 14B) show that MEL consistently improves Pass@1, Avg@8, and Pass@8. Training dynamics analysis reveals that MEL accelerates convergence and raises performance upper bounds, with gains that scale positively with model size. Ablation studies confirm the necessity of both the abstraction and empirical validation stages.
- We demonstrate the broad applicability of MEL across learning paradigms (e.g., RFT, GRPO, REINFORCE++), training datasets, model families, and task domains, establishing it as a versatile and generalizable framework for enhancing LLM reasoning.

2 Related Work

Reinforcement Learning with Verifiable Rewards. Reinforcement Learning with Verifiable Rewards (RLVR) leverages rule-based validators to provide deterministic feedback on models' self-generated solutions (Lambert et al., 2024). Extensive research has systematically investigated RLVR, exploring how this paradigm improves complex reasoning (Jaech et al., 2024; Guo et al., 2025; Liu et al., 2025; Zhang et al., 2025b). The pioneering framework Group Relative Policy Optimization (GRPO) (Shao et al., 2024) estimates advantages via group-wise relative comparisons, eliminating the need for a separate value model. Building on

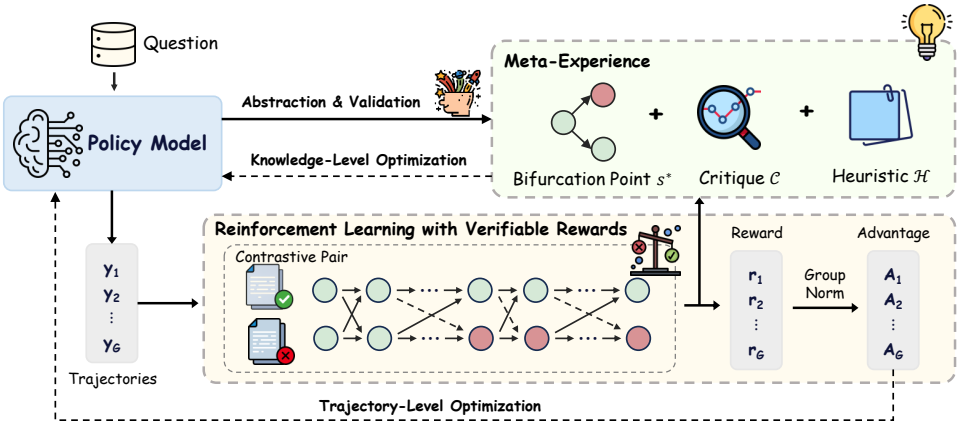

Figure 2: **M**eta-**E**xperience **L**earning (MEL) constructs meta-experiences from contrastive pairs via abstraction and validation, introducing an explicit knowledge-level learning loop over standard RLVR.

this base method, recent studies have introduced a range of algorithmic variants to improve training stability and efficiency. For instance, REINFORCE++ (Hu, 2025) enhances stability through global advantage normalization; DAPO (Yu et al., 2025) mitigates entropy collapse and improves reward utilization via relaxed clipping and dynamic sampling; and GSPO (Zheng et al., 2025) reduces gradient estimation variance with sequence-level clipping. Despite these algorithmic advancements, a fundamental limitation persists: current RLVR methods predominantly rely on outcome-level rewards. This failure to assign fine-grained credit to specific knowledge points prevents the construction of reusable knowledge formation, fundamentally constraining the development of systematic and generalizable reasoning capabilities.

Experience Learning. Recent studies have increasingly recognized that leveraging various forms of experience can substantially enhance LLM reasoning capabilities. One prominent line of research lies in test-time scaling methods, which store experience in external memory pools. For example, SpeedupLLM (Pan and Zhao, 2025) appends memories of previously reasoning traces as experience to accelerate inference, while Training Free GRPO (Cai et al., 2025) and ReasoningBank (Ouyang et al., 2025) distill accumulated experience into structured memory entries for retrieval-based augmentation. However, these approaches rely on ever-growing external memory, preventing the experience from being truly internalized and thus failing to substantively enhance intrinsic reasoning capabilities. Complementarily, another stream of research integrates experience directly into RL training as guiding signals. Methods such as Scaf-GRPO (Zhang et al., 2025d) and StepHint (Zhang et al., 2025c) employ external models to generate experiential hints, which are injected as prefixes or migration signals, to guide the policy toward higher-quality trajectories. Similarly, approaches like LUFFY (Yan et al., 2025) and SRFT (Fu et al., 2025) incorporate expert solution traces as additional experience. Despite improving exploration efficiency, these methods primarily induce trajectory-level imitation. Consequently, while models become proficient at mimicking specific solution patterns, they fail to internalize the underlying logic required for knowledge.

3 Meta-Experience Learning

Human learning follows a recurrent cognitive cycle consisting of practice and verification, error attribution, and experience internalization, which in turn informs subsequent practice. Motivated by this cognitive process, we define meta-experience for LLMs as generalizable and reusable knowledge derived from accumulated reasoning trials, capturing both underlying knowledge concepts and common failure modes. Building on this notion, we propose **M**eta-**E**xperience **L**earning **(MEL)**, a framework operating within the RLVR paradigm and designed to internalize such self-distilled, knowledge-level insights into the model's parametric memory. As illustrated in Figure 2, we first formalize the model exploration stage in RLVR (§3.1), then present the details of the Meta-Experience construction (§3.2). Finally, we describe the internalization mechanism (§3.3) for consolidating these insights into parametric memory, followed by the joint training objective for policy optimization (§3.4).

3.1 Explorative Rollout and Verifiable Feedback

Mirroring the "practice and check" phase in human learning, the RLVR framework engages the model in exploring potential solutions for reasoning tasks, while the environment serves as a deterministic verifier that provides verifiable feedback on the final answers. As mastering complex logic typically requires traversing the solution space through multiple attempts, we simulate this stochastic exploration by adopting the group rollout formulation from Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

Formally, given a query x sampled from the dataset $\mathcal{D}$, the policy model π_θ performs stochastic exploration over the solution space and generates a group of G independent reasoning trajectories $\mathcal{Y} = \{y_1, y_2, \ldots, y_G\}$. A rule-based verifier then evaluates each trajectory using a verification function $V(\cdot)$, which compares the extracted final answer from y_i against the ground-truth answer y^* and assigns a binary outcome reward:

$$r_i = \mathbb{I}\big[V(y_i, y^*)\big] \in \{0, 1\}. \tag{1}$$

This process partitions the generated group $\mathcal{Y}$ into two distinct subsets: the set of correct trajectories $\mathcal{Y}^+ = \{y_i \mid r_i = 1\}$ and the set of incorrect trajectories $\mathcal{Y}^- = \{y_i \mid r_i = 0\}$.

Beyond providing the necessary variance for standard RLVR policy updates, the coexistence of $\mathcal{Y}^+$ and $\mathcal{Y}^-$ for the same query serves as the critical foundation for our framework. Specifically, this natural pairing of successes and failures enables the extraction of knowledge-level insights through a systematic contrastive analysis.

3.2 Meta-Experience Construction

Prior studies (Lightman et al., 2023; Xie et al., 2025; Khalifa et al., 2025) have shown that effective learning does not arise from merely knowing that a final answer is incorrect, but rather from identifying the specific bifurcation point where the reasoning deviates and understanding the root causes of errors, a critical process known as error attribution. Building on this insight, we leverage pairs of correct and incorrect trajectories to localize reasoning errors and distill such bifurcation points into explicit meta-experiences.

Locating the Bifurcation Point. To extract knowledge-level learning signals from the exploration results, we focus on identifying the bifurcation points where the reasoning logic diverges into an erroneous path, which serves as the fundamental source of cognitive insights. With the exploration results partitioned into $\mathcal{Y}^+$ and $\mathcal{Y}^-$ by the verifier, we construct multiple contrastive pairs $\mathcal{P}_x = \{(y^+, y^-) \mid y^+ \in \mathcal{Y}^+, y^- \in \mathcal{Y}^-\}$ for each query x, whose contrast naturally exposes the specific errors in the reasoning process. Such contrastive analysis requires the presence of both positive and negative trajectories; accordingly, we only consider gradient-informative samples with non-empty $\mathcal{Y}^+$ and $\mathcal{Y}^-$.

For fine-grained comparison within each pair, each trajectory y can be formatted as a reasoning chain $y = (s_1, \ldots, s_L, a)$, where each s_t denotes an atomic reasoning step and a indicates the final answer. Since both trajectories originate from the same context, they typically share a correct reasoning path until a critical divergence step s^* occurs.

Given deterministic verification signals and full access to the reasoning chains, identifying the bifurcation point can be viewed as a discriminative task that is easier than reasoning from scratch (Saunders et al., 2022; Swamy et al., 2025). Motivated by this observation, we task the policy model with analyzing each contrastive pair to identify the reasoning bifurcation point s^*:

$$s^* \sim \pi_\theta\big(\cdot \mid I, x, y^+, y^-\big). \tag{2}$$

where I denotes a structured instruction guiding introspective analysis.

Deep Diagnosis and Abstraction. Identifying the bifurcation point s^* localizes exactly where the reasoning fails, serving as the anchor for subsequent learning. Grounded at this point, the policy model conducts a deep diagnostic analysis to contrast the strategic choices underlying both the successful and failed trajectories. Specifically, the model examines the local reasoning context around s^* to pinpoint the root cause of failure, such as violated assumptions, erroneous sub-goals, overlooked constraints, or the misuse of specific principles. In parallel, it inspects the successful trajectory to uncover the mechanisms that prevented these pitfalls, including precise knowledge application, explicit constraint verification, coherent knowledge representations, or emergent self-correction behaviors. By synthesizing these perspectives, the model distills the structural divergence between the correct and incorrect logic, crystallizing it into explicit knowledge. Formally, we model this diagnostic

process as generating a critique $\mathcal{C}$ that encapsulates the error attribution, the comparative strategic gap, and the corresponding corrective principle:

$$\mathcal{C} \sim \pi_\theta\big(\cdot \mid I, x, y^+, y^-, s^*\big). \tag{3}$$

To ensure generalization, it is imperative for the model to distill instance-specific critiques into abstract heuristics capable of guiding future reasoning. This abstraction step systematically strips away context-dependent variables, extracting the core logic of successes and failures into general rules. Structurally, such heuristics synthesize abstract problem categorization with the corresponding reasoning principles, encompassing the essential knowledge points, theoretical theorems, and decision criteria. Crucially, they also highlight common pitfalls and latent constraints to prevent recurrent errors in similar tasks. We define the extraction of this heuristic knowledge $\mathcal{H}$ as a generation process conditioned on the full critique:

$$\mathcal{H} \sim \pi_\theta\big(\cdot \mid I, x, y^+, y^-, s^*, \mathcal{C}\big). \tag{4}$$

Finally, we consolidate these components into a raw *Meta-Experience* tuple $\mathcal{M}$, which elevates experience learning from trajectory-level instances to unified knowledge-level representations.

$$\mathcal{M} = \big(s^*, \mathcal{C}, \mathcal{H}\big). \tag{5}$$

This formulation enables meta-experiences to be reused across tasks that share analogous reasoning structures, serving as a fine-grained learning signal. By applying the extraction process across distinct contrastive pairs for a query x, we construct a candidate pool of meta-experiences $\mathcal{D}_{\mathcal{M}} = \{(x, y_i^+, y_i^-, \mathcal{M}_i)\}_{i=1}^N$, where N denotes the total number of meta-experiences derived from x, and (y_i^+, y_i^-) represents the specific contrastive pair used to derive $\mathcal{M}_i$.

Empirical Validation via Replay. Closing the cognitive loop requires re-instantiating theoretical insights derived from past failures in future problem-solving to assess their validity. We recognize that the raw meta-experience $\mathcal{M}$ may still suffer from intrinsic hallucinations or causal misalignment. To mitigate this, we conduct empirical verification by incorporating $\mathcal{M}$ as short-term working memory into the prompt, thereby guiding the model to re-attempt the original query x. This procedure tests whether the injected meta-experience can effectively steer the model away from the previously identified bifurcation point s^* and toward a correct reasoning trajectory.

We retain a meta-experience only if the associated replay trajectory $y_{\text{val}} \sim \pi_\theta(\cdot \mid x, \mathcal{M})$ satisfies the verifier by producing the correct answer:

$$\mathcal{D}_{\mathcal{M}^*} = \big\{(x, y^+, y^-, \mathcal{M}) \in \mathcal{D}_{\mathcal{M}} \mid V(y_{\text{val}}, y^*) = 1\big\}. \tag{6}$$

Consequently, this empirical validation preserves only high-quality meta-experiences for integration into parametric long-term memory, guaranteeing the reliability of the supervision signals used in the subsequent optimization phase.

3.3 Internalization Mechanism

The verified meta-experiences $\mathcal{D}_{\mathcal{M}^*}$ constitute a high-quality reservoir of reasoning guidance. However, treating these insights as retrieval-augmented memory imposes a substantial inference burden by requiring elongated contexts for every query. To overcome this, we transfer these insights into the model's parametric memory. Unlike the finite context buffer, the model parameters offer a virtually unlimited capacity for accumulating diverse meta-experiences, allowing the policy to internalize vast amounts of reasoning patterns without incurring inference-time latency.

We establish this internalization process as a self-distilled paradigm, where the model learns from its own verified experiences. Specifically, we employ fine-tuning based on the token-averaged negative log-likelihood (NLL) objective to compile the meta-experiences into the policy's weights. Formally, given the retrospective context $C_{\text{retro}} = [I, x, y^+, y^-]$, we first define the log-likelihood of generating the meta-experience as:

$$L(\mathcal{M}^*) = \frac{1}{|\mathcal{M}^*|} \sum_{t=1}^{|\mathcal{M}^*|} \log \pi_\theta(\mathcal{M}_t^* \mid C_{\text{retro}}, \mathcal{M}_{<t}^*). \tag{7}$$

Accordingly, the overall internalization loss is defined as:

$$\begin{aligned} \mathcal{L}_{\text{NLL}}(\theta) &= -\mathbb{E}_{(x,y^+,y^-,\mathcal{M}^*)\sim\mathcal{D}_{\mathcal{M}^*}}\big[L(\mathcal{M}^*)\big] \\ &= -\mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}\sim\pi_{\theta_{\text{old}}}} \\ &\qquad \Big[\mathbb{E}_{(y^+,y^-,\mathcal{M}^*)\sim\mathcal{T}(x,\{y_i\})}\big[L(\mathcal{M}^*)\big]\Big]. \end{aligned} \tag{8}$$

where $\mathcal{T}(\cdot)$ represents the stochastic construction process detailed in §3.2.

Based on this formulation, the internalization process can be viewed as a specialized sampling form within the RLVR framework. By inverting the loss, we define the Meta-Experience Return $\mathcal{R}_{\text{MEL}}$ as the expected log-likelihood over the stochastically constructed verification set:

$$\mathcal{R}_{\text{MEL}} = \mathbb{E}_{(y^+,y^-,\mathcal{M}^*)\sim\mathcal{T}(x,\{y_i\}_{i=1}^G)}\big[L(\mathcal{M}^*)\big]. \tag{9}$$

Table 1: **Main Results Comparison.** Comparison of Pass@1, Avg@8, and Pass@8 accuracy (%) across different model scales. The best results within each model scale are marked in **bold**.

	AIME 2024			AIME 2025			AMC 2023		
Method	**Pass@1**	**Avg@8**	**Pass@8**	**Pass@1**	**Avg@8**	**Pass@8**	**Pass@1**	**Avg@8**	**Pass@8**
Qwen3-4B-Base									
Baseline	13.33	9.90	30.00	10.00	6.56	23.33	45.00	42.73	72.50
GRPO	13.33	18.33	30.00	6.67	17.50	30.00	57.50	58.13	85.00
MEL	**20.00**	**20.83**	**33.00**	**16.67**	**18.33**	**33.00**	**60.00**	**60.31**	**87.50**
Qwen3-8B-Base									
Baseline	6.67	10.00	26.67	13.33	15.00	33.33	65.00	52.50	87.50
GRPO	16.67	24.58	43.33	20.00	20.83	**36.67**	67.50	69.06	87.50
MEL	**30.00**	**25.42**	**60.00**	**23.33**	**23.33**	**36.67**	**70.00**	**70.31**	**90.00**
Qwen3-14B-Base									
Baseline	13.33	10.83	36.67	6.66	9.58	33.33	60.00	51.25	82.50
GRPO	30.00	35.41	56.67	33.33	24.17	43.33	75.00	75.94	**95.00**
MEL	**33.33**	**35.83**	**60.00**	**36.67**	**30.00**	**46.67**	**82.50**	**82.81**	**95.00**

	MATH 500			OlympiadBench			Average		
Method	**Pass@1**	**Avg@8**	**Pass@8**	**Pass@1**	**Avg@8**	**Pass@8**	**Pass@1**	**Avg@8**	**Pass@8**
Qwen3-4B-Base									
Baseline	74.20	65.74	89.60	39.17	35.37	60.38	36.34	32.06	55.16
GRPO	81.80	82.20	93.00	**48.51**	48.46	67.21	41.56	44.92	61.04
MEL	**82.20**	**82.30**	**93.80**	**48.51**	**49.48**	**69.73**	**45.48**	**46.25**	**63.41**
Qwen3-8B-Base									
Baseline	77.00	73.40	91.40	44.51	39.41	64.09	41.30	38.06	60.60
GRPO	84.40	86.28	95.40	53.56	54.60	**73.74**	48.43	51.07	67.33
MEL	**86.60**	**86.70**	**96.20**	**54.01**	**55.60**	73.00	**52.79**	**52.27**	**71.17**
Qwen3-14B-Base									
Baseline	80.80	74.15	93.60	45.25	40.50	65.58	41.21	37.26	62.34
GRPO	85.00	88.35	96.40	58.16	58.46	74.78	56.30	56.47	73.24
MEL	**90.80**	**90.80**	**97.20**	**61.87**	**60.90**	**75.82**	**61.03**	**60.07**	**74.94**

3.4 Joint Training Objective

To simultaneously encourage exploration and consolidate the internalized meta-experiences, achieving dual optimization across trajectory-level behaviors and knowledge-level representations, we train the policy π_θ using a joint optimization objective. This objective synergizes the RLVR signal from diverse explorative rollouts with the supervised signal high-quality meta-experiences:

$$\mathcal{J}(\theta) = \mathcal{J}_{\text{RLVR}}(\theta) + \mathcal{J}_{\text{MEL}}(\theta). \tag{10}$$

We adopt GRPO (Shao et al., 2024) as the RLVR component and compute group-normalized advantages by standardizing rewards within the group and broadcasting to each token. Let $y_{i,t}$ denote the t-th token in trajectory y_i and $y_{i,<t}$, the corresponding prefix. The objective $\mathcal{J}_{\text{RLVR}}$ is formulated as:

$$\begin{aligned}\mathcal{J}_{\text{RLVR}}(\theta) =& \mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \\ & \Big[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(\rho_{i,t}(\theta)\hat{A}_{i,t}, \\ & \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_{i,t}\Big)\Big] \\ =& \mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\big[\mathcal{R}_{\text{RLVR}}\big]\end{aligned} \tag{11}$$

Crucially, as formulated in Eq. 8, the objective $\mathcal{J}_{\text{MEL}}$ operates over the exact same outer expectation. By sharing this unified sampling space, the joint optimization inherently aggregates trajectory-level and knowledge-level learning signals, effectively optimizing a unified hybrid return. Substituting these definitions, the joint objective in Eq. 10 can be elegantly expanded and merged as:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\{y_i\}\sim\pi_{\theta_{\text{old}}}}\Big[\mathcal{R}_{\text{RLVR}} + \mathcal{R}_{\text{MEL}}\Big] \tag{12}$$

This unification reveals a profound reinforcement learning perspective. Although $\mathcal{J}_{\text{MEL}}$ is derived from a log-likelihood objective, its optimization gradient is mathematically equivalent to a policy gradient update where the reward signal is a constant positive scalar. In this unified view, the meta-experiences function as dense supervision over the reasoning process. Unlike the sparse outcome rewards in standard RLVR that only evaluate final correctness, MEL ensures that the model not only pursues correct outcomes via broad exploration but is also continuously shaped by its own successful cognitive patterns, effectively bridging the gap between trajectory-level search and token-level knowledge encoding.

4 Experiments

Datasets. We train our model on the DAPO-Math-17k dataset (Yu et al., 2025) and evaluate it

on five challenging mathematical reasoning benchmarks: AIME24, AIME25, AMC23 (Li et al., 2024), MATH500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024).

Experimental Setups. All reinforcement learning training is conducted using the VERL framework (Sheng et al., 2024) on 8×H20 GPUs, with Math-Verify providing rule-based outcome verification. During training, we sample 8 responses per prompt at a temperature of 1.0 with a batch size of 128. Optimization uses a learning rate of 1×10^{-6} and a mini-batch size of 128. For evaluation, we report Pass@1 at temperature 0, and Avg@8 and Pass@8 at temperature 0.6.

Models and Baselines. To demonstrate the general applicability of MEL, we conduct experiments across a diverse range of model scales, including Qwen3-4B-Base, Qwen3-8B-Base, and Qwen3-14B-Base (Yang et al., 2025). We adopt GRPO (Shao et al., 2024) as the base reinforcement learning algorithm for MEL, and thus perform a direct and controlled comparison between the vanilla GRPO and our meta-experience learning approach.

4.1 Experimental Results

As shown in Table 1, MEL achieves consistent and significant improvements over vanilla GRPO and the basemodel across multiple benchmarks and model scales. We report three complementary metrics: Pass@1 reflects one-shot reliability, Avg@8 measures the average performance over 8 samples, and Pass@8 reports the best-of-8 success rate.

First, the gains in Pass@1 demonstrate that MEL substantially enhances the model's confidence in following correct reasoning paths, achieving a consistent improvement of 3.92–4.73% over the strong GRPO baseline across all model scales. This indicates that MEL effectively internalizes the explored insights into parametric memory, generating high-confidence solutions and markedly reducing the need for extensive test-time sampling. This reliability is further corroborated by the gains in Avg@8, which confirm that internalized meta-experiences function as intrinsic process-level guidance, continuously steering the generation toward valid logic and enhancing reasoning consistency. Finally, the sustained gains in Pass@8 suggest that learning from meta-experience does not harm exploration; instead, it expands the reachable solution space and raises the upper bound of best-of-k performance.

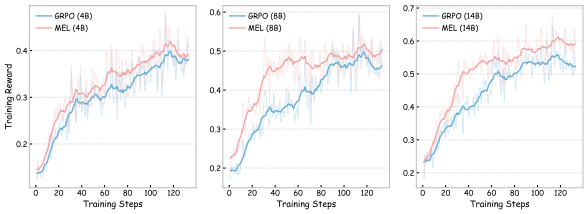

Figure 3: Training curves comparing GRPO and MEL across three model scales (4B, 8B, 14B).

Beyond mathematical reasoning, models trained with MEL also exhibit strong out-of-domain (OOD) generalization, improving on knowledge-intensive and coding benchmarks (Appendix A.1).

4.2 Training Dynamics and Scalability

To understand the performance gains under MEL, we monitored the training dynamics and validation performance in Figures 3 and 8–10.

Accelerated Convergence. Vanilla GRPO often struggles to obtain positive reinforcement in the early stages, particularly when initial performance is low, due to the sparsity of outcome-based rewards. As illustrated in Figure 3, vanilla GRPO exhibits a relatively slow ascent initially. In contrast, MEL demonstrates a sharp, rapid reward accumulation immediately from the onset of training. This acceleration is attributed to the internalized meta-experience return, $\mathcal{R}_{\text{MEL}}$. By functioning as a dense process supervision signal, $\mathcal{R}_{\text{MEL}}$ continuously provides informative gradient signals for every reasoning step, even when successful trajectories yielding positive reinforcement are scarce.

Beyond sample efficiency, the average reward of MEL maintains a persistent lead over vanilla GRPO throughout training. Crucially, the downstream validation trajectories reveal that even as performance growth begins to plateau in the later stages, MEL maintains a distinct and sustained advantage over the baseline, suggesting that internalized meta-experiences empower the model to navigate more complex, long-horizon solutions that remain inaccessible to the baseline. A qualitative case study in Appendix C further illustrates how internalized meta-experiences concretely reshape the model's reasoning behavior.

Positive Scaling with Model Size. A striking pattern emerges from the training curves in Figure 3: the performance margin between MEL and the baseline widens significantly as the model size increases from 4B to 14B, exhibiting a distinct positive scaling law that consistently extends to

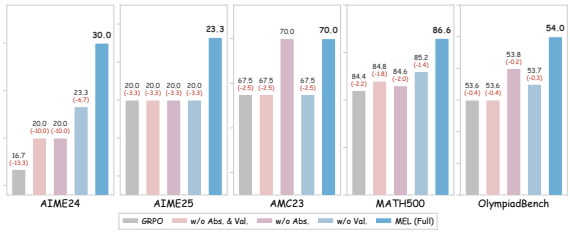

Figure 4: Ablation of MEL's core stages (Abstraction and Validation) on Qwen3-8B-Base.

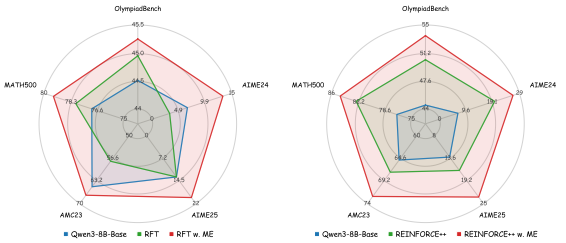

Figure 5: Performance of **Meta-Experience** (ME) integrated into RFT and REINFORCE++.

downstream validation benchmarks. We attribute this to the quality of self-generated supervision, which is inherently bounded by the model's intrinsic capability. As shown in Figure 6, the 14B model achieves a significantly higher yield rate of valid meta-experiences than its smaller counterparts. While limited-capacity models introduce noise due to imprecise error attribution, larger models benefit from stronger self-verification, enabling the distillation of higher-quality heuristics that provide more accurate gradient signals and fully realize the potential of our framework.

This observation raises the question of whether using a stronger model to generate meta-experiences for a weaker one further boosts performance. We explore this cross-model transfer in Appendix B.2, where supplying 14B-generated meta-experiences to the 8B model yields further gains, confirming that the quality of meta-experiences is the primary driver of this scaling behavior.

4.3 Ablation Study

To systematically evaluate the individual contributions of our framework's core components, we conduct an ablation study on Qwen3-8B-Base, comparing the full MEL against variants with the abstraction stage removed (w/o Abs.), validation removed (w/o Val.), and both removed (w/o Abs. & Val.). As shown in Figure 4, the full framework achieves the best performance across all benchmarks.

Specifically, removing abstraction (w/o Abs.) preserves training reward but clearly degrades downstream performance, indicating that the model overfits to instance-specific reward gains rather than acquiring transferable reasoning capabilities. Distilling concrete error patterns into higher-level heuristic knowledge $\mathcal{H}$ is therefore crucial for translating training signals into generalizable reasoning strategies. Omitting validation (w/o Val.) impairs performance even more severely, as unverified meta-experiences introduce hallucinated outputs. This underscores the indispensable role of empirical validation: it ensures the causal validity of the internalized knowledge, which is essential for enabling the model to sustain deep and reliable reasoning trajectories. When both stages are removed, the degradation compounds further, confirming the joint necessity of both core components.

4.4 Generality Across Learning Paradigms

To demonstrate the versatility of meta-experience, we integrated it into RFT and REINFORCE++ using the Qwen-8B-Base model as the backbone and the same training set in our experiments. As shown in Figure 5, while vanilla RFT often suffers from rote memorization and tends to overfit to specific samples in this training set, the incorporation of meta-experiences introduces robust reasoning heuristics. This allows the model to internalize the underlying logic rather than merely imitating specific answers, thereby effectively mitigating overfitting and enhancing generalization to unseen test sets. Similarly, applying meta-experiences to REINFORCE++ significantly raises the performance ceiling on benchmarks. This confirms that the benefit of internalized meta-experiences is a universal enhancement, not limited to the GRPO framework.

Beyond algorithmic generality, MEL consistently outperforms vanilla RLVR across diverse datasets, models, and domains (Appendix A).

5 Conclusion

In this paper, we introduce **MEL** to overcome the meta-learning bottleneck in standard RLVR by transforming instance-level failures into reusable cognitive assets. Unlike traditional outcome-reward methods, it performs granular error attribution, distilling failure modes into natural language heuristics formalized as **Meta-Experiences**. By internalizing these experiences into parametric memory, our approach bridges the gap between verifying a solution and understanding the underlying reasoning logic, consistently boosting mathematical reasoning across diverse model scales.

Limitations

While MEL substantially enhances reasoning capabilities across diverse model scales, two practical limitations remain. (1) The meta-experience construction pipeline introduces additional computational overhead during training compared to standard RLVR. However, this cost is strictly confined to the training phase and imposes no burden on inference efficiency. Furthermore, the accelerated convergence exhibited by MEL partially offsets this overhead by requiring fewer overall training steps to reach comparable performance. (2) The quality of generated meta-experiences is inherently bounded by the base model's intrinsic capability. Weaker models may produce noisier heuristic signals, a phenomenon consistent with our observed scaling trends. Nevertheless, as the community continues to advance foundation model capabilities, this concern is expected to diminish naturally.

References

Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, and 1 others. 2025. Training-free group relative policy optimization. *arXiv preprint arXiv:2510.08191*.

Jiangjie Chen, Qianyu He, Siyu Yuan, Aili Chen, Zhicheng Cai, Weinan Dai, Hongli Yu, Qiying Yu, Xuefeng Li, Jiaze Chen, and 1 others. 2025. Enigmata: Scaling logical reasoning in large language models with synthetic verifiable puzzles. *arXiv preprint arXiv:2505.19914*.

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

Yuqian Fu, Tinghong Chen, Jiajun Chai, Xihuai Wang, Songjun Tu, Guojun Yin, Wei Lin, Qichao Zhang, Yuanheng Zhu, and Dongbin Zhao. 2025. Srft: A single-stage method with supervised and reinforcement fine-tuning for reasoning. *arXiv preprint arXiv:2506.19767*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the Association for Computational Linguistics*, pages 3828–3850.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv: 2103.03874*.

Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*.

Wenxuan Huang, Yu Zeng, Qiuchen Wang, Zhen Fang, Shaosheng Cao, Zheng Chu, Qingyu Yin, Shuang Chen, Zhenfei Yin, Lin Chen, and 1 others. 2026. Vision-deepresearch: Incentivizing deepresearch capability in multimodal large language models. *arXiv preprint arXiv:2601.22060*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

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

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *Proceedings of the International Conference on Learning Representations*.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. 2025. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle. *arXiv preprint arXiv:2509.16679*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1

others. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, and 1 others. 2025. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*.

Bo Pan and Liang Zhao. 2025. Can past experience accelerate llm reasoning? *arXiv preprint arXiv:2505.20643*.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. 2022. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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

Gokul Swamy, Sanjiban Choudhury, Wen Sun, Zhiwei Steven Wu, and J Andrew Bagnell. 2025. All roads lead to likelihood: The value of reinforcement learning in fine-tuning. *arXiv preprint arXiv:2503.01067*.

Qiuchen Wang, Ruixue Ding, Yu Zeng, Zehui Chen, Lin Chen, Shihang Wang, Pengjun Xie, Fei Huang, and Feng Zhao. 2025a. Vrag-rl: Empower vision-perception-based rag for visually rich information understanding via iterative reasoning with reinforcement learning. *arXiv preprint arXiv:2505.22019*.

Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. 2025b. Co-evolving llm coder and unit tester via reinforcement learning. *arXiv preprint arXiv:2506.03136*.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, and 1 others. 2025. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*.

Markus Wulfmeier, Michael Bloesch, Nino Vieillard, Arun Ahuja, Jorg Bornschein, Sandy Huang, Artem Sokolov, Matt Barnes, Guillaume Desjardins, Alex Bewley, and 1 others. 2024. Imitating language via scalable inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 37:90714–90735.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. 2025. Capo: Towards enhancing llm reasoning through verifiable generative credit assignment. *arXiv e-prints*, pages arXiv–2508.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu Zeng, Wenxuan Huang, Shiting Huang, Xikun Bao, Yukun Qi, Yiming Zhao, Qiuchen Wang, Lin Chen, Zehui Chen, Huaian Chen, and 1 others. 2025. Agentic jigsaw interaction learning for enhancing visual perception and reasoning in vision-language models. *arXiv preprint arXiv:2510.01304*.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, and 1 others. 2025a. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025b. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. 2025c. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. 2025d. Scaf-grpo: Scaffolded group relative policy optimization for enhancing llm reasoning. *arXiv preprint arXiv:2510.19807*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Broader Applicability

A critical question for any learning framework is whether its benefits are confined to a specific experimental configuration or generalize across diverse settings. To comprehensively evaluate the applicability of MEL, we conduct experiments along four orthogonal dimensions: cross-domain evaluation of math-trained models, alternative training datasets, different model families, and cross-domain training on non-mathematical tasks.

A.1 Cross-Domain Generalization

A common concern with intensive specialization on a single domain is catastrophic forgetting, where targeted training degrades the model's broader capabilities. To investigate whether this issue arises under MEL, we evaluate the math-trained Qwen3-8B-Base model (trained on DAPO-Math-17k) on knowledge-intensive benchmarks (MMLU, GPQA-main) and coding benchmarks (MBPP, HumanEval).

As shown in Table 2, MEL not only preserves but also *improves* performance on these out-of-distribution tasks compared to both the base model and vanilla GRPO. Notably, while vanilla GRPO causes a slight decline in MMLU, MEL achieves a 2.6-point improvement, suggesting that the meta-experience internalization process does not merely memorize domain-specific patterns but cultivates transferable reasoning heuristics. The consistent gains on GPQA-main further corroborate this finding, as improvements on both knowledge-intensive and reasoning-heavy benchmarks suggest that the heuristics internalized by MEL capture transferable reasoning patterns rather than domain-specific shortcuts. Meanwhile, coding capabilities (MBPP, HumanEval) remain fully intact, confirming that MEL does not sacrifice breadth for depth.

Table 2: Cross-domain evaluation of math-trained models on out-of-distribution benchmarks. All models are trained on DAPO-Math-17k.

Method	MMLU	GPQA	MBPP	HumanEval
Qwen3-8B-Base	76.7	37.1	70.0	80.0
GRPO	76.2	38.0	70.4	86.7
MEL	**78.8**	**39.8**	**71.6**	**86.7**

A.2 Alternative Training Dataset

To verify that the effectiveness of MEL is not contingent on the specific characteristics of the DAPO-Math-17k corpus, we train Qwen3-8B-Base on a 25k subset of the DeepScalaR dataset, which differs substantially in problem distribution and difficulty composition.

As shown in Table 3, MEL consistently outperforms vanilla GRPO across all benchmarks, with particularly pronounced gains on competition-level tasks. The fact that these improvements persist under a completely different data distribution indicates that the meta-experience construction and internalization pipeline is robust to the source of training problems.

Table 3: Results when training on a 25k subset of the DeepScalaR dataset. OlyBench is short for OlympiadBench.

Method	AIME24	AIME25	AMC23	MATH500	OlyBench
GRPO	23.3	16.7	62.5	85.2	51.2
MEL	**30.0**	**20.0**	62.5	**86.0**	**54.9**

A.3 Cross-Family Generalization

To demonstrate that MEL is not restricted to the Qwen model family, we apply it to LLaMA-3.1-8B-Instruct, which differs in tokenizer and pre-training corpus, and LLaMA-3.1-8B-Instruct has relatively weak mathematical reasoning capabilities as a starting point.

As shown in Table 4, MEL consistently outperforms vanilla GRPO, doubling the AIME24 score and largely improving OlympiadBench. Interestingly, vanilla GRPO actually degrades OlympiadBench performance relative to the baseline, while MEL not only recovers but surpasses it. This suggests that when model capability is limited, standard RLVR may lead to reward hacking or degenerate behaviors, whereas the meta-experience internalization provides a more structured learning signal that prevents such degradation.

Table 4: Results on LLaMA-3.1-8B-Instruct. OlyBench is short for OlympiadBench.

Method	AIME24	AIME25	AMC23	MATH500	OlyBench
Baseline	6.7	0.0	22.5	51.4	19.7
GRPO	6.7	0.0	35.0	52.6	17.5
MEL	**13.3**	0.0	**37.5**	**54.8**	**21.4**

A.4 Cross-Domain Training

All preceding experiments train on mathematical reasoning tasks. To prove that the meta-experience learning paradigm itself is domain-agnostic, we train Qwen3-8B-Base on a 20k subset of the MMLU dataset, which spans 57 diverse subjects

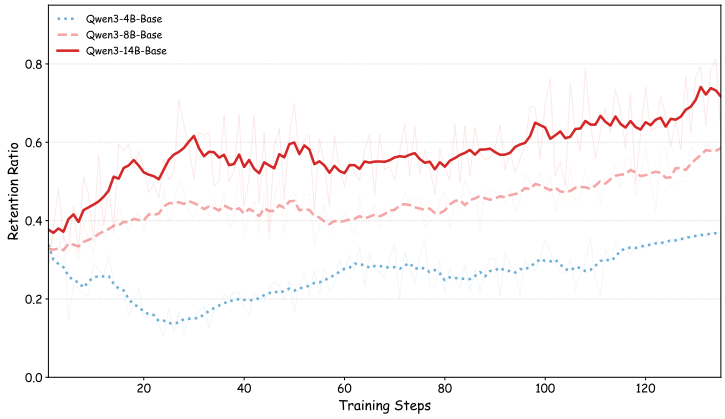

Figure 6: Dynamics of the retention ratio of MEL across different model scales.

including humanities, social sciences, and natural sciences.

As shown in Table 5, MEL significantly surpasses the GRPO baseline by 5.1 points. The magnitude of this improvement is notably larger than the gains observed in mathematical benchmarks, which we attribute to the nature of MMLU tasks: many questions require recalling and applying domain-specific knowledge with precise reasoning, which aligns well with the meta-experience framework's strength in distilling and internalizing knowledge-level heuristics. This result demonstrates that the MEL framework generalizes beyond the mathematical domain and can be readily applied to a wide range of reasoning tasks to enhance model capabilities.

Table 5: Results when training on a 20k subset of the MMLU dataset.

Method	MMLU Score
Qwen3-8B-Base	76.7
GRPO	81.9
MEL	**87.0**

B Meta-Experience Quality and Scalability

B.1 Retention Ratio across Model Scales

Through empirical validation via replay, MEL is able to collect high-quality meta-experiences. To examine the utilization of collected meta-experiences, Figure 6 reports the retention ratio of meta-experiences after empirical validation throughout training. We observe that the retention ratio consistently increases with model scale, indicating that larger models are more effective at abstracting high-quality knowledge into meta-experiences, thereby achieving higher retention.

B.2 Cross-Model Meta-Experience Transfer

As discussed in §4.2, the quality of self-generated meta-experiences is inherently limited by the model's own capability, creating a natural bottleneck for self-distillation. A key question is whether this limitation can be alleviated by using a stronger model to generate meta-experiences for a weaker one, thereby separating the quality of reasoning supervision from the student model's intrinsic capacity.

To investigate this, we use the MEL-trained Qwen3-14B model as a meta-experience generator to produce heuristic knowledge for training the Qwen3-8B model, while keeping all other training configurations identical. The 14B generator is expected to produce more precise error attribution and higher-quality heuristics due to its stronger intrinsic reasoning ability.

As shown in Table 6, cross-model transfer yields consistent improvements across all benchmarks. The most pronounced gains appear on the challenging competition-level tasks: AIME25 improves from 23.3 to 26.7 (+3.4), and AMC23 from 70.0 to 75.0 (+5.0). These are precisely the benchmarks where the 8B model's self-generated meta-experiences are most likely to suffer from imprecise error attribution, confirming our hypothesis that meta-experience quality is the key scaling bottleneck.

Importantly, this result demonstrates that the MEL framework is not restricted to self-distillation. When a stronger teacher is available, it can serve as a higher-quality meta-experience generator, further amplifying the benefits of the framework. This

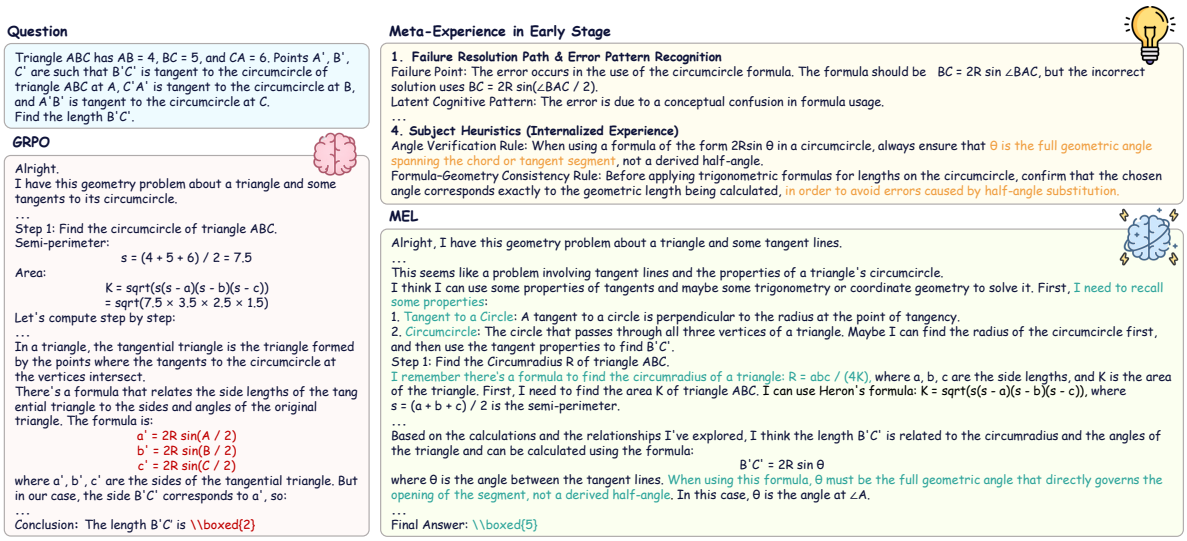

Figure 7: Case study comparing GRPO and MEL, with visualization of meta-experience in early stage.

positions MEL as a flexible paradigm that scales naturally with improvements in meta-experience quality, whether through larger self-models or external teachers.

Table 6: Cross-model meta-experience transfer. Using MEL (14B) to generate meta-experiences for Qwen3-8B consistently outperforms self-generated meta-experiences. OlyBench is short for OlympiadBench.

ME Generator	AIME24	AIME25	AMC23	MATH500	OlyBench
Self-generated	30.0	23.3	70.0	86.6	54.0
MEL (14B)	**30.0**	**26.7**	**75.0**	**87.4**	**54.7**

C How Meta-Experience Shapes Reasoning Patterns

To investigate how MEL shapes the model's cognitive processes beyond numerical metrics, we conduct a qualitative analysis comparing the reasoning trajectories of MEL and the baseline GRPO model, as visualized in Figure 7.

A distinct behavioral divergence is observed from the onset of the solution. While the GRPO baseline tends to prioritize immediate execution through direct numerical operations, MEL adopts a structured preparatory strategy by explicitly outlining relevant theorems and formulas. Although the direct approach may appear efficient for simple queries, it increases the susceptibility to errors in complex tasks due to the lack of a holistic view of problem constraints.

Notably, MEL exhibits an emergent cognitive behavior. When applying specific theorems, it spontaneously activates internalized "bitter lessons" as endogenous safeguards to regulate its actions. These active signals effectively reduce reasoning drift by encouraging earlier constraint checking and consistent self-correction when the model enters uncertain regions.

D Result of Performance Evolution

As illustrated in Figures 8, 9, and 10, we visualize the performance evolution of models with different scales (Qwen3-4B-Base, Qwen3-8B-Base, and Qwen3-14B-Base) across multiple benchmarks throughout training. It can be observed that MEL consistently outperforms standard GRPO in terms of average performance on all benchmarks.

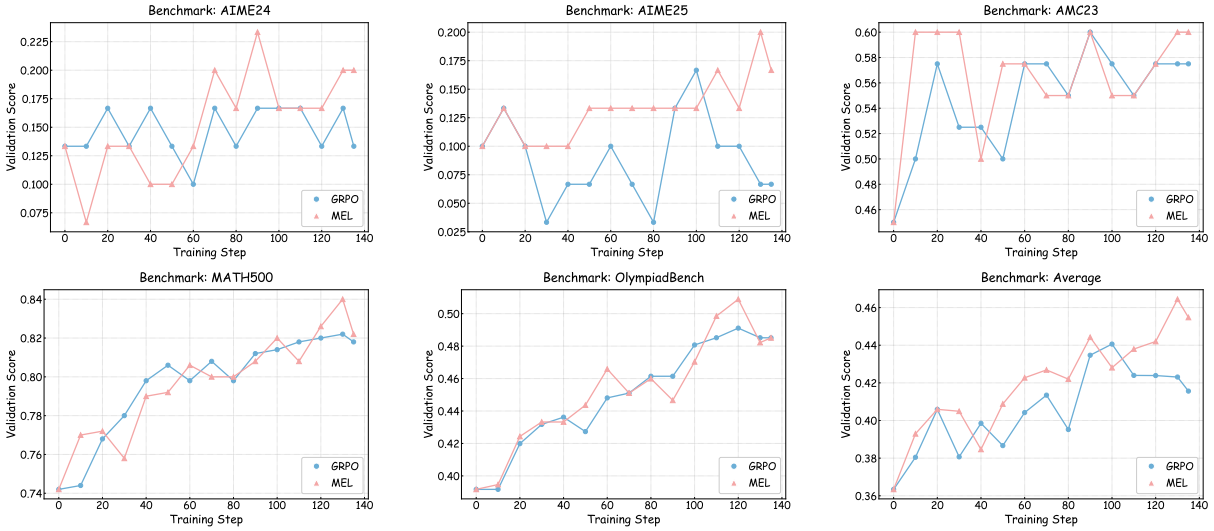

Figure 8: Performance evolution of GRPO and MEL on Qwen3-4B-Base across training steps on multiple benchmarks.

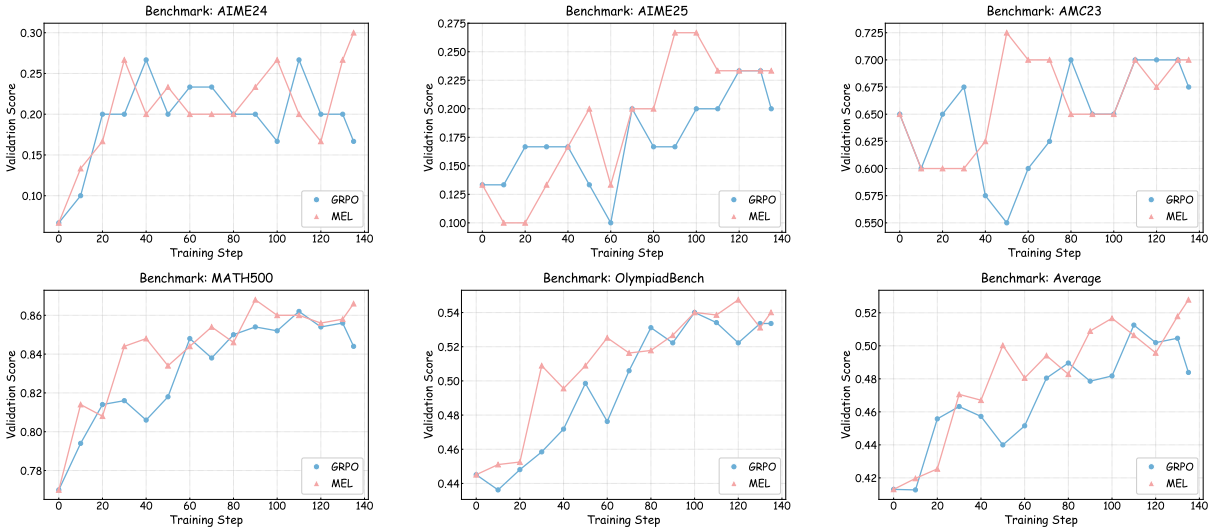

Figure 9: Performance evolution of GRPO and MEL on Qwen3-8B-Base across training steps on multiple benchmarks.

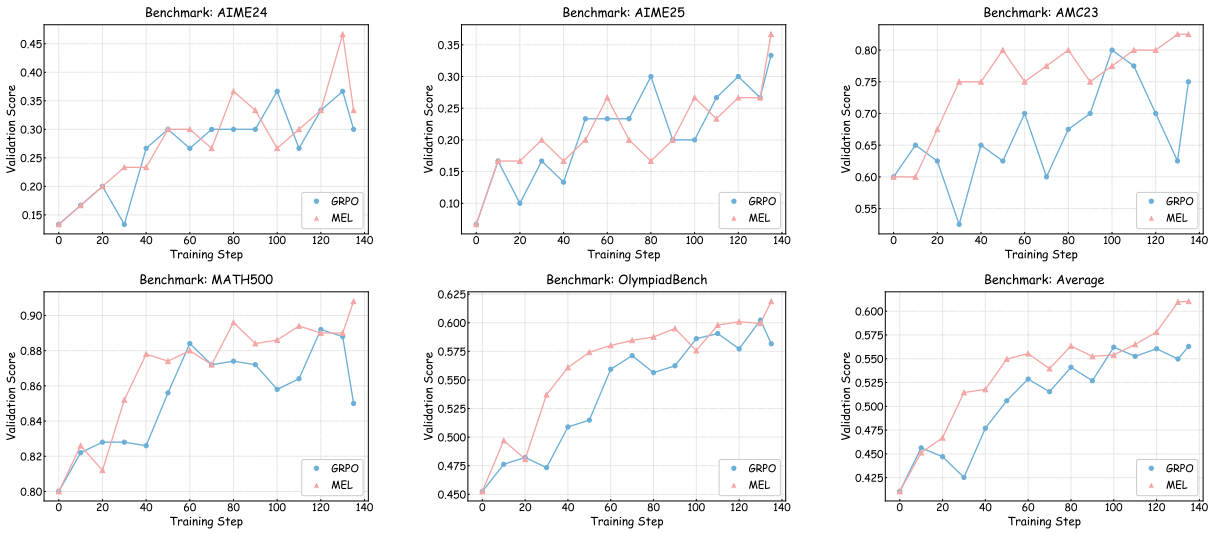

Figure 10: Performance evolution of GRPO and MEL on Qwen3-14B-Base across training steps on multiple benchmarks.

E Prompt Template

Meta-Experience Prompt

You are a **Meta-Cognitive Reasoning Analyst** specializing in self-reflection, error root-cause analysis, and the extraction of generalizable heuristics.
You are provided with multiple solution trajectories for the same problem. Note that the labels **Correct** or **Incorrect** apply to the **final answer**, but the reasoning process itself may contain twists and turns. Your task is to conduct a **deep comparative autopsy** of the thinking processes. You must identify the **structural** differences in cognition that led to success or failure, and synthesize these into abstract principles for future use.
Core Analysis Requirements:

1. **Deep Dive into Correct Trajectories (Resilience & Robustness Analysis):**
   - **Scenario A (Self-Correction):** If you find the reasoning contains initial errors or uncertainties, look for moments of **self-correction**. What triggered the realization? What structural insight allowed the reasoning to pivot back to the right track?
   - **Scenario B (Flawless Execution):** When every step of the reasoning is correct from the start, identify the **Foundational "Immunity"**. What specific definition, clear knowledge representation, or disciplined step-by-step verification prevented this Agent from falling into the traps that the Incorrect Agent fell into?
   - **Goal:** Extract the specific logic validation technique or robust mental representation that saved the solution.
2. **Deep Dive into Incorrect Trajectories (Vulnerability Analysis):**
   - You must identify not only *where* the math/logic went wrong, but also *why* the reasoning drifted.
   - Identify: The **"Bifurcation Point"** where a correct start turned into a hallucination or logic gap.
   - Analyze: The **latent cognitive defect** (e.g., concept conflation, rigid mindset, overlooking edge cases, intuitive bias) that caused the error.
   - Identify: What specific knowledge point or constraint was violated?
3. **Comparative Synthesis:**
   - Contrast the **Solutions and Decision Boundaries**. Why did the successful trajectory avoid the trap that the failed one fell into?
   - What **structural insight** did the winner have that the loser missed? (e.g., The winner treated the problem as a geometric issue, while the loser got stuck in algebra).
4. **Strict Generalization Constraint:**
   - **Forbidden:** Do NOT mention the specific numbers, variables, or exact answer of the current problem in your "Heuristics" or "Reflective Summary".
   - **Required:** Convert specific lessons into **abstract heuristics** (e.g., instead of "The integral of x^2 is...", use "When integrating polynomial functions..."). Formulate them as conditionally triggered rules ("If...Then...", "When dealing with [Concept X]...I should...").

Output Format (Strict Adherence Required)
1. Failure Resolution Path & Error Pattern Recognition (Mandatory for incorrect samples)

- **Failure Point:** Identify the exact step where logic diverged. Did it start correctly? Where did the drift happen?

- **Latent Cognitive Pattern:** Reveal the deep-seated reasoning defect. Was it a bias? A missing prerequisite? A misunderstanding of the prompt's intent? Do not list surface-level calculation errors.

2. Analysis of Success Factors (Mandatory for correct samples)

- **Reasoning Pivot (If applicable):** If the path involved self-correction, describe the moment of realization and the strategy used to fix it.
- **Robustness Factor (If flawless):** If the path was perfect, explain the fundamental concept or structural approach that effectively "immunized" the reasoning against common errors.
- **Reason for Effectiveness:** Why did this perspective work? What fundamental logic did it satisfy?

3. First-Person Reflective Summary (Mandatory)
Write a meta-cognitive reflection from the first-person perspective ("I").

- **Review:** Briefly review the thinking process differences.
- **Insight:** Discuss the specific knowledge point or cognitive habit that was critical.
- **Action:** Explain how you will restructure your approach to avoid the identified "Internal Reasoning Defects" in the future.

Focus on the "How" of thinking, not the "What" of the answer.
4. Subject Heuristics (Internalized Experience) (Mandatory)

- **[Pattern Name]:** If [abstract condition] occurs, then [abstract action]...
- **[Pattern Name]:** When dealing with [concept type], I must strictly verify [constraint]...

(Note: These must be applicable to *future* problems of a similar class, completely stripped of this problem's specifics.)
Here the question and the corresponding solutions. <question> {question} </question>

Solution 1:
<answer> {error_ans} </answer>
<judge> Incorrect </judge>

Solution 2:
<answer> {correct_ans} </answer>
<judge> Correct </judge>

Empirical Validation Prompt

Prior study has provided some internal reference information relevant to this question, including the **key approaches, steps, and reasoning** needed for a correct solution; the **typical reasoning biases**, **logical flaws**, or **pitfalls** that appear in incorrect solutions; and various **heuristic insights** on how to complete this problem more effectively.

{experience}

Now, please fully internalize this information as your own experience, then independently think through the problem in detail and produce a complete answer.
Note:

- You must perform full, in-depth reasoning internally and arrive at the final answer while making full use of the information above.

Answer the following question:
{question}