LLM-as-a-Coach: Experiential Learning for Non-Verifiable Tasks

Tianzhu Ye* **Li Dong***
Guanheng Chen[1] **He Zhu**[2] **Xun Wu** **Shaohan Huang** **Furu Wei**
Microsoft Research
[1] Tsinghua University
[2] Peking University
https://aka.ms/GeneralAI

Reinforcement learning (RL) on open-ended tasks compresses an LLM's rubric-based evaluation into a scalar reward, discarding rich textual feedback and conflating responses with distinct quality profiles. We propose **Experiential Learning** (EL), which repurposes the feedback model from an LLM-as-a-Judge into an LLM-as-a-Coach. The coach distills its assessment of each on-policy response into transferable experiential knowledge, which conditions a teacher model and is internalized by the policy through on-policy context distillation. Compared with scalar rewards, this higher-bandwidth feedback channel provides dense supervision and preserves fine-grained preferences among high-quality responses. Across two policy families, with feedback from the policy itself or a proprietary model, EL consistently outperforms rubric-based RL on held-out and unseen open-ended tasks. Notably, EL generalizes better beyond the training distribution, and mitigates reward hacking. These findings establish experiential knowledge as a richer and more generalizable learning signal for post-training on non-verifiable tasks.

Code: aka.ms/el-code

1 Introduction

Reinforcement learning [SB+98] has become a standard approach for post-training language models. It is particularly effective on verifiable tasks, where mathematical answers can be checked, programs can be executed, and constrained outputs can be validated automatically [LMP+24]. Open-ended tasks, however, rarely have a unique correct answer. The quality of an explanation, recommendation, or creative response instead depends on multiple dimensions, such as factuality, relevance, completeness, organization, and style. Training on such tasks therefore often relies on an LLM-as-a-Judge that evaluates responses according to natural-language rubrics [ZCS+23, GWL+25].

Standard RL reduces this evaluation to a scalar reward. Although the evaluator may identify specific strengths, failures, and possible improvements, only the final score is used for optimization, while the remaining textual analysis is discarded. This creates an information bottleneck: responses receiving the same score become indistinguishable to the optimizer, even when the evaluator expresses meaningful preferences among them. The problem is especially pronounced near the top of the reward scale, where many satisfactory responses may receive the same maximum score despite differing in subtle but important ways.

* Equal contribution.

[1] This is Part III of Experiential Learning series. Part I: On-Policy Context Distillation for Language Models. Part II: Online Experiential Learning for Language Models.

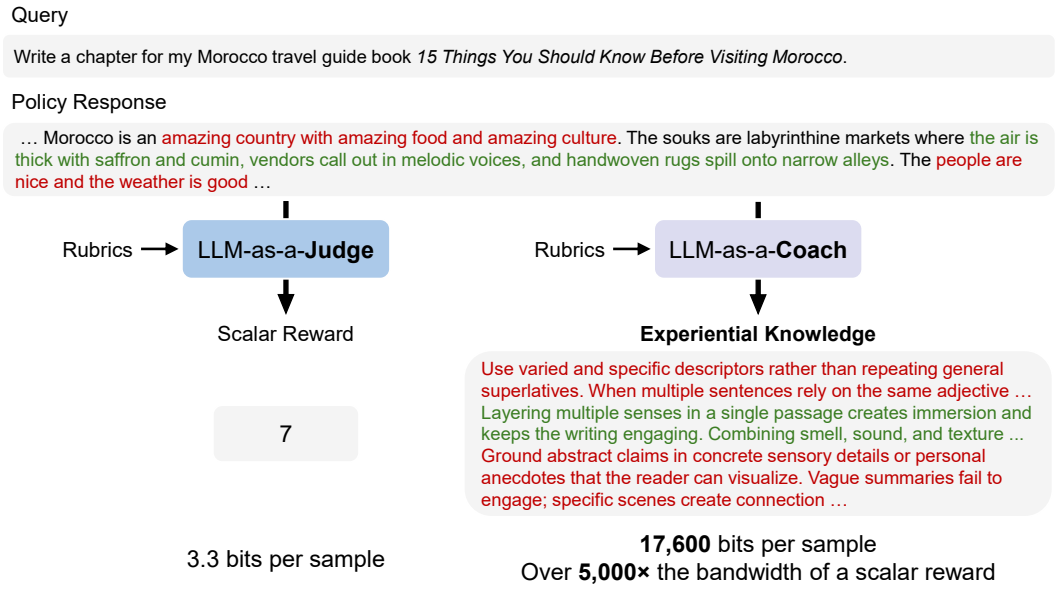

Figure 1: Comparison of the learning signals in RL and EL. RL compresses evaluation into a scalar reward, while EL learns from experiential knowledge generated by an LLM-as-a-Coach. This richer signal speeds up learning and distinguishes top-performing responses that scalar rewards conflate on non-verifiable tasks. Red and green highlights connect weaknesses and strengths in the response to corresponding corrective and positive knowledge. RL bandwidth assumes a discrete 1–10 reward.

In this work, we introduce **Experiential Learning** (EL), which learns directly from the rich knowledge produced during evaluation. As illustrated in Figure 1, EL repurposes the feedback model from an *LLM-as-a-Judge* into an *LLM-as-a-Coach*. Given a prompt, an on-policy response, and a set of rubrics, the coach analyzes the response and distills its assessment into *experiential knowledge*: general and transferable guidance for producing better responses to similar tasks. Rather than preserving instance-specific corrections, this knowledge captures reusable strategies derived from the policy's own experience.

To internalize this knowledge, EL uses on-policy context distillation [YDW+26]. The extracted experience is provided as context to a teacher model, inducing a distribution that reflects the coach's guidance. Along responses sampled from the policy, we then minimize the token-level reverse KL divergence between the policy and the context-conditioned teacher. This converts natural-language feedback into dense distributional supervision and consolidates it into the policy parameters, so neither the coach nor the experiential context is needed at inference time.

Experiential knowledge provides a potentially higher-bandwidth feedback channel than a scalar reward, as illustrated in Figure 1. A discrete 1–10 score has a theoretical maximum bandwidth of 3.3 bits per sample; a learned reward head producing a bfloat16 (bf16) value provides at most 16 representational bits. Under the same alphabet-size calculation, an experiential context of 1024 tokens drawn from a vocabulary of size 150,000 has a theoretical upper bound of 17,600 bits—over $1{,}000\times$ the maximum bandwidth of a scalar reward. This comparison is a feedback-bandwidth intuition rather than a measure of usable supervision. This information is translated by the context-conditioned teacher into dense, per-token supervision, accelerating learning when on-policy samples are limited. In the well-converged regime, where RL has saturated the available reward signal, scalar feedback conflates high-quality responses assigned the same reward level. Experiential knowledge retains these fine-grained preferences, enabling EL to further improve responses on non-verifiable tasks involving multiple quality dimensions and nuanced trade-offs.

We evaluate EL on open-ended instruction-following tasks across two policy families, using either the initial frozen policy checkpoint or a proprietary model accessed through an API as the feedback model. Across a held-out in-distribution evaluation and multiple unseen benchmarks, EL consis-

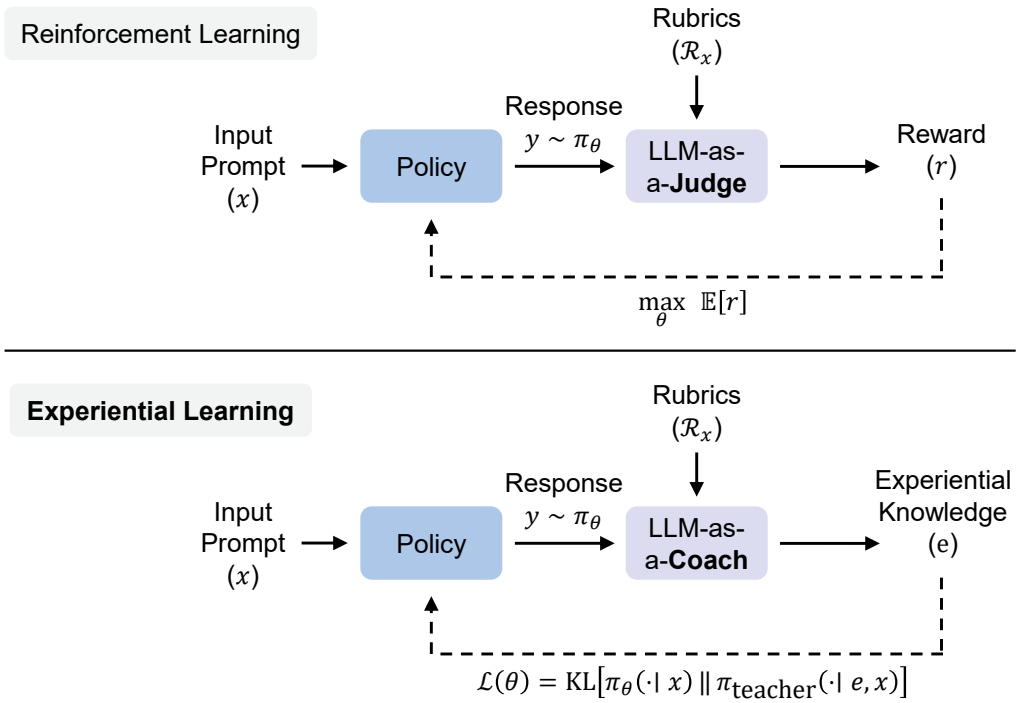

Figure 2: Comparison of RL and EL. Reinforcement learning updates the policy by maximizing rewards assigned by an LLM-as-a-Judge, whereas experiential learning learns from experiential knowledge provided by an LLM-as-a-Coach under the same rubrics. Experiential knowledge provides richer learning signals for policy improvement than scalar rewards.

tently improves over rubric-based RL, demonstrating the effectiveness of experiential supervision across different policy and feedback models.

Our results further suggest that scalar-reward optimization is susceptible to training-set reward overoptimization, a form of reward hacking [SHKK22], whereas the improvements from EL transfer more effectively beyond the training distribution. A controlled distribution-matching analysis illustrates that, under its simplified setup, quantized scalar feedback preserves a binary target but discards finer distinctions in a multimodal target, whereas distributional guidance more faithfully recovers the target distribution. We also find that distilled experiential knowledge is more effective than using raw critiques or rubrics. Iterative teacher updates improve task performance, while on-policy distillation over general-domain prompts mitigates forgetting in out-of-distribution capabilities.

2 Method

We present **Experiential Learning** (EL) to address the problem of training a language model policy π_θ on non-verifiable open-ended tasks, where no ground-truth answer exists and response quality is assessed by another model. Given a prompt x, the policy generates a response y. A feedback LLM M then assesses the response against a set of rubrics $\mathcal{R}_x$ that decompose quality into checkable criteria. We denote the full output of M as $M(x, y, \mathcal{R}_x)$, from which the policy is optimized. In standard RL, M acts as an LLM-as-a-Judge and reduces its assessment to a scalar reward, discarding all other textual content. In contrast, EL uses M as an LLM-as-a-Coach that extracts experiential knowledge to guide policy optimization, exposing a potentially higher-bandwidth signal than scalar feedback.

2.1 Preliminaries: RL with an LLM-as-a-Judge

Before introducing the learning objective of EL, we formalize the standard RL baseline, in which the feedback LLM M serves as an *LLM-as-a-Judge* [ZCS+23, LIX+23, APC+24, GCD+26]. For

Algorithm 1 EL: Experiential Learning

Input: Training data $\mathcal{D} = \{(x, \mathcal{R}_x)\}$; Policy π_θ; Teacher π_{teacher}; LLM-as-a-Coach M
Output: Trained policy π_θ
  for each batch $x \sim \mathcal{D}$ **do**
    // On-policy rollout
    Sample responses $y \sim \pi_\theta(\cdot \mid x)$

    // Extract experiential knowledge
    $e \leftarrow \text{Extract\_Knowledge}(M(x, y, \mathcal{R}_x))$

    // Compute token-level reverse KL toward context-conditioned teacher
    $\mathcal{L}(\theta) \leftarrow \frac{1}{|y|} \sum_{t=1}^{|y|} D_{\text{KL}}(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \pi_{\text{teacher}}(\cdot \mid e, x, y_{<t}))$

    // Update policy
    Update θ by minimizing $\mathcal{L}(\theta)$
  end for
  return π_θ

each on-policy response $y \sim \pi_\theta(\cdot \mid x)$, only a scalar reward is extracted, and the remaining textual output is discarded. The policy is trained on a dataset $\mathcal{D}$ of prompts and rubrics to maximize the expected reward:

$$\begin{gathered} \max_\theta \; \mathbb{E}_{(x,\mathcal{R}_x)\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[r \right], \\ \text{where } r = \text{Extract\_Reward}(M(x, y, \mathcal{R}_x)) \end{gathered} \tag{1}$$

We use GRPO [SWZ+24] to implement the RL baseline.

2.2 Experiential Learning with an LLM-as-a-Coach

In the coach interface, EL repurposes the same feedback LLM M as an *LLM-as-a-Coach*. For each on-policy response, M analyzes the rubric-based assessment and distills transferable insights into experiential knowledge e. The policy is then trained to minimize the reverse KL divergence between its own distribution $\pi_\theta(\cdot \mid x)$ and a context-conditioned teacher distribution $\pi_{\text{teacher}}(\cdot \mid e, x)$. The loss function is defined as:

$$\begin{gathered} \mathcal{L}(\theta) = \mathbb{E}_{(x,\mathcal{R}_x)\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} D_{\text{KL}} \left(\pi_\theta(\cdot \mid x, y_{<t}) \,\|\, \pi_{\text{teacher}}(\cdot \mid e, x, y_{<t}) \right) \right], \\ \text{where } e = \text{Extract\_Knowledge}(M(x, y, \mathcal{R}_x)) \end{gathered} \tag{2}$$

The teacher can be either the initial frozen policy checkpoint or an iteratively updated version, where the policy checkpoint at the end of each epoch serves as the teacher for the next. EL uses the experiential knowledge produced by the LLM-as-a-Coach as the learning signal, internalizing its rich distributional information into the policy weights via on-policy context distillation [YDW+26]. Refer to Appendix A.1 for prompt templates.

2.3 From Judge to Coach

The same feedback LLM M and prompt-specific rubrics $\mathcal{R}_x$ support both learning interfaces, but the two interfaces use the resulting assessment differently, as summarized in Table 1.

	LLM-as-a-Judge	**LLM-as-a-Coach (Ours)**
Role	Evaluate: assess response quality	Guide: extract transferable guidance
Learning Signal	Sequence-level scalar score	Experiential knowledge
Feedback Bandwidth	Low	High

Table 1: Comparison of an LLM-as-a-Judge and an LLM-as-a-Coach in their roles, learning signals, and feedback bandwidths. Both use the same feedback LLM and rubrics.

An LLM-as-a-Judge evaluates a response and exposes only a scalar score as the learning signal. An LLM-as-a-Coach instead guides improvement through experiential knowledge. Put simply, a judge tells the policy how well it performed, whereas a coach provides actionable guidance for performing better.

Importantly, this distinction concerns the feedback consumed by the optimizer, not the intrinsic capabilities of M. An LLM-as-a-Judge may produce detailed textual analysis, but standard RL retains only its final score, creating a low-bandwidth learning channel. In EL, the LLM-as-a-Coach distills the same rubric-based assessment into transferable experiential knowledge. A context-conditioned teacher then translates this higher-bandwidth feedback into dense token-level supervision. We develop this bandwidth intuition below.

2.4 Feedback-Bandwidth Intuition

A useful intuition for the advantage of EL over RL comes from the maximum representational bandwidth of their respective feedback channels.

RL In our experiments, we adopt a discrete 1–10 scoring scale following standard rubric-based evaluation practice, whose theoretical maximum bandwidth is $\log_2(10) \approx 3.3$ bits per sample. If rewards are produced in bfloat16 (bf16) using a learned projection head, the maximum representational bandwidth increases to 16 bits.

EL The teacher π_{teacher} is either the initial frozen policy checkpoint or an iteratively updated policy checkpoint. Together, π_θ and π_{teacher} form a closed system whose external learning signal is the experiential knowledge e produced by M. A theoretical upper bound on the bandwidth of this textual channel is determined by e: a context of L tokens drawn from a vocabulary of size $|\mathcal{V}|$ yields up to $L \cdot \log_2(|\mathcal{V}|)$ bits.[2] With L=1024 and $|\mathcal{V}|$=150,000, this amounts to approximately $1024 \times 17.2 \approx 17{,}600$ bits per sample—over **1,000**$\times$ the theoretical maximum bandwidth of a bf16 scalar reward and **5,000**$\times$ of a discrete 1–10 reward.

This potentially wider feedback channel can be advantageous on non-verifiable tasks. For verifiable tasks [LMP+24] with a binary correctness objective, a single bit suffices to represent the reward outcome. However, non-verifiable tasks involve multiple quality dimensions, stylistic preferences, and nuanced trade-offs. In the converged regime where rollouts are abundant and RL has saturated the reward signal, all responses achieving the maximum reward level become **indistinguishable** due to the information bottleneck, limiting further alignment with the fine-grained preferences expressed by the LLM-as-a-Judge. EL bypasses this bottleneck by transmitting richer information through the experiential knowledge context, enabling a **higher-fidelity approximation of the target distribution defined by** M.

3 Experiments

For non-verifiable tasks, on-policy training with rubrics provides a principled way to assess open-ended responses from the policy model. Standard RL uses an LLM-as-a-Judge to evaluate each response against rubrics, producing a scalar reward while discarding the textual feedback. EL instead employs an LLM-as-a-Coach that analyzes each response against the same rubrics and generates experiential knowledge, which is then consolidated into the policy model weights.

3.1 Setup

Training Data and Models We source training prompts from WildChat-IF [ZRH+24, HCZ+26, BYC25], a curated collection of 7500 real user queries emphasizing diverse conversational instructions. For each prompt, we pre-generate a set of evaluation rubrics using GPT-4o. These rubrics decompose response quality into fine-grained, independently checkable criteria. In experiments with iterative teacher model, we mix in prompts from Tulu3 [LMP+24] at a 1:0.25 ratio to preserve

[2] This quantity is the theoretical maximum encoding capacity of the textual channel. It does not measure the effective supervision carried by the experiential knowledge, nor its mutual information with the desired policy improvement; natural-language redundancy and imperfect knowledge extraction can make the usable information substantially smaller.

Policy + Feedback	Method	WildChat	AlpacaEval-v2	WildBench	ArenaHard-v2	CW-v3
		Score	Win Rate (%)	Score	Win Rate (%)	Score
Qwen3-8B + Qwen3-8B	Base Model	78.1	34.5	17.6	29.1	73.8
	RL	79.2	37.3	18.4	30.5	74.3
	EL	**80.0**	**40.0**	**21.8**	**31.0**	**76.0**
Qwen3-8B + GPT-4o	RL	80.4	**38.2**	19.2	31.2	74.4
	EL	**80.7**	37.4	**22.0**	**31.9**	**75.3**
OLMo-3-7B + OLMo-3-7B	Base Model	75.7	43.6	39.0	21.8	78.7
	RL	76.0	45.9	42.1	23.3	**78.8**
	EL	**77.1**	**50.8**	**47.5**	**26.2**	**78.8**
OLMo-3-7B + GPT-4o	RL	76.8	44.7	40.3	22.5	77.8
	EL	**78.2**	**48.5**	**46.2**	**25.1**	**78.0**

Table 2: Evaluation scores of RL and EL. "Policy + Feedback" denotes the policy model paired with the feedback model, which acts as an LLM-as-a-Judge for RL or an LLM-as-a-Coach for EL. CW-v3 is short for CreativeWritingV3. EL outperforms RL on most benchmarks across all model combinations.

general capabilities. These prompts use the initial frozen policy checkpoint as the OPD teacher and carry no extra context. We filter the Tulu3 SFT mixture to exclude WildChat-overlapping sources. We use Qwen3-8B [YLY+25] (non-thinking mode) and OLMo-3-7B-Instruct [OEB+25] as the policy models. We use either the initial frozen policy checkpoint or GPT-4o as the LLM-as-a-Judge for RL and the LLM-as-a-Coach for EL.

Training We use Rubric-as-Reward [GWL+25] implemented with GRPO [SWZ+24] as the baseline, which optimizes the policy using a 1–10 rating from the LLM-as-a-Judge as the scalar reward. For EL, the LLM-as-a-Coach generates experiential knowledge based on the rubrics and the response, which is then prepended as context to the teacher model. The teacher can be the initial frozen policy checkpoint or an iteratively updated version where the policy checkpoint at the end of each epoch becomes the teacher for the next epoch. Both methods use a batch size of 256 prompts and 8 sampled responses per prompt. Learning rate is set to 1e-6. The maximum prompt length is 8192 tokens, and the maximum response length is 4096 tokens. Rollout temperature is set to 1. Training runs for 3 epochs over the dataset and checkpoints are saved every 10 steps. Reverse KL is computed over the top 256 vocabulary tokens ranked by the student model's predicted probabilities without re-normalization to 1. Refer to Appendix A.1–A.5 for prompt templates and examples.

Evaluation We evaluate on a held-out test set of 250 WildChat-IF prompts using GPT-4o as a rubric-conditioned LLM-as-a-Judge, sampling 4 responses per prompt. We also measure transfer to four open-ended benchmarks not seen during training: AlpacaEval v2.0 [LZD+23], WildBench [LDC+25], ArenaHard v2.0 [LCF+24], and CreativeWritingV3 [Pae25]. GPT-4o serves as the evaluator across all benchmarks. The first three are pairwise evaluations against GPT-4-Turbo references: we report win rate (%) for AlpacaEval v2.0 and ArenaHard v2.0, and a preference score in $[-100, 100]$ for WildBench. CreativeWritingV3 uses direct rubric-based scoring, reported in $[0, 100]$. We additionally cross-checked the main conclusions with a stronger proprietary evaluator and observed consistent relative trends. We therefore report GPT-4o evaluations throughout to reduce evaluation cost.

3.2 Main Results

We present the main results in Table 2. In the "Policy + Feedback" column, the first model is the policy being optimized, and the second is the feedback LLM, acting as an LLM-as-a-Judge for RL and an LLM-as-a-Coach for EL. We consider two choices of feedback LLM: the initial frozen policy checkpoint and the closed-source GPT-4o. We report the average score (or win rate) across the top-3 performing checkpoints.

Both RL and EL are trained on WildChat-IF prompts and evaluated on the held-out WildChat test set together with four additional benchmarks. EL consistently outperforms RL on the WildChat

evaluation and generalizes better to unseen benchmarks, with particularly large gains on AlpacaEval v2.0 and WildBench. This advantage holds whether the feedback LLM is the initial frozen policy checkpoint or the closed-source GPT-4o, indicating that the benefit of EL stems from its richer information channel rather than from a specific feedback LLM. We further observe that using GPT-4o as the feedback LLM improves WildChat scores for both RL and EL relative to using the initial frozen policy checkpoint.

3.3 Experiential Learning Generalizes Better

We sample a subset of 1000 WildChat training examples and evaluate on it after training completes. We compare the results with those on WildChat test set and two benchmarks, as shown in Figure 3. Although EL achieves a smaller gain than RL on the training set, it obtains higher scores on the WildChat test set, AlpacaEval V2.0, and WildBench.

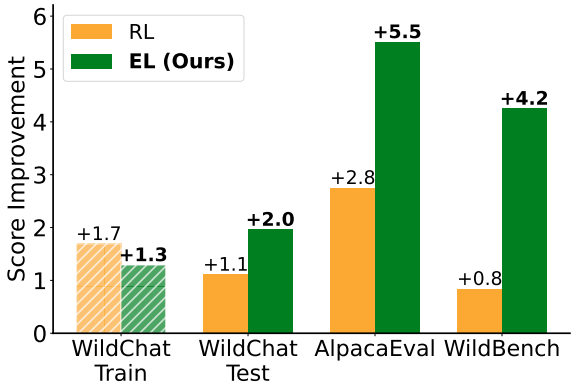

Figure 3: Score improvement over the base model. "WildChat Train" is evaluated on a subset of the training data after training completes. EL generalizes better than RL despite achieving lower training-set scores.

We hypothesize that this difference arises in part from the feedback-bandwidth difference between RL and EL. RL maximizes a scalar reward that carries limited information, allowing the policy to drift toward patterns that inflate training rewards without transferring—a manifestation of reward hacking [SHKK22]. EL never receives a scalar score from the LLM-as-a-Coach during training. Instead, its learning signal is the full distributional shift induced by the experiential knowledge context, which steers the policy toward more transferable behaviors. We use Qwen3-8B as both the policy model and the feedback LLM in Figure 3. We observe a similar phenomenon when replacing the Qwen3-8B feedback LLM with GPT-4o.

3.4 Ablations

Iterative Teacher We ablate the teacher update strategy in Table 3, using Qwen3-8B as both the policy model and the LLM-as-a-Coach. In the *Fixed Teacher* setting, the teacher remains the initial frozen policy checkpoint throughout training, which is also the default setting of our experiments. In the *Iterative* setting, the policy checkpoint at the end of each epoch becomes the teacher for the next epoch. In the *Iterative+General* setting, we mix Tulu3 [LMP+24] prompts into WildChat-IF at a 1:0.25 ratio. These general prompts carry no experiential context and use the initial frozen policy checkpoint as the OPD teacher [MWZ+26, LL25]; we filter the Tulu3 SFT mixture to exclude WildChat-overlapping sources.

Configuration	Score	OOD Acc.
Base Model	78.1	83.5
RL	79.2	81.3
Fixed Teacher	80.0	83.1
Iterative	80.7	74.9
Iterative+General	80.7	79.9

Table 3: Ablation on iterative teacher updates. "Score" is the WildChat test set score averaged over the top-3 performing checkpoints and "OOD Acc." is the IFEval accuracy averaged over the same checkpoints.

Iteratively updating the teacher improves the WildChat score from 80.0 to 80.7, but substantially reduces IFEval accuracy, indicating forgetting on out-of-distribution instruction following. Adding general-prompt OPD from the initial frozen policy checkpoint recovers much of this loss, improving IFEval from 74.9 to 79.9 while preserving the same WildChat score. This makes general-prompt OPD a useful stabilizer when applying iterative teacher updates.

Teacher Context We ablate the format of context provided by M in Table 4, using Qwen3-1.7B as the policy model and Qwen3-4B as the LLM-as-a-Coach. "Score" denotes the WildChat test set score evaluated by GPT-4o and averaged over the top-3 performing checkpoints. "OOD Acc." denotes the IFEval [ZLM+23] accuracy for evaluating out-of-distribution capability, averaged over the same three checkpoints.

Configuration	Score	OOD Acc.
Base Model	64.8	68.0
RL	67.7	67.6
Full Critique	67.9	64.6
Rubrics Only	68.1	65.6
Multiple-Choice	66.3	68.5
Default EL	**68.6**	**68.8**

Table 4: Ablation of teacher context on Qwen3-1.7B. "Score" is WildChat test set score (Top-3 Average); "OOD Acc." is the IFEval accuracy averaged over the same three checkpoints.

Full Critique prepends the entire scoring output from M, including the per-rubric analysis, as context for the teacher, without performing experiential knowledge extraction. *Rubrics Only* provides the rubrics as context without any response-specific feedback [RMW+26, GCZ+26]. *Multiple-Choice* asks the LLM-as-a-Coach to select one of 10 predefined improvement directives (e.g., "Improve factual accuracy"), compressing feedback to a single categorical choice. The maximum feedback bandwidth of this configuration reduces to $\log_2 10 \approx 3.3$ bits, equivalent to that of a 1–10 scalar reward in RL. We also experimented with including the response from the policy model alongside the full scoring output in the context. This configuration diverges within a few training steps. Prompt templates for all configurations are provided in Appendix A.2.

All variants improve over the base model on the WildChat test set, and the default EL setting that extracts transferable experiential knowledge performs best. *Full Critique* and *Rubrics Only* degrade IFEval accuracy because evaluation-oriented context biases the teacher toward a critiquing distribution rather than a task-solving distribution. The policy then imitates this misaligned behavior, leading to forgetting on OOD tasks. *Multiple-Choice* preserves OOD performance but offers limited gains, consistent with its lower feedback bandwidth.

3.5 Analysis

Feedback Bandwidth Comparison Table 5 gives a qualitative comparison of the feedback bandwidth of three learning methods. RL receives a scalar reward per sample, with $O(1)$ representational bandwidth regardless of response length. On-policy distillation (OPD) from a stronger teacher can provide feedback that scales as $O(N)$, where N is the response length, since the teacher's per-token distribution encodes knowledge from its larger parameter space. In EL, the student and the teacher form a closed system. The external learning signal comes entirely from the experiential knowledge context of length L produced by the coach, so its theoretical maximum textual bandwidth scales as $O(L)$. Although both OPD and EL use distillation, their information sources differ: OPD draws from the teacher's superior parametric knowledge, whereas EL draws from the coach's rubric-based assessment externalized as text.

For RL, a discrete 1–10 score has a theoretical maximum bandwidth of 3.3 bits per sample. A learned reward head that produces a BF16 value has a maximum representational bandwidth of 16 bits, but the reward model uses only a small subspace of this range in practice. The sparse signal landscape can make it easier for the policy to exploit by reward hacking rather than providing more usable information.

Method	Feedback Source	Bandwidth
Reinforcement Learning (RL)	Sequence-level scalar reward from LLM-as-a-judge	$O(1)$
On-Policy Distillation (OPD)	Larger teacher parameters	$O(N)$
On-Policy Self-Distillation (OPSD)	Privileged information, e.g., ground-truth answers, and hints	$O(L)$
Experiential Learning (EL)	Experiential knowledge context from LLM-as-a-coach	$O(L)$

Table 5: Qualitative comparison of feedback bandwidth across learning methods. N: response length; L: experiential context length.

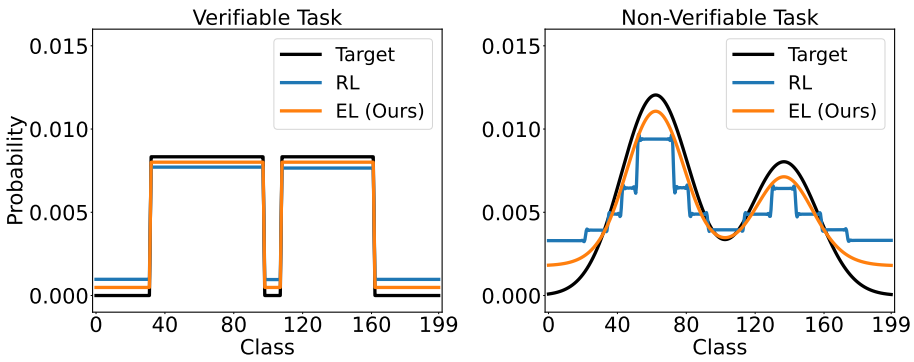

Figure 4: Controlled distribution-matching analysis. Left: for a binary target, RL with binary rewards and direct distributional matching converge to similar distributions. Right: for a bimodal target, the constructed five-level reward produces a staircase distribution, while direct distributional matching recovers the smooth target shape.

Distribution-Matching Analysis of Feedback Quantization We use a controlled toy setting to analyze how quantizing feedback into scalar levels can affect distribution matching. This construction isolates the representation of feedback: it is not intended as a faithful simulation of the full RL and EL training pipelines, nor as a proof of a fundamental limitation shared by all scalar-reward methods. Rather, it provides intuition for the information discarded when a fine-grained target is mapped to a small number of reward levels. The results are shown in Figure 4.

Suppose the feedback LLM, acting as either an LLM-as-a-Judge or an LLM-as-a-Coach, has an internal ideal distribution p^* over a discrete vocabulary of $V{=}200$ tokens that it wishes the policy to learn. We train a categorical policy $\pi_\theta = \text{softmax}(\theta)$ to approximate p^* via on-policy sampling: at each step, a token y is sampled from π_θ and a training signal is received. We compare two feedback mechanisms: (1) **RL**: the policy receives a scalar reward quantized to L levels: $r(y) = \left\lfloor \frac{p^*(y)}{\max_{y'} p^*(y')} \cdot (L-1) + 0.5 \right\rfloor, r(y) \in \{0, 1, \ldots, L{-}1\}$ and is optimized via REINFORCE [Wil92] with a KL penalty toward the uniform distribution. (2) **Distributional guidance (an idealized proxy for EL)**: assuming that the experiential knowledge carries sufficient information to specify the target distribution, we directly minimize the reverse KL divergence $\text{KL}(\pi_\theta \| p^*)$. We consider two scenarios: a *verifiable* task where p^* is binary (correct/incorrect tokens, $L{=}2$), and a *non-verifiable* task where p^* is a bimodal Gaussian mixture quantized into $L{=}5$ reward levels.

As shown in Figure 4 (left), on the constructed binary task both RL and direct distributional matching converge to nearly identical distributions matching p^*. Under this target and objective, responses are either correct or incorrect, and no finer-grained distinction is required; binary rewards therefore preserve the information needed for the specified target.

However, under the five-level reward construction (Figure 4, right), a clear gap emerges. Direct distributional matching recovers the smooth bimodal shape of p^*, while RL converges to a staircase-shaped distribution. Specifically, tokens at the peak plateau that receive the same maximum reward learn identical probability under this RL objective, even though p^* assigns different likelihoods to them. Within this setup, the quantized scalar reward cannot further distinguish among tokens assigned the same top score. Direct distributional guidance retains those fine-grained distinctions and therefore matches p^* without the quantization artifacts.

4 Discussion

Reward Hacking EL mitigates reward hacking that arises from the information bottleneck between the feedback model and the policy: when an LLM-as-a-Judge compresses its assessment into a scalar reward, the policy can exploit patterns that inflate the score without genuinely improving response quality. By transmitting richer information through experiential knowledge, EL reduces this failure mode. We note that a separate source of reward hacking can occur when the feedback model itself is biased or miscalibrated. Since EL addresses the *feedback bandwidth* between the

feedback model and the policy rather than the quality of the feedback model itself, improving the calibration and accuracy of the feedback model remains a complementary direction.

Distinction from Reinforcement Learning Although the reverse KL objective of EL can be decomposed into a token-level advantage-weighted form, it differs from RL in its optimization target. RL maximizes a scalar reward: its optimal policy concentrates probability mass on the highest-reward responses. EL, through on-policy context distillation, instead minimizes divergence from a context-conditioned teacher distribution. RL is therefore reward-seeking, while EL is distribution-matching. The former exploits any feature correlated with higher reward, whereas the latter stays close to the teacher's distributional shape.

Training Cost and Efficiency Both RL and EL spend most of their end-to-end training time in the online data-generation stage: sampling rollouts from the policy and invoking the feedback LLM as an LLM-as-a-Judge or LLM-as-a-Coach. These two expensive components are shared by both methods. EL additionally requires a context-conditioned teacher forward pass and token-level distillation, whereas RL uses reward-based GRPO updates. In our setting, the cost of these method-specific updates is small relative to the shared rollout and feedback-generation cost.

5 Related Work

Learning from Experience Recent work advocates learning from experience through interactions with the environment [SS25]. Such experience has been shown to accelerate future learning [ZCL+25] and enable reasoning agents to discover effective strategies through self-play and reflection [WWX25]. For language models, prior work leverages interaction histories by reflecting on past failures [SCG+23], storing reusable knowledge in external memory [ZHX+24], or optimizing with multi-turn textual feedback [YBB+24, ATS+25, LBF25]. More recently, critiques have been incorporated into RL objectives by conditioning policy optimization on critique contexts [ZZS+25, HCZ+26]. However, these methods remain off-policy with respect to the original prompt, and the learning signal is ultimately bottlenecked by scalar rewards.

Context Distillation and Self-Distillation Context distillation aims to compress knowledge provided in context into model parameters, eliminating inference-time context processing [ABC+21, SKZ22, CCL25]. To mitigate the exposure bias of off-policy training, on-policy distillation trains the student on its own generated trajectories [GDWH24, LL25, AVZ+24]. Building on these ideas, on-policy context distillation (OPCD) performs context distillation with a context-conditioned teacher under on-policy training [YDW+26]. A closely related line of work is on-policy self-distillation, where the teacher is similarly conditioned on context (e.g., solutions, demonstrations, or environmental feedback), but is typically instantiated by the student model itself [ZXL+26, HLB+26, SDHA26, PVG+26]. Recent advances have extended this paradigm to combining RLVR [YQS+26, LYF+26, KJLY26], integrated it into agentic tasks [WWX+26, WZS+26, YWL+26], introduced more improved training algorithms [XSZ+26, ZZS+26, ZDP+26, WCL+26, SXZ+26, YLX+26], and adapted it for online learning [YDD+26, BHP+26, WCJ+26]. Concurrently, a growing body of work has investigated the mechanisms underlying both its empirical success and its potential degradation [LZH+26, KLK+26, ZBZ+26].

6 Conclusion

We introduced **Experiential Learning** (EL), a framework for post-training language models on non-verifiable tasks using rich experiential knowledge rather than scalar rewards. By repurposing the feedback model as an LLM-as-a-Coach, EL distills rubric-based assessments into transferable guidance and internalizes it through on-policy context distillation. This higher-bandwidth feedback channel provides dense supervision and preserves fine-grained preferences that scalar rewards discard. Across different policy families and feedback models, EL consistently outperforms rubric-based RL on held-out and unseen open-ended tasks while exhibiting stronger generalization beyond the training distribution. Our findings highlight experiential knowledge as an informative and transferable learning signal, opening directions for scalable iterative learning, improved knowledge extraction, and broader post-training of language models on complex open-ended tasks.

References

[ABC+21] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, T. J. Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova Dassarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, John Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *ArXiv*, abs/2112.00861, 2021.

[APC+24] Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*, 2024.

[ATS+25] Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

[AVZ+24] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

[BHP+26] Thomas Kleine Buening, Jonas Hübotter, Barna Pásztor, Idan Shenfeld, Giorgia Ramponi, and Andreas Krause. Aligning language models from user interactions. In *Continual Adaptation at Scale: Towards Sustainable AI Workshop*, 2026.

[BYC25] Adithya Bhaskar, Xi Ye, and Danqi Chen. Language models that think, chat better. In *First Workshop on Foundations of Reasoning in Language Models*, 2025.

[CCL25] Bowen Cao, Deng Cai, and Wai Lam. Infiniteicl: Breaking the limit of context window size via long short-term memory transformation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11402–11415, 2025.

[GCD+26] Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning models. *Advances in Neural Information Processing Systems*, 38:150477–150510, 2026.

[GCZ+26] Siyi Gu, Jialin Chen, Sophia Zhou, Arman Cohan, and Rex Ying. Rethinking reward supervision: Rubric-conditioned self-distillation. *arXiv preprint arXiv:2606.19327*, 2026.

[GDWH24] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-policy distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[GWL+25] Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean M Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025.

[HCZ+26] Lei Huang, Xiang Cheng, Chenxiao Zhao, Guobin Shen, Junjie Yang, Xiaocheng Feng, Yuxuan Gu, Xing Yu, and Bing Qin. Bootstrapping exploration with group-level natural language feedback in reinforcement learning. *arXiv preprint arXiv:2603.04597*, 2026.

[HLB+26] Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026.

[KJLY26] Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled rlvr. *arXiv preprint arXiv:2605.10781*, 2026.

[KLK+26] Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026.

[LBF25] Yoonho Lee, Joseph Boen, and Chelsea Finn. Feedback descent: Open-ended text optimization via pairwise comparison. *arXiv preprint arXiv:2511.07919*, 2025.

[LCF+24] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. In *Forty-second International Conference on Machine Learning*, 2024.

[LDC+25] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations*, 2025.

[LIX+23] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 2511–2522, 2023.

[LL25] Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. https://thinkingmachines.ai/blog/on-policy-distillation.

[LMP+24] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[LYF+26] Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026.

[LZD+23] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[LZH+26] Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huan-ang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026.

[MWZ+26] Wenhan Ma, Jianyu Wei, Liang Zhao, Hailin Zhang, Bangjun Xiao, Lei Li, Qibin Yang, Bofei Gao, Yudong Wang, Rang Li, et al. Mopd: Multi-teacher on-policy distillation for capability integration in llm post-training. *arXiv preprint arXiv:2606.30406*, 2026.

[OEB+25] Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

[Pae25] Samuel J Paech. Eq-bench creative writing benchmark v3, 2025.

[PVG+26] Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models, 2026.

[RMW+26] MohammadHossein Rezaei, Anas Mahmoud, Zihao Wang, Utkarsh Tyagi, Advait Gosai, Razvan-Gabriel Dumitru, Aakash Sabharwal, Bing Liu, and Yunzhong He. Rubric-guided self-distillation: Post-training without rubric verifiers. *arXiv preprint arXiv:2606.12507*, 2026.

[SB+98] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[SCG+23] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

[SDHA26] Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026.

[SHKK22] Joar Max Viktor Skalse, Nikolaus HR Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. In *Proceedings of NeurIPS*, 2022.

[SKZ22] Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022.

[SS25] David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1:11, 2025.

[SWZ+24] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[SXZ+26] Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

[WCJ+26] Yinjie Wang, Xuyang Chen, Xiaolong Jin, Mengdi Wang, and Ling Yang. Openclaw-rl: Train any agent simply by talking. *arXiv preprint arXiv:2603.10165*, 2026.

[WCL+26] Xun Wang, Ruishuo Chen, Zhuoran Li, Yu Chen, and Longbo Huang. When context returns: Toward robust internalization in on-policy distillation. *arXiv preprint arXiv:2606.11627*, 2026.

[Wil92] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 1992.

[WWX25] Sai Wang, Yu Wu, and Zhongwen Xu. Cogito, ergo ludo: An agent that learns to play by reasoning and planning. *arXiv preprint arXiv:2509.25052*, 2025.

[WWX+26] Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, et al. Skill-sd: Skill-conditioned self-distillation for multi-turn llm agents. *arXiv preprint arXiv:2604.10674*, 2026.

[WZS+26] Jiaqi Wang, Wenhao Zhang, Weijie Shi, Yaliang Li, and James Cheng. Tcod: Exploring temporal curriculum in on-policy distillation for multi-turn autonomous agents. *arXiv preprint arXiv:2604.24005*, 2026.

[XSZ+26] Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, and Zhipeng Wang. Paced: Distillation and on-policy self-distillation at the frontier of student competence. *arXiv preprint arXiv:2603.11178*, 2026.

[YBB+24] Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*, 2024.

[YDD+26] Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Online experiential learning for language models. *arXiv preprint arXiv:2603.16856*, 2026.

[YDW+26] Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*, 2026.

[YLX+26] Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026.

[YLY+25] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[YQS+26] Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026.

[YWL+26] Shuo Yang, Jinyang Wu, Zhengxi Lu, Yuhao Shen, Fan Zhang, Lang Feng, Shuai Zhang, Haoran Luo, Zheng Lian, Zhengqi Wen, et al. Opid: On-policy skill distillation for agentic reinforcement learning. *arXiv preprint arXiv:2606.26790*, 2026.

[ZBZ+26] Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026.

[ZCL+25] Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, et al. Agent learning via early experience. *arXiv preprint arXiv:2510.08558*, 2025.

[ZCS+23] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

[ZDP+26] Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models. *arXiv preprint arXiv:2604.17535*, 2026.

[ZHX+24] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.

[ZLM+23] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

[ZRH+24] Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild. In *The Twelfth International Conference on Learning Representations*, 2024.

[ZXL+26] Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models, 2026.

[ZZS+25] Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025.

[ZZS+26] Qiyong Zhong, Mao Zheng, Mingyang Song, Xin Lin, Jie Sun, Houcheng Jiang, Xiang Wang, and Junfeng Fang. Sod: Step-wise on-policy distillation for small language model agents. *arXiv preprint arXiv:2605.07725*, 2026.

A Details of Experiments

A.1 Prompt Templates

For EL, the LLM-as-a-Coach M receives the same inputs but is additionally asked to distill transferable experiential knowledge. We use the prompt template in Figure 5. The experiential knowledge is extracted from the `<experience>` tags.

You are an expert evaluator. Given a user prompt, a generated response, and a list of quality rubrics, please rate the overall quality of the response on a scale of 1 to 10 based on how well it satisfies the rubrics.
Consider all rubrics holistically when determining your score. A response that violates multiple rubrics should receive a lower score, while a response that satisfies all rubrics should receive a higher score.

<prompt>
{instruction}
</prompt>

<response>
{response}
</response>

<rubrics>
{rubric_list_string}
</rubrics>

First, analyze the response against each rubric item, discussing how well the response meets or fails each criterion. Then, provide your final score as an integer between 1 and 10, wrapped in <score> and </score> tags.

After the score, distill your analysis into transferable experiential knowledge which is general, high-level, widely applicable insights that would help improve future responses to similar tasks. Focus on reusable strategies and patterns rather than details specific to this particular response. Output this knowledge wrapped in <experience> and </experience> tags.

Example ending:
<score> your_integer_score_from_1_to_10 </score>

<experience>
some experiential knowledge...
</experience>

Your evaluation:

Figure 5: Prompt template for EL knowledge extraction. The text within the `<experience>` tags is extracted as experiential knowledge e.

The extracted experiential knowledge is then prepended as context to the teacher model using the template in Figure 6.

For RL, the LLM-as-a-Judge M evaluates each on-policy response against rubrics and produces a scalar score. We use the prompt template in Figure 7.

Here is some experiential knowledge:

<experience>
{experience}
</experience>

Given the above experiential knowledge, solve the following problem:
{prompt}

Figure 6: Prompt template for conditioning the teacher model on experiential knowledge.

You are an expert evaluator. Given a user prompt, a generated response, and a list of quality rubrics, please rate the overall quality of the response on a scale of 1 to 10 based on how well it satisfies the rubrics.
Consider all rubrics holistically when determining your score. A response that violates multiple rubrics should receive a lower score, while a response that satisfies all rubrics should receive a higher score.

<prompt>
{instruction}
</prompt>

<response>
{response}
</response>

<rubrics>
{rubric_list_string}
</rubrics>

First, analyze the response against each rubric item, discussing how well the response meets or fails each criterion. Then, provide your final score as an integer between 1 and 10, wrapped in <score> and </score> tags.
Example ending:
<score> your_integer_score_from_1_to_10 </score>

Your evaluation:

Figure 7: Prompt template for RL rubric-based scoring. The integer score extracted from the `<score>` tags is used as the scalar reward.

A.2 Ablation Prompt Templates

Below are the prompt templates used in the ablation configurations (Table 4). Each configuration has a scoring prompt (sent to M) and a context template (used to condition the teacher).

Full Critique. As shown in Figure 8. The scoring prompt is identical to RL (Figure 7). The context template prepends the entire scoring output from M.

Rubrics Only. As shown in Figure 9. The context template provides only the rubrics without any response-specific feedback.

Multiple-Choice. As shown in Figure 10. The scoring prompt extends the RL prompt by appending an instruction to select one of 9 predefined directives. If M outputs none of them, the experiential

Scoring prompt: Same as Figure 7.

Context template:

Here is an evaluation of a previous attempt on this problem:

[Evaluation by Scoring Model]
{scoring_output}

Now, given the above evaluation, please solve the problem with improvements:
{prompt}

Figure 8: *Full Critique* ablation. The scoring prompt is the same as RL. The full textual output from M (including per-rubric analysis and score) is used as context for the teacher.

Scoring prompt: None

Context template:

Here are quality rubrics that your response will be evaluated against:

<rubrics>
{rubric_list_string}
</rubrics>

Please solve the following problem, keeping the above evaluation criteria in mind:
{prompt}

Figure 9: *Rubrics Only* ablation. There is no scoring prompt. Only the rubrics are provided as context, with no response-specific feedback from M.

knowledge is left empty. So there are 10 context choices in total. The context template is the same as default EL (Figure 6), with the selected directive as the experiential knowledge.

A.3 Rubric Examples

Each training prompt is paired with a set of evaluation rubrics generated by GPT-4o. Each rubric has a title, description, and weight indicating its importance. Below are three examples spanning different task types as in Figure 11.

A.4 Experiential Knowledge Examples

Below are three examples of experiential knowledge extracted by the LLM-as-a-Coach during training. Each example corresponds to a different type of task, as shown in Figure 12.

A.5 End-to-End Example

We present a complete end-to-end example of the EL pipeline in Figure 13, showing the prompt, rubrics, on-policy response from the policy model, and the experiential knowledge extracted by the LLM-as-a-Coach.

Scoring prompt (appended after the standard scoring instructions in Figure 7):

After the score, select exactly ONE of the following 9 improvement directives that best matches the primary area where the response needs improvement. Output your selection wrapped in <experience> and </experience> tags. You must output the directive exactly as written below, with no modifications.

1. "Improve factual accuracy and correctness of claims."
2. "Follow the user's instructions and constraints more precisely."
3. "Provide more depth, detail, and thorough coverage of the topic."
4. "Improve clarity, organization, and logical flow of the response."
5. "Adopt a more appropriate tone, style, or register for the context."
6. "Be more concise and eliminate unnecessary verbosity or repetition."
7. "Enhance creativity, originality, or engagement in the response."
8. "Improve safety, sensitivity, and avoidance of harmful content."
9. "No significant improvement needed. Maintain current quality."

Example ending:
<score> your_integer_score_from_1_to_10 </score>

<experience>
SELECTED_IMPROVEMENT
</experience>
Your evaluation:

Context template: Same as Figure 6, with the selected directive as `{experience}`.

Figure 10: *Multiple-Choice* ablation. The scoring prompt appends a directive selection task. The selected directive is extracted from the `<experience>` tags and used as context with the same template as default EL. If M outputs none of required directives, the experiential knowledge is left empty. So there are 10 context choices in total.

Prompt: Given the current state of the world, what are the most pressing issues that humanity faces and what are some possible solutions or actions that could be taken to address them? Explain your reasoning and provide evidence or examples to support your claims.

Rubrics:
• **Comprehensive Issue Coverage** (weight 5): Identifies and discusses multiple pressing global issues.
• **Solution Proposals** (weight 4): Offers feasible solutions or actions for the issues discussed.
• **Evidence and Examples** (weight 4): Provides evidence or examples to support claims and solutions.
• **Reasoning Clarity** (weight 4): Explains reasoning clearly and logically for the issues and solutions presented.
• **Factual Accuracy** (weight 5): Ensures all factual statements are accurate and verifiable.
• **Empathetic Tone** (weight 2): Maintains an empathetic and understanding tone throughout the response.

Prompt: From the following tables, write a SQL query to find the director of a movie that cast a role as Sean Maguire. Return director first name, last name and movie title.

Rubrics:
• **SQL Syntax Correctness** (weight 5): The response must provide a SQL query without syntax errors.
• **Correct Table and Column Usage** (weight 5): The response must demonstrate the correct usage of table and column names as specified or implied in the prompt.
• **Query Completeness** (weight 4): The SQL query must include all necessary joins and conditions to accurately find the director of the movie with the specified role.
• **Result Clarity** (weight 4): The SQL query must clearly return the director's first name, last name, and the movie title, as requested.
• **Query Efficiency** (weight 1): The SQL query should aim to be efficient, using appropriate indexing or limiting strategies where applicable.

Prompt: We're doing a catering for 150 people. The serving portion is 2 tablespoons. I am going to give you a recipe which yields 12 portions. You will then give me the proper ratios if we're doing it for 150 people.

Rubrics:
• **Mathematical Accuracy** (weight 5): Calculates correct ingredient ratios for 150 servings based on the initial 12-portion recipe.
• **Clarity of Instructions** (weight 4): Provides clear and unambiguous steps for adjusting the recipe quantities.
• **Recipe Completeness** (weight 3): Includes all ingredients from the original recipe in the adjusted version.
• **Formatting** (weight 2): Presents the adjusted recipe in a well-organized and easy-to-read format.
• **Pitfall: Incorrect Scaling** (weight -2): Does not mention or incorrectly scales the recipe to accommodate 150 servings.

Figure 11: Three rubric examples from the training set, covering an open-ended analytical question (top), a technical coding task (middle), and a recipe scaling task with a negative-weight pitfall criterion (bottom).

Key strategies include:
1. **Maintaining Emotional Arc**: The response respects the emotional progression of the character, allowing vulnerability to slowly transition into curiosity and cautious hope. This creates a believable and engaging character experience.
2. **Respecting the Prompt Context**: The response stays closely aligned with the given dialogue and scenario, focusing on the interaction without introducing unrelated or off-track information.
3. **Clear and Accessible Language**: The language is simple and conversational, avoiding jargon or overly complex descriptions. This makes the world feel both immersive and understandable.
4. **Safe and Supportive Tone**: The response consistently uses a comforting and supportive tone, addressing distress and offering reassurance, which aligns with the emotional needs of the character.
5. **Balanced World-Building**: The response provides enough detail about the setting without overwhelming the character. It introduces the environment in a way that feels organic and grounded in the scene.
6. **Avoiding Pitfalls**: The response avoids both overwhelming information and misleading directions, ensuring that the narrative remains safe and coherent.

These strategies can be applied to future responses to maintain emotional authenticity, clarity, and narrative coherence while staying true to the character and setting.

When responding to factual questions, especially those involving specific measurements, it is crucial to prioritize **accuracy** and **clarity**. Always verify the exact figures and ensure that the response directly addresses the user's query without unnecessary elaboration.

To improve future responses, focus on **precision** in factual data and **directness** in addressing the question. Avoid assumptions or explanations that may introduce confusion. When providing additional context, ensure it is relevant and enhances understanding without diverting from the core question. Also, maintain a **clear structure** that separates factual data from explanatory content to improve readability and user satisfaction.

To create a high-quality technical report, it is essential to thoroughly understand the subject matter and structure the content in a logical, well-organized manner. The report should be comprehensive, covering all key aspects of the topic while maintaining a clear focus and depth of analysis. Ensuring that the response adheres to the specified word count is crucial for meeting user expectations and demonstrating thoroughness. Including relevant technical details and using appropriate terminology shows a deep understanding of the subject. Maintaining correct grammar and formatting enhances professionalism and readability. Finally, ensuring that the content is accurate, safe, and compliant with technical standards is essential to avoid misinformation and ensure the report is suitable for its intended audience.

Figure 12: Three examples of experiential knowledge extracted by the LLM-as-a-Coach. Each captures transferable strategies rather than instance-specific corrections.

Prompt

What is a disclaimer to put on a mental wellness app that lets others know that the relationship between the app and client is not clinical in nature and not to be a substitute for professional clinical service?

Rubrics

• **Explicit Disclaimer** (weight 5): Clearly states that the app is not a substitute for professional clinical services.
• **Non-Clinical Nature** (weight 4): Specifies that the app-client relationship is not clinical in nature.
• **Professional Advice Caution** (weight 5): Advises users to seek professional clinical advice for mental health issues.
• **Legal Liability Limit** (weight 4): Limits legal liability by clarifying the app's role and scope.
• **Clear Language** (weight 3): Uses clear and understandable language to convey the disclaimer.
• **Pitfall: Non-Specific Advice** (weight -2): Recommends specific treatments or therapies without professional input.

On-Policy Response (truncated)

Here's a professional and clear disclaimer you can include on your mental wellness app:

Disclaimer
The information and services provided by [App Name] are for general informational and educational purposes only and are not intended to replace professional mental health care. The app does not establish a clinical or therapeutic relationship with users. [App Name] is not a substitute for the advice, diagnosis, or treatment of a qualified mental health professional.

Users are encouraged to consult with a licensed mental health provider for any mental health concerns. [App Name] does not provide medical, psychological, or clinical services ...

[195 words total]

Extracted Experiential Knowledge

1. **State limitations before capabilities**: When drafting disclaimers or boundary-setting text, lead with explicit negations (e.g., "does not establish," "is not a substitute") to prevent misinterpretation of the service's scope.
2. **Pair each limitation with a constructive redirect**: Every statement of what the service does *not* provide should be followed by guidance on where to find that service (e.g., "consult with a licensed provider"), ensuring the user is not left without a path forward.
3. **Use standard legal phrasing for credibility**: Recognized formulations such as "for general informational purposes only" and "does not constitute a professional relationship" convey authority and completeness without requiring custom legal review.
4. **Keep tone neutral and accessible**: Disclaimers should be written in plain language that any user can understand, avoiding both overly casual phrasing and impenetrable legalese.
5. **Offer adaptation at the end**: When the user's exact deployment context is unclear, close with an offer to tailor the output (e.g., different platforms or tones), demonstrating helpfulness beyond the literal request.

Figure 13: End-to-end example of the EL pipeline. Given a prompt and rubrics, the policy generates a response; the LLM-as-a-Coach then extracts transferable experiential knowledge that is prepended as context to the teacher model for on-policy context distillation.