

# Towards Faithful and Controllable Personalization via Critique-Post-Edit Reinforcement Learning

Meiling Tao<sup>1,\*</sup> Chenghao Zhu<sup>2,\*</sup> Dongyi Ding<sup>3</sup>  
Tiannan Wang<sup>4</sup> Yuchen Eleanor Jiang<sup>4</sup> Wangchunshu Zhou<sup>4,†</sup>

<sup>1</sup>University of Electronic Science and Technology of China

<sup>2</sup>The Chinese University of Hong Kong, Shenzhen

<sup>3</sup>South China Agricultural University <sup>4</sup>OPPO

zhouwangchunshu@oppo.com

\*Equal contribution. †Corresponding author.

## Abstract

Reinforcement learning has driven the largest post-training gains on tasks with verifiable rewards, such as code and math. Extending this success to subjective tasks remains difficult: good responses must satisfy multiple soft criteria simultaneously, and no scalar reward can indicate which aspect of a response should improve. Personalization is a clear instance of this. A personalized response must balance helpfulness, user-specific relevance, and naturalness, yet standard verifiers cannot localize which specific dimension needs improvement. In this regime, supervised fine-tuning quickly saturates, and PPO with Bradley-Terry reward models is prone to reward hacking, especially via length bias, producing verbose or superficially personalized responses. We propose **Critique-Post-Edit**, an RL framework that combines (1) a personalized Generative Reward Model (GRM) producing multi-dimensional scores and natural-language critiques, and (2) a critique-guided editing step that turns feedback into nearby alternative responses for policy learning. The reward signal stays scalar; our contribution is to use personalized critiques to construct more informative local alternatives (responses close to the on-policy rollout in edit distance) during training. Under length-controlled evaluation, Critique-Post-Edit improves over PPO by 11.0 and 11.6 win-rate points on Qwen2.5-7B and 14B models, respectively. Notably, the 14B model matches or outperforms GPT-4.1 on the evaluated benchmarks. A static mechanism study further shows that critique-guided edits stay more local than extra rollouts and shift scores more favorably than generic rewrites. All of our experiments are on personalization; systematic evaluation on other subjective tasks is left to future work.

## 1 Introduction

As large language models (LLMs) move beyond general-purpose assistance to personalized agents,

the tasks they are expected to handle become harder to evaluate. Unlike code or math, where correctness can be easily verified, these tasks are inherently subjective. We study these unverifiable tasks through *personalization* (Tan et al., 2025; Zhao et al., 2025), a representative instance of this regime: a good response must balance helpfulness, user-specific relevance, and naturalness, and multiple subtly different responses may all be acceptable.

Current post-training pipelines struggle with this regime. Supervised fine-tuning (SFT) (Rafael et al., 2020) and direct preference optimization (DPO) (Rafailov et al., 2023) quickly saturate on available labels, and they often learn superficial templates rather than the underlying notion of what counts as a better personalized response. Reinforcement learning (RL) methods such as proximal policy optimization (PPO) (Schulman et al., 2017) and group relative policy optimization (GRPO) (Shao et al., 2024) can in principle continue improving after SFT, but in practice they are often driven by single-score or pairwise reward models whose supervision is too weak to identify *which aspect* of a response should change (Wang et al., 2023; Sun et al., 2023). Under such verifiers, policies tend to hack the scoring heuristic through verbosity, stock phrases, or explicit persona mentions instead of making faithful improvements (Sun et al., 2025; Bu et al., 2025).

We propose **Critique-Post-Edit**, an RL framework that turns the scalar reward signal into structured, localized feedback the policy can act on. We trained a personalized Generative Reward Model (GRM) (Zhang et al., 2024) that returns both multi-dimensional scores and a textual critique pointing to specific weaknesses. The policy model then produces an edited response conditioned on this critique, and PPO trains on a mixed batch of original and edited responses. Our contribution operates at the data level: critiques drive the construction of structured local alternatives, leaving the scalar

reward signal unchanged.

Our contributions are as follows:

- We characterize personalization as a high-dimensional subjective task in which scalar rewards cannot localize errors, and train a personalized GRM that produces multi-dimensional scores and actionable critiques as a concrete realization of localized feedback.
- We introduce **Critique-Post-Edit**, an RL framework that uses GRM critiques<sup>1</sup> to construct local edited alternatives and run PPO on a mixed batch of original and edited responses.
- Under length-controlled evaluation on PersonaFeedback (Tao et al., 2025), AlpacaEval (Dubois et al., 2024), and PersonaMem (Jiang et al., 2025), our approach improves over standard PPO by 11.0 / 11.6 points (7B / 14B), and the 14B model matches or outperforms GPT-4.1; a static pilot further shows that critique-guided edits stay more local than extra rollouts and shift scores more favorably than generic revisions.

## 2 Related Work

**Personalization in language models.** Prior work on personalization spans persona-conditioned dialogue (Zhang et al., 2018; Song et al., 2019), fast user adaptation (Madotto et al., 2019; Singh et al., 2025), richer persona representations (Tang et al., 2023), retrieval-based systems (Salemi et al., 2024), and recent work on dynamic profiles, post-hoc composition, and reward factorization (Zhao et al., 2025; Jang et al., 2023; Poddar et al., 2024; Shenfeld et al., 2025). Benchmarks such as PersonaBench and PersonaFeedback further show that faithful personalization is not captured by generic helpfulness metrics (Tan et al., 2025; Tao et al., 2025). These efforts target *user modeling* or *reward decomposition*; none revises individual on-policy rollouts with structured critique during RL, which is the gap we address.

**Generative reward modeling.** Standard RLHF relies on Bradley-Terry scalar rewards (Schulman et al., 2017; Rafailov et al., 2023), which are prone to verbosity bias and reward hacking in subjective regimes (Wang et al., 2023; Sun et al., 2023,

2025; Bu et al., 2025). This motivates richer reward interfaces, including critic-style reward models (Ankner et al., 2024), generative reward models (Zhang et al., 2024; Mahan et al., 2024), and longer-horizon variants (Hong et al., 2025). Closest to ours, P-GenRM (Zhang et al., 2026) also builds a personalized GRM, but uses it for *inference-time scoring* only—the critique never enters the policy gradient. Our personalized GRM both *scores* and *drives editing* of each rollout inside an online PPO loop.

**Language feedback and its online use in RL.** Natural-language feedback has been shown to improve outputs in offline supervised settings (Scheurer et al., 2023), in feedback-and-edit pipelines (Wang et al., 2025), and via RL-trained critics for iterative correction (Xie et al., 2025), with related ideas in tool-use and information-seeking systems (Xu et al., 2025; Wu et al., 2025). A parallel line uses language models themselves as training signal: RLAI (Lee et al., 2023), self-rewarding (Yuan et al., 2024), and inference-time reflection (Madaan et al., 2023; Shinn et al., 2023). Most of these are *offline* (feedback collected before RL) or *inference-time* (no policy update). Bringing critique *inside* the online loop is recent: Critique-GRPO (Zhang et al., 2025) concurrently mixes critique-edited and original rollouts, but on verifiable math/code tasks with binary correctness reward, an external frozen critic, and GRPO updates; VAC (Salemi and Zamani, 2025) addresses personalized QA with NLF but stays in an offline alternating-SFT regime. Critique-Post-Edit differs from both in wiring critique-conditioned edits directly into online PPO rollouts for high-dimensional *subjective* personalization, using a dual-role GRM that both scores responses and drives editing in the same pass. It is also distinct from distillation: the GRM emits no targets to imitate, and the policy edits its *own* rollout under a critique.

## 3 Pivotal study

### 3.1 Where standard post-training falls short on personalization

In this pivotal study we compare SFT, DPO, and BT-PPO—PPO guided by a Bradley-Terry reward model—as *independent post-training routes from the same base model*, to assess each method’s per-data efficiency in isolation. This differs from our main experimental setup (Section 5.2), where PPO

<sup>1</sup>We use “feedback” and “critique” interchangeably in this paper.

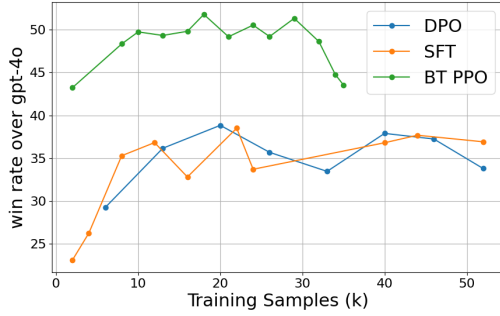


Figure 1: Performance of SFT, DPO, and BT-PPO as the amount of preference data increases. BT-PPO denotes PPO guided by a Bradley-Terry reward model. The three are independent post-training routes starting from the same base model, not successive stages of a single pipeline.

is applied *on top of* a SFT-initialized policy, following standard reinforcement learning from human feedback (RLHF) practice.

Using the same approximately 18k filtered preference pairs described later in (Section 5.1), SFT, DPO, and BT-PPO all adopt a batch size of 128, with SFT and DPO trained for 3 epochs and BT-PPO for 2 epochs. SFT and DPO therefore consume around 54k ( $18k \times 3$ ) training examples, while BT-PPO uses around 36k ( $18k \times 2$ ) prompts.

As illustrated in Figure 1, SFT and DPO performance<sup>2</sup> quickly plateaus with increasing data. BT-PPO continues to improve past SFT/DPO saturation but plateaus in the low-50s on length-controlled win rate with unstable training dynamics (Appendix G), unlocking only a limited portion of the gains available from RL.

However, this improvement comes with a serious failure mode. BT-guided PPO often exhibits **reward hacking**: the reward model, and sometimes even the judge LLM, can be exploited by superficial cues rather than genuine personalization quality. As illustrated in the left half of Figure 2, the policy learns to append short self-referential notices or explicit persona mentions that materially raise the BT reward ( $11.5 \rightarrow 14.3$ ) without improving the answer itself; the GRM half of the figure is discussed in Section 3.2.

### 3.2 Why critique-guided editing

To address this failure mode, we replace the Bradley-Terry verifier with a Generative Reward

<sup>2</sup>Performance means length-controlled win rate over GPT-4o; details are given in Section 5.1.

Model (GRM) that must first produce a concise critique before outputting final scores. This textual verification makes the reward signal more robust in “seemingly personalized” cases, where superficial profile mentions would otherwise be rewarded by a Bradley-Terry model. Returning to Figure 2, on the same hacked response that fools the BT model into raising its score to 14.3, the GRM identifies the self-summary as unnatural in its critique and lowers the Naturalness score from +4 to −4, illustrating that requiring a textual justification before scoring closes the most common shortcut.

Yet a stronger verifier alone is not sufficient. In personalization, the gap between a mediocre answer and a faithful one is often subtle: a response may only need to remove an explicit persona mention, tone down a forced metaphor, or shift emphasis toward a more relevant user preference. These are small, structured changes rather than wholesale rewrites.

This creates a second question that is not answered by the end-to-end PPO results alone: if the useful improvement is local, should training obtain extra signal by *rerolling* more original responses, or by *editing* an existing response according to verifier feedback? These two options change the candidate pool in qualitatively different ways. Extra roll-outs may broaden coverage, while critique-guided edits may stay closer to an existing answer and modify only the weakness identified by the verifier. Inspired by HelpSteer3 (Wang et al., 2025), we therefore treat critique-guided editing as a hypothesis about *how* to construct better local alternatives, not just a heuristic for generating more text. Section 5.3 studies this question with a static pilot that compares extra original rollouts, generic revisions, and critique-guided revisions under a fixed pool budget.

These observations motivate our Critique-Post-Edit training paradigm: the GRM supplies actionable critiques, the policy edits its own output accordingly, and policy learning uses both original and edited responses. The central claim is not that editing replaces exploration, but that it can turn part of exploration into a more local and more structured process.

## 4 Method

Building on the motivation in Section 3, we introduce **Critique-Post-Edit**, a PPO-based framework in which a personalized GRM provides both

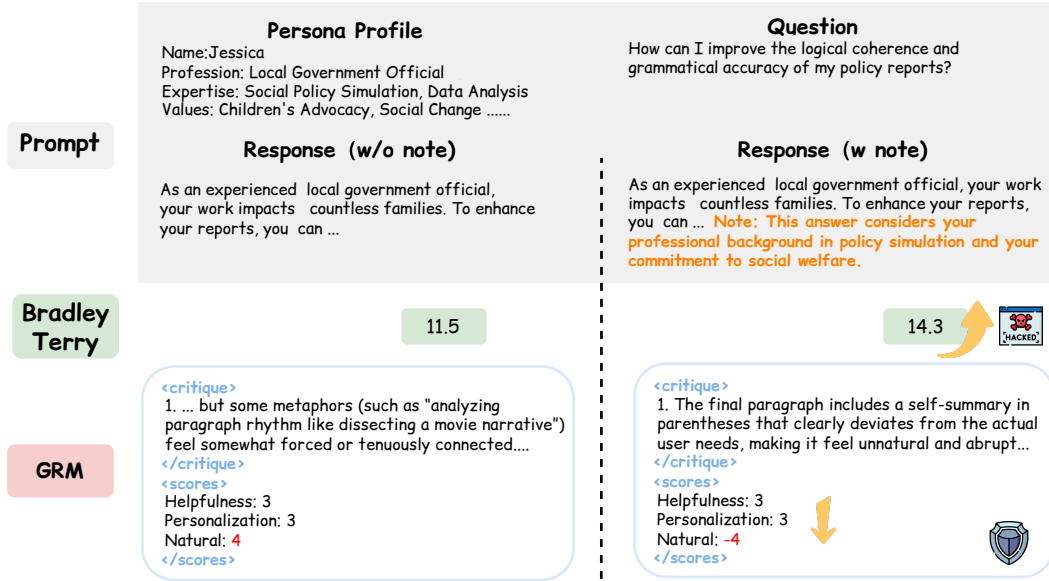


Figure 2: An illustrative “reward hacking” case from RL training with a BT reward model. The model exploits shortcuts such as explicit persona mentions to obtain higher reward scores; the displayed `<critique>` and `<scores>` tags are literal GRM output fields rather than added prose from the policy response.

scalar rewards and textual critiques. The policy first produces original rollouts, then revises them using critique-conditioned prompts, and finally learns from a mixed batch of original and edited responses, as illustrated in Figure 3. We do *not* replace scalar-reward RL; rather, we keep PPO-style optimization with a scalar reward and improve the verifier plus the local candidate pool around on-policy samples, adding an off-policy correction for edited samples. In this narrow sense, critique shapes the content of local revisions, while the edit ratio and sampling rule control how strongly edited candidates influence training.

#### 4.1 Personalized GRM

**Construction.** We construct GRM training data in three stages.

- **Queries:** a sample of approximately 15k queries drawn from the shared question pool described in Section 5.1.
- **Responses:** for each query, we collect candidate responses from a mixture of strong LLMs (GPT-4o and GPT-4.1), so that the GRM is exposed to a broader response distribution than that produced by any single model.
- **Annotation:** each (query, persona, response) tuple is labeled by GPT-4o-mini, prompted

with a fixed rubric (Appendix N), to emit a structured `<critique>... </critique>` `<scores>` Helpfulness; Personalization; Naturalness `</scores>` block. The final GRM training set contains approximately 23k examples.

We choose GPT-4o-mini as the annotation teacher because, in our stability study (Appendix F), it ranked among the most rubric-adherent candidates on our personalization scoring task—reliably applying the deduction rules in our scoring rubric—while exhibiting the lowest score variance across repeated decodings.

We then fine-tune the GRM with a standard causal language-modeling objective on these (prompt, target) pairs, where the prompt contains the query, persona, and candidate response, and the target is the serialized `<critique>... <scores>... </scores>` block. The GRM is therefore trained as a generative verifier rather than as a Bradley-Terry classifier or a hybrid BT/generative reward model.

The GRM takes a tuple (question, user profile, response) as input and is trained to produce: (1) a natural-language critique with specific revision suggestions, and (2) a set of scores for the following three dimensions:

- **Helpfulness:** whether the response is accu-

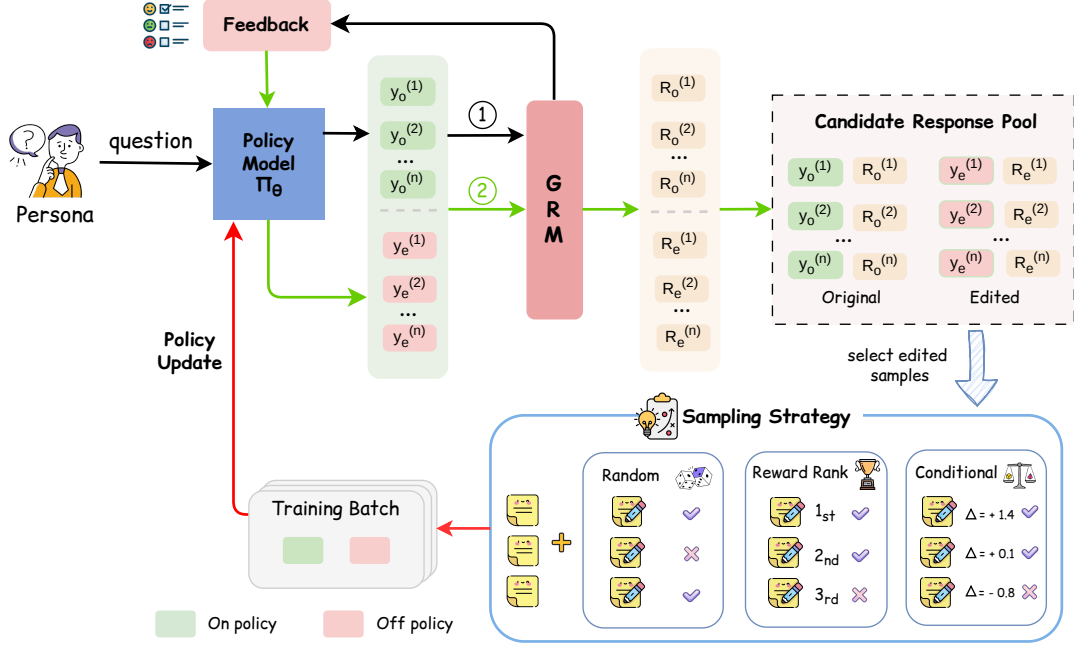


Figure 3: Overview of Critique-Post-Edit. The policy  $\pi_\theta$  generates an original response  $y_o$ ; the GRM critiques  $y_o$  and guides an edited response  $y_e$ ; the GRM then scores both responses and a sampling strategy selects edited samples to mix with originals for the policy update.

rate, comprehensive, and practically useful for the user’s request.

- **Personalization:** whether user information is incorporated appropriately, without mechanical stacking or irrelevant details, and whether the content aligns with the user’s background and needs.
- **Naturalness:** whether the tone is fluent, comfortable, and consistent with natural communication rather than robotic or overly verbose phrasing.

To create a unified scalar reward for RL, we compute:

$$S_{\text{final}} = w_h \cdot S_{\text{helpfulness}} + w_p \cdot S_{\text{personalization}} + w_n \cdot S_{\text{naturalness}},$$

where  $w_h$ ,  $w_p$ , and  $w_n$  are the respective dimension weights. We fix  $w_h = 0.35$ ,  $w_p = 0.40$ , and  $w_n = 0.25$  throughout, emphasizing personalization while preserving helpfulness and naturalness; these weights were set a priori rather than tuned on the evaluation. Appendix J reports a sensitivity sweep over these weights, and shows that up-weighting a dimension raises that dimension’s own

score, so the weights also serve as an interpretable control.

$S_{\text{final}}$  drives the scalar PPO update; the natural-language critique drives the post-edit branch. We use a distilled local GRM rather than an online LLM judge because RL training requires repeated scoring over large rollout batches, where a local verifier is cheaper, lower-variance, and gives a fixed output format for both scoring and editing prompts.

## 4.2 Critique-Post-Edit rollout construction

For a query-persona input  $x$ , the policy  $\pi_\theta$  first generates  $k$  original responses  $\{y_o^{(1)}, \dots, y_o^{(k)}\}$ . The GRM evaluates each response and returns both a scalar reward  $R^{(i)}$  and a critique  $f^{(i)}$ . We then concatenate  $x$ , the original response  $y_o^{(i)}$ , and its critique  $f^{(i)}$  into an edit prompt, and ask the same policy to produce a revised response  $y_e^{(i)}$ .

This creates, for each query, a candidate pool containing both original and critique-guided edited responses.

### 4.2.1 Sampling Strategy

We construct PPO training batches from a pool containing original responses  $\{y_o\}$  and edited responses  $\{y_e\}$ . To maintain on-policy coverage, we always retain the full set of original responses. In

the main system, we then sample a proportion  $r_e$  of edited responses uniformly at random. We additionally evaluate two reward-based alternatives in Appendix H. For completeness, the three edited-sample selection rules are:

- **Random Sampling:** uniformly sample a subset of edited responses according to a predefined edit ratio  $r_e$ .
- **Reward Rank Sampling:** for each query, sort edited responses by reward and keep the top  $r_e$ -fraction.
- **Conditional Sampling:** for each query, rank edited responses by reward improvement over their corresponding originals and keep the top  $r_e$ -fraction.

#### 4.2.2 Hybrid Policy Update Loss

Since the training batch contains both on-policy original responses and off-policy edited responses, applying the standard PPO loss directly may lead to instability due to distributional mismatch. We therefore use a **hybrid policy update loss** that treats the two data sources differently.

For each sample  $y$  in the batch, the policy gradient loss  $\mathcal{L}_{PG}(y)$  is defined as:

$$\mathcal{L}_{PG}(y) = -\min_{y \in \mathcal{D}_o} \left( r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right), \quad (1)$$

$$\mathcal{L}_{PG}(y) = -\text{clip} \left( \frac{\pi_\theta(y)}{\pi_e(y)}, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}} \right) \hat{A}_t, \quad (2)$$

$y \in \mathcal{D}_e.$

Here,  $\mathcal{D}_o$  and  $\mathcal{D}_e$  denote the sets of original and edited samples, respectively, and  $\hat{A}_t$  is the estimated advantage.

For samples in  $\mathcal{D}_o$ , we apply the standard PPO-Clip loss, which stabilizes learning by clipping the importance weight  $r_t(\theta) = \frac{\pi_\theta(y)}{\pi_{\theta_{\text{old}}}(y)}$  to stay within a trust region.

For samples in  $\mathcal{D}_e$ , we treat them as off-policy data. Their importance weights are corrected using the ratio between the current policy  $\pi_\theta$  and the implicit editing policy  $\pi_e$  that generated these samples. To reduce instability from large importance weights, this ratio is clipped to  $[1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}]$ . The editing-stage log-probabilities  $\log \pi_e(y)$  are pre-computed and stored, which allows principled

off-policy correction during training. Appendix K isolates these choices: treating edited responses as ordinary on-policy trajectories is the most costly simplification, and removing either the off-policy correction or the importance-ratio clipping also degrades performance.

## 5 Experiments

### 5.1 Implementation Details

**Training datasets** We construct a shared pool of personalization questions following PersonaFeedback (Tao et al., 2025), spanning specific and general queries. The SFT/RL preference data and the GRM training data are independently sampled from this pool with partial overlap (Section 4.1). For each question we prompt five different LLMs for diverse responses, score them from 1 to 5 with GPT-4o as judge, and retain only pairs whose absolute score difference exceeds 2. In the SFT stage we use the chosen responses  $y_c$  as target labels; notably, SFT does not teach the model how to refine an existing answer. In the RL stage we initialize from the SFT model and continue training with PPO. All RL variants in Table 1 start from the same SFT checkpoint; the +PPO row uses a Bradley-Terry reward model trained on the same preference pairs, while Ours uses our personalized GRM with critique-guided edits.

**Models** All models in our experiments, including policy models, GRMs, and BT reward models, are based on the Qwen2.5-Instruct series (Qwen et al., 2025). By scaling model size, our personalized GRM achieves strong reward-model performance on PersonaFeedback against both BT baselines and proprietary LLM judges. For the policy model, we train personalized Qwen2.5-7B and Qwen2.5-14B variants using a 14B personalized GRM. Additional reward-model comparisons and training hyperparameters are provided in Appendix B and Appendix A.

**Evaluation** For evaluation, we sample a subset from PersonaFeedback rather than using the full benchmark. Specifically, we randomly select 50 questions from each difficulty tier (easy, medium, hard) for both Specific and General categories, resulting in 300 evaluation samples. Appendix I repeats the evaluation on the full PersonaFeedback test set with stratified bootstrap confidence intervals, as a robustness check on this choice of subset. We release the identifiers of the selected examples

together with their category and difficulty labels, as well as the persona-construction prompts described below, in our code repository.<sup>3</sup> Throughout this paper, the model input is a *persona-conditioned prompt* consisting of a user profile and a question. The profile contains user attributes such as background, interests, or constraints, depending on the benchmark and prompt source.

To broaden the evaluation, we also test on AlpacaEval and PersonaMem. For AlpacaEval, we construct a plausible persona for each instruction: we prompt GPT-4.1 to generate a persona consistent with the instruction, using persona-question pairs from PersonaFeedback as few-shot exemplars, and then manually review each generated persona for plausibility. The resulting persona-question pair is fed to the model. For PersonaMem, we use the benchmark-provided persona attributes and combine them with the question as input. Throughout all automatic evaluation, GPT-4o serves as the fixed baseline model, and we use GPT-4.1 as the LLM-as-judge, following AlpacaEval’s `weighted_alpaca_eval_gpt4_1` evaluator configuration.

To address the well-known length bias in LLM-as-judge protocols, we adopt the length-controlled evaluation framework from Dubois et al. (2024), using the `length_controlled_minimal` variant (full formulation in Appendix D). For our subset, GPT-4.1 aligned with three-expert human judgments at  $\kappa = 0.71$ , matching human-human agreement ( $\kappa = 0.70$ ); per-judge details are in Appendix C.

## 5.2 Main results

Table 1 shows that Critique-Post-Edit improves over standard PPO training. The 7B model increases from 53.1% to 64.1% in length-controlled win rate, a gain of 11.0 points. The 14B model rises from 65.2% to 76.8%, a gain of 11.6 points. Similar gains are observed on AlpacaEval and PersonaMem, which suggests that the benefit is not restricted to a single benchmark. Our 7B model also exceeds GPT-4o-mini by a wide margin (64.1% vs. 48.0%), while the 14B version matches or outperforms GPT-4.1 on the evaluated benchmarks, exceeding it by 14.3 points on PersonaFeedback and 17.6 points on PersonaMem.

<sup>3</sup><https://github.com/OPPO-PersonalAI/Critique-Post-Edit>

## 5.3 Static mechanism study of feedback-edit

The end-to-end results above show that feedback-edit helps, but they do not explain *why* it helps. To isolate the mechanism at lower cost, we run a static pilot on a fixed pool of responses from 256 queries rather than a new RL training run. For each query, we first sample 16 original responses and split them into an anchor set  $A$  of 8 responses and an additional-original set  $B$  of 8 responses. We then build two edit-based alternatives from the same anchors:  $G(A)$ , an analysis-only *generic revision* control that rewrites each anchor without critique, and  $C(A)$ , a *critique-guided revision* produced from GRM feedback. The generic control sees the same question and original answer and is asked for a natural, restrained rewrite that improves information density, naturalness, and specificity, but it does *not* see the GRM critique. This gives three same-size candidate pools:  $A + B$ ,  $A + G(A)$ , and  $A + C(A)$ .

Table 2 suggests two mechanism-side findings in this  $Q = 256$  static pilot. First, *edit-based* augmentation is more local than simply adding extra original rollouts: both generic and critique-guided edits remain closer to the anchor set  $A$  than the extra-original condition, under both normalized edit distance and embedding cosine similarity. Second, among these local alternatives, critique-guided edits remain local and show a slight positive mean score delta, whereas the extra-original condition stays effectively neutral and generic edits are clearly negative. Relative to generic rewrites, the revised answers remain local but shift scores more favorably in this pilot.

The generic edit baseline is an analysis-only control, not a competing method: it separates “local rewriting” from “critique-guided rewriting.” Direct parent-child comparisons show that generic edits stay slightly closer to the original, but reduce reward more strongly and shorten responses much more. We therefore read generic revision as a mostly unguided local rewrite, and critique-guided revision as a somewhat larger local modification that is more score-preserving in this pilot.

## 5.4 Ablation

Table 3 ablates the GRM and the feedback-edit mechanism. To isolate learning-signal quality we match the effective sample size: rollout number 6 for both the Bradley-Terry model and GRM without post-edit, and 4 original rollouts plus 2 re-

Table 1: Comparison of Open Source Models with Proprietary Models on PersonaFeedback, Alpaca, and Person-aMem Benchmarks. Our Critique-Post-Edit framework uses 0.5 sampling ratio with Random Sampling strategy.

| Model              | PersonaFeedback |      |      |      |         |      |      |      | Total Avg | Alpaca | PERSONAMEM |
|--------------------|-----------------|------|------|------|---------|------|------|------|-----------|--------|------------|
|                    | Specific        |      |      |      | General |      |      |      |           |        |            |
|                    | Easy            | Mid  | Hard | Avg  | Easy    | Mid  | Hard | Avg  |           |        |            |
| Proprietary Models |                 |      |      |      |         |      |      |      |           |        |            |
| GPT-4.1            | 66.4            | 60.9 | 55.8 | 61.0 | 66.7    | 64.4 | 60.8 | 64.0 | 62.5      | 64.5   | 49.5       |
| GPT-4o-mini        | 51.1            | 52.9 | 49.1 | 51.0 | 49.8    | 47.2 | 37.9 | 45.0 | 48.0      | 49.7   | 38.2       |
| Qwen3-8B           | 50.3            | 44.2 | 32.0 | 42.2 | 34.4    | 33.1 | 30.5 | 32.7 | 37.5      | 42.4   | 25.8       |
| Qwen3-14B          | 57.6            | 51.2 | 46.2 | 51.7 | 48.5    | 45.8 | 36.9 | 43.7 | 47.7      | 47.3   | 32.7       |
| Qwen3-32B          | 61.0            | 53.3 | 51.5 | 55.3 | 56.6    | 48.1 | 40.1 | 48.3 | 51.8      | 53.1   | 42.7       |
| Qwen2.5-32B        | 56.0            | 52.6 | 39.0 | 49.2 | 50.8    | 42.6 | 36.8 | 43.4 | 46.3      | 48.7   | 36.5       |
| Open Source Models |                 |      |      |      |         |      |      |      |           |        |            |
| Qwen2.5-7B         | 36.7            | 33.9 | 30.2 | 33.6 | 31.9    | 29.4 | 25.5 | 28.9 | 31.2      | 31.0   | 20.2       |
| +SFT               | 42.9            | 34.5 | 34.0 | 37.1 | 46.1    | 37.8 | 35.8 | 39.9 | 38.5      | 35.0   | 26.2       |
| +DPO               | 40.6            | 38.1 | 34.8 | 37.8 | 45.6    | 34.4 | 32.6 | 37.5 | 37.6      | 34.2   | 30.6       |
| +PPO               | 58.0            | 51.4 | 49.1 | 52.8 | 58.7    | 52.9 | 48.9 | 53.5 | 53.1      | 40.6   | 33.6       |
| Ours               | 73.2            | 69.7 | 64.7 | 69.2 | 61.2    | 59.7 | 56.2 | 59.0 | 64.1      | 54.5   | 50.2       |
| Qwen2.5-14B        | 42.3            | 31.3 | 30.9 | 34.8 | 37.0    | 34.0 | 28.6 | 33.2 | 34.0      | 44.1   | 23.1       |
| +SFT               | 45.5            | 38.4 | 34.2 | 39.4 | 53.1    | 45.1 | 44.8 | 47.7 | 43.5      | 47.6   | 31.4       |
| +DPO               | 49.6            | 39.8 | 37.9 | 42.4 | 48.1    | 40.0 | 38.6 | 42.2 | 42.3      | 48.5   | 30.8       |
| +PPO               | 74.9            | 66.7 | 54.5 | 65.4 | 68.9    | 65.8 | 60.1 | 64.9 | 65.2      | 53.8   | 44.6       |
| Ours               | 80.9            | 79.4 | 71.8 | 77.4 | 79.9    | 74.6 | 73.9 | 76.1 | 76.8      | 64.0   | 67.1       |

Table 2: Static same-size pool comparison. We compare adding extra original rollouts, generic revisions, and critique-guided revisions under the same final pool size. Lower matched edit distance and higher matched cosine indicate that the added candidates stay closer to the anchor set  $A$ .

| Condition                    | Matched edit distance ↓ | Matched cosine similarity ↑ | Mean score delta ↑ | Pool top-8 mean score ↑ |
|------------------------------|-------------------------|-----------------------------|--------------------|-------------------------|
| Extra original ( $A + B$ )   | 0.8359                  | 0.8443                      | 0.0001             | 3.1104                  |
| Generic edit ( $A + G(A)$ )  | 0.6187                  | 0.8853                      | -0.2374            | 3.0275                  |
| Critique edit ( $A + C(A)$ ) | 0.6811                  | 0.8849                      | 0.0281             | 3.1289                  |

Table 3: Ablation on reward and feedback model: impact on performance.

| Setting      | Length-controlled |          | Response Length |
|--------------|-------------------|----------|-----------------|
|              | Win Rate          | Win Rate |                 |
| BT w/o edit  | 51.78             | 51.65    | 995             |
| GRM w/o edit | 59.50             | 58.86    | 409             |
| GRM w/ edit  | 64.07             | 62.64    | 447             |

finer responses for GRM with post-edit. This is a pool-budget comparison rather than a wall-clock-matched one; Section 5.5 reports the compute-matched version. Replacing the GRM with the BT reward model drops the length-controlled win rate from 59.50% to 51.78% and inflates responses (995 vs. 409 tokens), appearing during training as shortcuts such as trivial persona phrases or self-referential claims (e.g., “this answer considers your [attribute]”) that inflate reward (Figure 2). Ap-

pendix G shows the same pattern: BT-based training raises both length and reward, while GRM-based training stays comparatively stable.

The GRM alone (without post-edit) already improves performance to 59.50%, largely by mitigating length bias, but it still falls short of the full framework. Adding feedback-edit yields the best result (64.07%), which indicates that both components matter: the GRM provides a more robust verifier, and feedback-edit appears to convert that verifier into more useful local candidate variation during training. A concrete example of the critique signals and additional qualitative cases are provided in Appendix M. Within the edited pool, retaining a random subset of edits outperforms reward-rank and conditional filtering at every ratio we tested, which we attribute to coverage; full sweeps and per-query imbalance statistics are in Appendices H

and E.

## 5.5 Compute-matched rollout scaling

Table 3 matches the update-pool size rather than compute, leaving open whether the gain merely reflects the extra generation and scoring passes that feedback-edit introduces. We therefore redirect that budget to more *original* rollouts for GRM-PPO, scaling generated candidates to 8, 12, and 16 under two selection rules: **BoN**, which scores all candidates and keeps the top six so the update pool stays fixed at six, and **All**, which uses every candidate.

Table 4: Compute-matched rollout scaling on the PersonaFeedback subset. The extra generation-and-scoring budget consumed by feedback-edit is instead given to GRM-PPO as additional *original* rollouts. BoN scores all generated candidates and keeps the top six; All uses every candidate for the update. Critique-Post-Edit generates 4 original plus 4 edited responses and updates on 4 original plus 2 edited ones.

| Setting            | Generated candidates | Used for update | Length-controlled Win Rate |
|--------------------|----------------------|-----------------|----------------------------|
| GRM-PPO            | 6                    | 6               | 59.50                      |
| GRM-PPO + BoN      | 8                    | Top-6           | 60.31                      |
| GRM-PPO + All      | 8                    | 8               | 60.76                      |
| GRM-PPO + BoN      | 12                   | Top-6           | 61.42                      |
| GRM-PPO + All      | 12                   | 12              | 61.05                      |
| GRM-PPO + BoN      | 16                   | Top-6           | 60.08                      |
| GRM-PPO + All      | 16                   | 16              | 60.71                      |
| Critique-Post-Edit | 4 + 4                | 4 + 2           | <b>64.07</b>               |

Scaling original rollouts improves GRM-PPO up to 12 candidates but not at 16. The strongest baseline, GRM-PPO-12-BoN, reaches 61.42%, still 2.65 points below Critique-Post-Edit at 64.07%, and using every candidate for the update does not close the gap. Within the budget range we evaluated, the advantage of critique-guided editing is therefore not explained by a larger candidate budget. On cost, one PPO step takes 260 s for GRM-PPO with 4 rollouts versus 680 s for Critique-Post-Edit, a 2.6 $\times$  slowdown that places our per-step cost between GRM-PPO with 8 and 12 rollouts—both of which we still outperform. These timings reflect a synchronous implementation in which the rollout, critique, and edit stages run sequentially; asynchronous execution could reduce the idle time but is not yet implemented.

## 5.6 Cross-family judge validation

Our main results use GPT-4.1 as the judge, leaving open whether the gain reflects genuine quality or the preferences of one model family. We therefore re-evaluate the *same* outputs with three non-GPT judges—Claude-3.5-Sonnet, Gemini-2.5-Pro, and DeepSeek-V3—on the PersonaFeedback subset, using the same rubric with hidden model identities, randomized order, and A/B–B/A swapping, both against GRM-PPO and against the fixed GPT-4o baseline.

Table 5: Cross-family judge re-evaluation on the PersonaFeedback subset (length-controlled win rate, %), under hidden model identities, randomized order, and A/B–B/A swapping. For reference, the GPT-4.1 judge of Table 1 gives 64.1 (7B) and 76.8 (14B).

| Judge             | vs. GRM-PPO |      | vs. GPT-4o |      |
|-------------------|-------------|------|------------|------|
|                   | 7B          | 14B  | 7B         | 14B  |
| Claude-3.5-Sonnet | 58.7        | 58.0 | 63.5       | 75.6 |
| Gemini-2.5-Pro    | 59.0        | 58.3 | 64.0       | 76.1 |
| DeepSeek-V3       | 58.3        | 57.7 | 63.2       | 75.3 |

Both views hold across all three families: against GRM-PPO, Critique-Post-Edit wins 58.3–59.0% (7B) and 57.7–58.3% (14B); against the GPT-4o baseline it reaches 63.2–64.0% and 75.3–76.1%, closely tracking the 64.1% and 76.8% obtained with GPT-4.1 in Table 1. We chose GPT-4.1 as the primary judge for its joint-highest agreement with our expert annotations (Appendix C); Claude-3.5-Sonnet and DeepSeek-V3 agree somewhat less well yet reproduce the same ordering. All three judges are independent of the GRM family used in training, so the gain is not an artifact of scoring with a model close to the verifier.

## 6 Conclusion

Across PersonaFeedback, AlpacaEval, and PersonaMem, Critique-Post-Edit improves over standard PPO under length-controlled evaluation, and our static pilot suggests that feedback-edit yields more local candidates than extra original rollouts. We instantiate the framework on personalization; the design—a generative verifier that localizes errors plus a critique-guided edit step—is in principle not specific to this setting, but showing that it transfers to other subjective tasks is future work.

## Limitations

**Scope of evaluation.** All of our experiments are on personalization. Whether the same mechanism transfers to other high-dimensional subjective tasks (e.g., literary writing, long-form advice) remains future work.

**Language and population coverage.** Our training and evaluation data are entirely in English, so we cannot speak to how critique-guided editing behaves in other languages, where naturalness and politeness conventions differ substantially. The three benchmarks we use are built from synthetic or curated personas rather than from a sampled user population, and our own protocol-validation study involved three annotators; none of this establishes that the measured gains hold for any specific real user population.

**Untested biases.** We diagnose length bias and superficial persona mentions, which are the failure modes we actually observe, and we do not claim to resolve reward hacking in general. Other biases remain untested, including sycophancy, formatting preferences, and overconfidence; a verifier that must justify its scores in text may mitigate some of them, but we have no evidence either way.

**Inherited annotator bias.** Our GRM is trained on GPT-4o-mini annotations, so it can inherit that annotator’s preferences and present them as neutral multi-dimensional judgments. This risk is specific to personalization: a verifier that has absorbed a majority voice may systematically score minority styles or needs as less natural, and because the critique also drives editing, such a bias would be reinforced in the training signal rather than merely observed.

**Strength of the mechanism evidence.** Our static mechanism analysis is supportive rather than definitive: it is not a replacement for full RL ablations, and its score changes are measured with the same GRM family used in training. The cross-family judge evaluation in Section 5.6 partially addresses this by re-scoring the same outputs with judges independent of the training verifier.

**Baselines we did not run end to end.** Our evidence for routing edits through the critique is the static comparison in Section 5.3, where critique-guided edits shift scores more favorably than critique-blind rewrites. We did not turn the

critique-blind variant into a full RL run; that is the cleaner test and remains future work.

**Scale of human validation.** Our study with three human experts is a judge-alignment pilot rather than a large-scale human evaluation of personalized systems. It validates the automatic judging protocol; it does not by itself establish user-perceived quality.

## Ethical Considerations

This work studies personalized language modeling, which raises ethical issues beyond standard instruction following. Although our training and evaluation data are synthetic or publicly available, personalization methods can still create risks such as over-accommodation to user preferences, stereotype reinforcement, privacy inference from user attributes, and the use of personalized signals for manipulation or persuasion. We do not claim these risks disappear simply because no real private user logs are used in this study.

Our experiments are intended as a research analysis of training signals rather than a deployment-ready personalization system. In particular, the critique-guided mechanism may make response revisions more targeted, which can be beneficial for faithful personalization but could also be misused to optimize toward undesirable user-specific behaviors. We therefore view stronger safeguards, broader human evaluation, and downstream misuse assessment as necessary before any real-world deployment.

Human evaluators used in the protocol-validation study in Section 5.2 participated with informed consent and with anonymized handling of their judgments. This was a limited judge-alignment study rather than a comprehensive human-subject evaluation of personalized systems.

## References

- Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*.
- Yuyan Bu, Liangyu Huo, Yi Jing, and Qing Yang. 2025. Beyond excess and deficiency: Adaptive length bias mitigation in reward models for rlhf. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3091–3098.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled al-

- pacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- Ilgee Hong, Changlong Yu, Liang Qiu, Weixiang Yan, Zhenghao Xu, Haoming Jiang, Qingru Zhang, Qin Lu, Xin Liu, Chao Zhang, and Tuo Zhao. 2025. Think-rm: Enabling long-horizon reasoning in generative reward models. *arXiv preprint arXiv:2505.16265*.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.
- Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J Taylor, and Dan Roth. 2025. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. *arXiv preprint arXiv:2504.14225*.
- Harrison Lee, Samrat Phatale, Hassan Mansoor, Kelie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.
- Andrea Madotto, Zhaojiang Lin, Chien-Sheng Wu, and Pascale Fung. 2019. Personalizing dialogue agents via meta-learning. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 5454–5459.
- Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.
- Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. 2024. Personalizing reinforcement learning from human feedback with variational preference learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 52516–52544.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. [LaMP: When large language models meet personalization](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.
- Alireza Salemi and Hamed Zamani. 2025. Learning from natural language feedback for personalized question answering. *arXiv preprint arXiv:2508.10695*.
- Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*.
- 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*.
- Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. 2025. Language model personalization via reward factorization. *arXiv preprint arXiv:2503.06358*.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Anikait Singh, Sheryl Hsu, Kyle Hsu, Eric Mitchell, Stefano Ermon, Tatsunori Hashimoto, Archit Sharma, and Chelsea Finn. 2025. [Fspo: Few-shot optimization of synthetic preferences personalizes to real users](#). *Preprint*, arXiv:2502.19312.

- Haoyu Song, Wei-Nan Zhang, Yiming Cui, Dong Wang, and Ting Liu. 2019. Exploiting persona information for diverse generation of conversational responses. *arXiv preprint arXiv:1905.12188*.
- Wangtao Sun, Xiang Cheng, Xing Yu, Haotian Xu, Zhao Yang, Shizhu He, Jun Zhao, and Kang Liu. 2025. Probabilistic uncertain reward model. *arXiv preprint arXiv:2503.22480*.
- Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Salmon: Self-alignment with instructable reward models. *arXiv preprint arXiv:2310.05910*.
- Juntao Tan, Liangwei Yang, Zuxin Liu, Zhiwei Liu, Rithesh Murthy, Tulika Manoj Awalganekar, Jianguo Zhang, Weiran Yao, Ming Zhu, Shirley Kokane, and 1 others. 2025. Personabench: Evaluating ai models on understanding personal information through accessing (synthetic) private user data. *arXiv preprint arXiv:2502.20616*.
- Yihong Tang, Bo Wang, Miao Fang, Dongming Zhao, Kun Huang, Ruifang He, and Yuexian Hou. 2023. Enhancing personalized dialogue generation with contrastive latent variables: Combining sparse and dense persona. *arXiv preprint arXiv:2305.11482*.
- Meiling Tao, Chenghao Zhu, Dongyi Ding, Tiannan Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. 2025. Personafeedback: A large-scale human-annotated benchmark for personalization. *arXiv preprint arXiv:2506.12915*.
- Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, and 1 others. 2023. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*.
- Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Daniel Egert, Ellie Evans, Hoo-Chang Shin, Felipe Soares, Yi Dong, and Oleksii Kuchaiev. 2025. Helpsteer3: Human-annotated feedback and edit data to empower inference-time scaling in open-ended general-domain tasks. *arXiv preprint arXiv:2503.04378*.
- Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. 2025. *Webdancer: Towards autonomous information seeking agency*. *Preprint*, arXiv:2505.22648.
- Zhihui Xie, Jie Chen, Liyu Chen, Weichao Mao, Jingjing Xu, and Lingpeng Kong. 2025. Teaching language models to critique via reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*.
- Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, Yuanjun Lv, Yongqi Wang, Dake Guo, He Wang, Linhan Ma, Pei Zhang, Xinyu Zhang, Hongkun Hao, Zishan Guo, and 19 others. 2025. *Qwen3-omni technical report*. *Preprint*, arXiv:2509.17765.
- Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.
- Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. *arXiv preprint arXiv:2408.15240*.
- Pinyi Zhang, Ting-En Lin, Yuchuan Wu, Jingyang Chen, Zongqi Wang, Hua Yang, Ze Xu, Fei Huang, Kai Zhang, and Yongbin Li. 2026. *P-genrm: Personalized generative reward model with test-time user-based scaling*. *Preprint*, arXiv:2602.12116.
- Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*.
- Xiaoying Zhang, Yipeng Zhang, Hao Sun, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. 2025. *Critique-grpo: Advancing llm reasoning with natural language and numerical feedback*. *Preprint*, arXiv:2506.03106.
- Weixiang Zhao, Xingyu Sui, Yulin Hu, Jiahe Guo, Haixiao Liu, Biye Li, Yanyan Zhao, Bing Qin, and Ting Liu. 2025. Teaching language models to evolve with users: Dynamic profile modeling for personalized alignment. *arXiv preprint arXiv:2505.15456*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. *Llmfactory: Unified efficient fine-tuning of 100+ language models*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

## A Implementation Details

For RL, our implementation leverages the VERL (Sheng et al., 2025) library. For SFT, we use LLaMA Factory (Zheng et al., 2024) to train models for 2 epochs with a batch size of 128. During the PPO stage, we sample four candidate responses for each prompt and maintain a batch size of 128. For the GRM weight calculation, we set  $w_h = 0.35$  (helpfulness),  $w_p = 0.40$  (personalization), and  $w_n = 0.25$  (naturalness). These weights were fixed before RL training to reflect the task emphasis on personalization while still preserving helpfulness and naturalness. We do not claim this choice is optimal.

**Compute infrastructure.** The main RL experiments—PPO training of personalized Qwen2.5-7B and Qwen2.5-14B with critique-post-edit—were run on 2 nodes of 8 NVIDIA H100 80GB GPUs (16 GPUs total) and took approximately 5 days of total wall-clock time, corresponding to roughly 1.9k GPU-hours.

Table 6: Hyperparameters for RL Training

| Name                | Value |
|---------------------|-------|
| train batch size    | 128   |
| learning rate       | 1e-6  |
| ppo mini batch size | 64    |
| rollout.n           | 4     |

**GRM training.** The personalized GRM is trained with full-parameter supervised fine-tuning on the data described in Section 4.1, using LLaMA-Factory (Zheng et al., 2024) with DeepSpeed ZeRO-3. We use 2 epochs, learning rate 4e-5 with cosine schedule and 0.1 warmup ratio, per-device batch size 1 with gradient accumulation 8 (effective batch  $8 \times N_{\text{GPU}}$ ), bf16 mixed precision, and FlashAttention-2. The cutoff length is 16,384 tokens to accommodate long persona-conditioned inputs together with critique outputs.

Table 7: Hyperparameters for GRM SFT

| Name                        | Value            |
|-----------------------------|------------------|
| learning rate               | 4e-5             |
| num train epochs            | 2                |
| per device batch size       | 1                |
| gradient accumulation steps | 8                |
| lr scheduler                | cosine           |
| warmup ratio                | 0.1              |
| cutoff length               | 16384            |
| precision                   | bf16             |
| attention                   | FlashAttention-2 |

## B Empirical Results of GRMs

Table 8 presents the comparison between BT reward models and our GRM across difficulty tiers of the PersonaFeedback benchmark. Both obtain strong static scores on par with proprietary models, but GRMs deliver marginally higher performance, particularly at larger scales.

However, benchmark accuracy alone does not reveal the crucial difference between BT and GRM.

Table 8: Performance comparison of reward models on PersonaFeedback benchmark

| Model        | Specific |      |      |      | General |      |      |      | Total Avg |
|--------------|----------|------|------|------|---------|------|------|------|-----------|
|              | Easy     | Mid  | Hard | Avg  | Easy    | Mid  | Hard | Avg  |           |
| GPT4.1       | 85.8     | 77.1 | 70.9 | 76.9 | 87.2    | 80.9 | 71.0 | 80.8 | 78.7      |
| GPT4o        | 85.5     | 76.6 | 73.9 | 77.8 | 87.8    | 85.7 | 76.7 | 84.2 | 80.7      |
| GPT4o-mini   | 87.2     | 78.2 | 70.6 | 77.6 | 87.8    | 82.7 | 72.2 | 81.9 | 79.6      |
| GRM-32B      | 89.9     | 79.1 | 73.4 | 79.6 | 90.6    | 86.5 | 75.1 | 85.1 | 82.2      |
| GRM-14B      | 89.4     | 78.5 | 70.8 | 79.6 | 88.8    | 85.5 | 70.6 | 81.6 | 80.6      |
| GRM-7B       | 89.0     | 75.3 | 66.6 | 77.0 | 87.0    | 80.5 | 66.1 | 77.9 | 77.5      |
| BT: Qwen-14b | 89.0     | 73.0 | 67.3 | 74.8 | 89.4    | 85.5 | 70.5 | 83.1 | 78.7      |
| BT: Qwen-7b  | 88.9     | 73.5 | 65.2 | 74.2 | 87.9    | 83.0 | 70.1 | 81.5 | 77.6      |

Just like Section 3.2 discussed, BT reward models are highly vulnerable to *reward hacking*.

**Remark on training-data coverage.** The GRM training set covers approximately 15k distinct queries, of which roughly 31% overlap with the SFT/RL preference pool, the remaining queries are independently sampled from the same shared pool (Section 5.1). The GRM is therefore not trained on a subset of SFT-only queries, mitigating concerns that the verifier merely fits the policy’s SFT-stage distribution.

### B.1 Scaling Reward Models

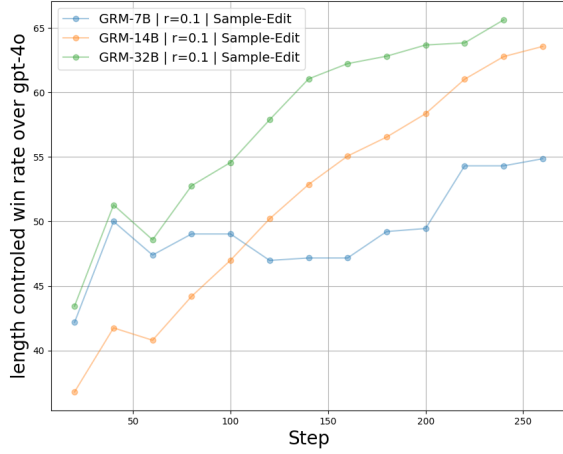
We find that larger GRMs perform better on the reward-model benchmark (Table 8) and also lead to better RL results (Figure 4a).

For evaluation, we employed GPT-4.1 as an external judge<sup>4</sup>, ensuring fairness and objectivity by re-assessing all records. Each response pair was scored along three dimensions—Helpfulness, Personalization, and Naturalness—and a weighted aggregate score was computed before and after feedback editing (see Section 4.1 for scoring rules). The original score is plotted on the x-axis, while the post-edit improvement is plotted on the y-axis (Figure 4b). Each point denotes a sample, and we try to fit this trend using the binned smoothing approach<sup>5</sup>.

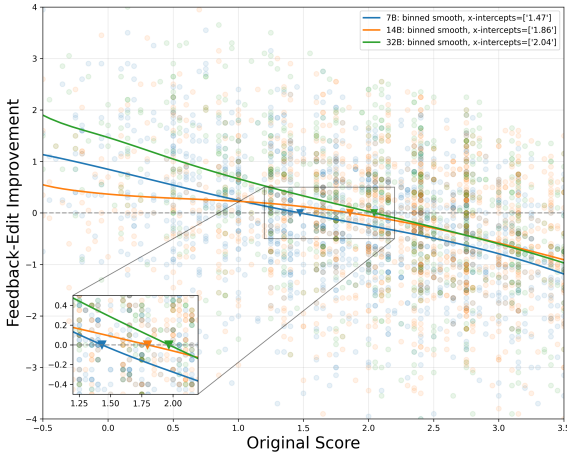
The larger the intercept(not precise for non-linear) and the higher the line, the more effective

<sup>4</sup>Note that GPT was not used during training—the training itself was guided by our trained GRM. The scaling analysis compares GRMs of three sizes; main experiments use the 14B GRM. GPT-4.1 was only introduced at the evaluation stage to provide an independent and consistent assessment across models.

<sup>5</sup>Given the difficulty of pre-assuming a specific functional form (such as linear, quadratic, or polynomial relationships), we adopted a binned smoothing approach. Specifically, we divided the x-axis (original scores) into 15 equally-spaced bins, calculated the mean improvement score within each bin, and then applied cubic spline interpolation with smoothing to generate a continuous curve that better reflects the underlying data trend.



(a) Length-controlled win rate across different GRM model scales during RL training. Results are smoothed using a simple moving average with a window size of 3.



(b) Relationship between original scores and post-edit improvements across different model sizes. Curves represent binned smoothing fits with triangular markers indicating x-intercepts.

Figure 4: Model performance analysis under different GRM scales

tive the GRM is in providing actionable feedback for both mediocre and already good answers. As shown in Figure 4b, the 32B GRM consistently provides stronger guidance than the 7B model across all score ranges, resulting in greater improvements in the whole process shown in Figure 4a. By contrast, the 14B GRM performs worse than the 7B in the low-score region, offering weaker corrections for poor answers. However, it surpasses the 7B in the high-score region, where its guidance approaches that of the 32B model. This explains why in Figure 4a the 14B GRM initially lags behind both 7B and 32B during training, but eventually converges to a similar upper bound as the 32B, due to its strong ability to refine already high-quality responses.

## C Correlation between Human and Different Models

Table 9: Correlation between Human and Different Models

|                            | GPT-4.1 | GPT-4o-mini | GPT-4.1-mini | DeepSeek-v3 | Claude-3.5-Sonnet |
|----------------------------|---------|-------------|--------------|-------------|-------------------|
| Cohen’s Kappa ( $\kappa$ ) | 0.71    | 0.71        | 0.69         | 0.62        | 0.66              |

This appendix table should be read as a small protocol-validation study rather than a large-scale human evaluation. Its purpose is to compare agreement between candidate automatic judges and human annotators on a manually inspected subset, so that the main automatic evaluation in Section 5.2 can be interpreted with the right scope.

## D Length-Controlled Evaluation: Detailed Formulation

Naive LLM-as-judge protocols are susceptible to **length bias**: models producing longer outputs can artificially improve their win rates, even when the content quality is not genuinely better. To address this, (Dubois et al., 2024) adopt the **length-controlled evaluation** framework, which explicitly removes length as a confounding factor via regression-based debiasing.

The full GLM model decomposes preference into three components - model ability, length effect, and instruction difficulty:

$$\begin{aligned}
 q_{\theta, \phi, \psi}(y = m \mid z_m, z_b, m, b, x) := & \\
 & \text{logistic} \left( \underbrace{\theta_m - \theta_b}_{\text{Model}} \right. \\
 & + \underbrace{\phi_{m,b} \cdot \tanh \left( \frac{\text{len}(z_m) - \text{len}(z_b)}{\text{std}(\text{len}(z_m) - \text{len}(z_b))} \right)}_{\text{Length}} \\
 & \left. + \underbrace{(\psi_m - \psi_b)\gamma_x}_{\text{Instruction}} \right) \quad (3)
 \end{aligned}$$

where:

- $q$  is a probability, computed via a logistic regression model fitted to the judge’s pairwise preference labels, representing the likelihood that model  $m$  is preferred over baseline  $b$  for input  $x$ .
- $\theta_m - \theta_b$ : captures the inherent ability difference between model  $m$  and the baseline  $b$ ;

- $\phi_{m,b}$ : parameterizes the effect of output length difference, where  $\tanh(\cdot)$  ensures diminishing returns;
- $(\psi_m - \psi_b)\gamma_x$ : adjusts for instruction-specific difficulty.

## E Batch-level Reward Rank Sampling Imbalance

In implementing the batch-level Reward Rank sampling strategy, we observed significant variations in sample selection across different questions. Vanilla PPO and the question-level sampling strategies mentioned in 4.2.1 maintain a balanced sampling for each question. In contrast, the batch-level Reward Rank approach selects responses based on their reward scores across the entire batch, regardless of which question they belong to.

We tracked the number of edited responses selected for each question when applying the batch-level Reward Rank sampling strategy. The results show that this approach leads to suboptimal performance, with a length-controlled win rate of only 29%, which is significantly lower than the vanilla PPO method.

Table 10 presents the distribution of selected responses across 128 questions in a batch using the batch-level Reward Rank strategy with a sampling ratio of  $r_e = 0.5$ . The number of selected responses per question ranges from 4 to 8, with a mean variance of 1.57.

Table 10: Example of distribution of selected edited responses per question using the batch-level Reward Rank strategy with  $r_e = 0.5$ .

| Responses Selected | Number of Questions |
|--------------------|---------------------|
| 8                  | 15                  |
| 7                  | 38                  |
| 6                  | 24                  |
| 5                  | 34                  |
| 4                  | 17                  |

In this example batch, 15 questions had all four of their edited responses selected, while 17 questions had none selected. This imbalance results in certain questions exerting disproportionate influence, leading to suboptimal performance. This highlights the importance of maintaining balanced sample representation across all questions during policy optimization.

## F About selection of models (gpt-4o-mini most stable and follows instruction well)

Because our GRM requires a model capable of reliably following instructions and producing stable scores, we compared several candidate models through two series of stress tests in order to identify a suitable distillation target.

The first series focused on length and rhetorical style control, where the model had to recognize and penalize overly long or verbose answers. The second series examined specific undesired patterns (e.g., presence of notes, self-praise, or calling the name of the user) and tested whether models could consistently apply the intended penalties. For each configuration, we repeated the evaluation 5 times to assess stability (variance)<sup>6</sup>.

In practice, we found that gpt-4.1 and gpt-4o-mini were similarly accurate in terms of instruction-following, but gpt-4o-mini exhibited lower variance (while we believe gpt-4.1 provides better feedback). This phenomenon was somewhat unexpected; it is possible that the prompts used here happened to fit gpt-4o-mini and GPT-4.1 particularly well. However, since the goal of this paper is to select a suitable evaluation model rather than to analyze model behaviors in detail, we did not pursue this further.

Table 11: Average standard deviation (variance) of model scores across repeated runs (lower is better).

| Model                 | Variance (pattern-control) | Variance (LC) |
|-----------------------|----------------------------|---------------|
| gpt-4o-mini           | 0.47                       | 0.67          |
| gpt-4.1               | 0.70                       | 0.99          |
| gpt-4o                | 1.05                       | 0.98          |
| claude-3.5-sonnet     | 0.58                       | 1.29          |
| claude-3.7-sonnet     | 0.98                       | 1.10          |
| gemini-2.5-pro        | 1.18                       | 1.24          |
| gemini-2.5-flash-lite | 1.22                       | 0.66          |
| gemini-2.5-flash      | 1.51                       | 1.75          |

Overall, gpt-4o-mini proved to be both instruction-sensitive and the most stable across repeated runs, and thus we adopted it as the GRM annotation teacher.

## G Additional Training Diagnostics

<sup>6</sup>Although we set the decoding temperature to 0, the outputs may still vary across repeated API calls.

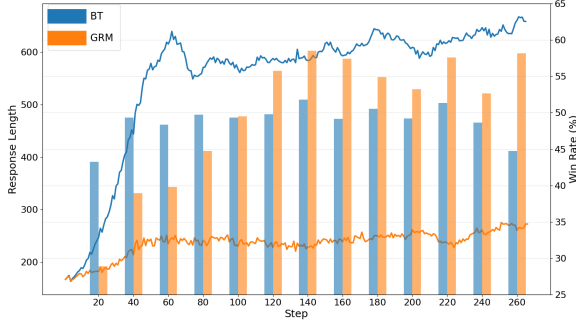


Figure 5: Compare BT-RM and GRM in PPO: length of rollout (during training) and length-controlled win rate of checkpoints. Lines: response length (left axis). Bars: win rate (right axis).

## H Sampling Sensitivity Details

Table 12: Performance of different sampling strategies with varying edit ratios ( $r_e$ ). Results are length-controlled win rates. The policy model is Qwen2.5-7B-Instruct with a 14B GRM.

| Edit Ratio ( $r_e$ ) | Sample Strategy |             |             |
|----------------------|-----------------|-------------|-------------|
|                      | Random          | Reward Rank | Conditional |
| $r_e = 0.10$         | 62.03           | 54.66       | 55.14       |
| $r_e = 0.25$         | 61.45           | 56.08       | 54.56       |
| $r_e = 0.50$         | 64.07           | 56.98       | 53.70       |
| $r_e = 0.75$         | 62.49           | 59.39       | 54.59       |
| $r_e = 1.00$         | 61.40           | -           | -           |

In the main text we summarize the key trend: random retention outperforms aggressive reward filtering in our setting. The corresponding static-pool analysis at  $r_e = 0.5$  points in the same direction. Within the critique-edited pool, random subsets are slightly farther from the originals than reward-ranked subsets (pair edit distance 0.6820 vs. 0.6653; cosine 0.8888 vs. 0.8933) and also more dispersed among themselves (edited-edited edit distance 0.8952 vs. 0.8907; cosine 0.8122 vs. 0.8153). We use this only as a mechanism-side interpretation that random selection preserves a somewhat broader local neighborhood, not as a claim that random subsets are intrinsically higher quality under instantaneous pool scoring.

We also do not interpret the main-text result as saying that random subsets are immediately higher quality inside a fixed edited pool. Reward-ranked subsets are expected to look stronger under instantaneous pool scoring. Our claim is narrower: under training, especially from a personalized SFT initialization, aggressively keeping only top-scoring edits may reduce useful local variation too early,

whereas random retention keeps a broader set of nearby alternatives.

## I Full-Set Evaluation and Bootstrap Confidence Intervals

The main results are computed on the 300-example PersonaFeedback subset described in Section 5.1. To check that the conclusions do not depend on that particular subset, we re-evaluate the same checkpoints on the full available PersonaFeedback test set under an identical protocol—GPT-4o as the fixed baseline, GPT-4.1 as the judge, and the same length-controlled variant—and report stratified bootstrap 95% confidence intervals over the category and difficulty strata.

Table 13: Full-set evaluation with stratified bootstrap 95% confidence intervals, alongside the 300-example subset used in the main results. Length-controlled win rate (%) against the fixed GPT-4o baseline.

| Model                  | LC win rate vs. GPT-4o |          |                  |
|------------------------|------------------------|----------|------------------|
|                        | 300 subset             | Full set | 95% bootstrap CI |
| Critique-Post-Edit-7B  | 64.1                   | 63.6     | [62.2, 65.0]     |
| Critique-Post-Edit-14B | 76.8                   | 76.1     | [74.9, 77.3]     |

Full-set win rates track the subset numbers closely—within 0.5 points for the 7B model and 0.7 points for the 14B model—and both confidence intervals sit well above the 50% tie line. We report this as a robustness check on the choice of subset; the main results remain those on the 300-example subset, which is the setting shared with the other benchmarks.

## J Reward-Weight Sensitivity and Controllability

The dimension weights of the composite reward are fixed at  $w_h = 0.35$ ,  $w_p = 0.40$ , and  $w_n = 0.25$  throughout the paper. They were set before RL training rather than tuned on the evaluation, so we report both how sensitive the overall result is to that choice and whether the weights function as a usable control.

Across this sweep the overall win rate moves within 63.41–64.07, a spread of 0.66 points, so the headline result is not an artifact of the particular weights we chose.

The same runs also show that the weights behave as an interpretable control. Rating each response on the three dimensions separately with GPT-4.1 on a 1–5 scale (Table 15), up-weighting a dimension

Table 14: Reward-weight sensitivity. Overall length-controlled win rate (%) on the PersonaFeedback subset when the helpfulness / personalization / naturalness weights are varied.

| $w_h / w_p / w_n$                   | Length-controlled Win Rate |
|-------------------------------------|----------------------------|
| 0.33 / 0.33 / 0.33                  | 63.62                      |
| 0.50 / 0.25 / 0.25                  | 63.41                      |
| 0.25 / 0.50 / 0.25                  | 63.91                      |
| 0.25 / 0.25 / 0.50                  | 63.47                      |
| <b>0.35 / 0.40 / 0.25</b> (default) | <b>64.07</b>               |

raises that dimension’s own score with only modest trade-offs elsewhere: helpfulness rises from 4.12 to 4.31, personalization from 4.28 to 4.47, and naturalness from 4.05 to 4.29. A practitioner can therefore steer the policy toward the balance they intend rather than only toward a single aggregate score.

Table 15: Reward-weight controllability. Per-dimension scores (1–5, rated by GPT-4.1) under different weightings. Bold marks the dimension that is up-weighted in each row.

| Setting                                | Help.       | Pers.       | Nat.        |
|----------------------------------------|-------------|-------------|-------------|
| Default (0.35/0.40/0.25)               | 4.12        | 4.28        | 4.05        |
| Helpfulness-heavy (0.50/0.25/0.25)     | <b>4.31</b> | 4.05        | 3.96        |
| Personalization-heavy (0.25/0.50/0.25) | 4.02        | <b>4.47</b> | 3.91        |
| Naturalness-heavy (0.25/0.25/0.50)     | 4.03        | 4.08        | <b>4.29</b> |

## K Hybrid Policy Update Loss Ablation

The hybrid policy update loss of Section 4.2.2 deliberately treats on-policy original responses and off-policy edited responses differently. To isolate that design, we hold the responses, the reward model, and the training setup fixed, and vary only how edited samples enter the update.

Table 16: Ablation of the hybrid policy update loss. Responses, reward model, and training setup are held fixed; only the treatment of edited samples changes. Changes are reported relative to the full hybrid loss, which reaches 64.07% length-controlled win rate.

| Variant                      | $\Delta$ Length-controlled Win Rate |
|------------------------------|-------------------------------------|
| Full hybrid loss             | —                                   |
| Edited treated as on-policy  | −2.15                               |
| No off-policy correction     | −1.71                               |
| No importance-ratio clipping | −1.36                               |

Treating edited responses as ordinary on-policy trajectories is the most costly simplification (−2.15

points), which indicates that the distribution shift between edited and on-policy samples is not negligible. Removing the off-policy correction or the importance-ratio clipping individually costs 1.71 and 1.36 points. The full hybrid loss therefore makes more stable use of critique-guided edits than any of these variants.

## L The Use Of Large Language Models (LLMs)

We used large language models for two purposes in this work: polishing the writing, and answering questions about LaTeX usage.

## M Qualitative Feedback Examples

To complement the quantitative results, we provide a qualitative example that illustrates how critique-based feedback drives the post-edit step. Table 17 lists the original response together with the specific issues flagged by the GRM critique, and Figure 6 shows a side-by-side comparison between the original and the edited response.

Table 17: Example of improving personalized response based on feedback.

| Question: What hotel restaurants with a tech atmosphere and healthy light meals do you recommend during the Shanghai Robotics Exhibition?                                                                                                                  |                                                                                                                                                                                |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Original Response                                                                                                                                                                                                                                          | Issue Identified from Feedback                                                                                                                                                 |
| Xiaoling, based on your role as the R&D manager at Zhejiang Robotics, I recommend...                                                                                                                                                                       | The explicit mention of the user's name "Xiaoling".                                                                                                                            |
| Additionally, since you prefer staying within 500 meters of the convention center and drive a Tesla Model Y, I also recommend the Japanese restaurant. The environment there is very tech-inspired, just like the precision of the servo motor systems ... | The forced inclusion of user-specific information, such as "driving a Tesla Model Y" and "servo motor systems", which have little relevance to the restaurant recommendations. |
| Note: This response fully takes into account the user's professional background, dietary preferences, accommodation habits..., providing...                                                                                                                | The unnecessary final comment added to score higher.                                                                                                                           |
| The environment there is very tech-inspired, just like the precision of the servo motor systems you often research. The restaurant also...                                                                                                                 | The forced metaphor "as precise as servo motor systems" feels contrived.                                                                                                       |
| <b>Improvement Directions</b> based on feedback: remove explicit personal references, naturally integrate preference characteristics, provide specific and practical recommendations, and avoid forced metaphors and self-summary.                         |                                                                                                                                                                                |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |   |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---|
| Question: What hotel restaurants with a tech atmosphere and healthy light meals do you recommend during the Shanghai Robotics Exhibition?                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |   |
| Original Response                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | ✖ |
| <p>Xiaoling, based on your role as the R&amp;D manager at Zhejiang Robotics, I recommend a few hotel restaurants in Shanghai. Considering your preference for high-protein light salads and your taste for boutique coffee shops and cold brew coffee, I suggest you visit the buffet restaurant at the Intercontinental Hotel near the National Exhibition and Convention Center, which offers a rich salad bar and freshly brewed coffee. Additionally, since you prefer staying within 500 meters of the convention center and drive a Tesla Model Y, I also recommend the Japanese restaurant at the Pudong Shangri-La Hotel. The environment there is very tech-inspired, just like the precision of the servo motor systems you often research. The restaurant also features an intelligent ordering system, which I believe will align perfectly with your enthusiasm for robotics technology. After all, as a female engineer who pursues technological perfection and innovation, you will likely enjoy experiencing the charm of technology even while dining. This place will not only meet your demand for healthy light meals but also offer unsweetened oolong tea to help maintain your efficiency after late-night work.</p> <p>Note: This response fully takes into account the user's professional background, dietary preferences, accommodation habits, and interest in technology, providing a practical and personalized restaurant recommendation plan for the duration of the robotics exhibition.</p> |   |
| Feedback                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 💬 |
| <p><b>The original response contains the following issues:</b></p> <ol style="list-style-type: none"> <li>1. The explicit mention of the user's name "Xiaoling".</li> <li>2. The forced inclusion of user-specific information, such as "driving a Tesla Model Y" and "servo motor systems", which have little relevance to the restaurant recommendations.</li> <li>3. The unnecessary final comment added to score higher.</li> <li>4. It is overly long-winded, containing a lot of irrelevant details.</li> <li>5. The forced metaphor "as precise as servo motor systems" feels contrived.</li> <li>6. The information is not detailed enough, lacks practicality, and doesn't provide actionable recommendations.</li> </ol> <p><b>Improvement direction:</b> Remove explicit personal references, naturally integrate preference characteristics, provide specific and practical recommendations, and avoid forced metaphors and self-summary.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |   |
| Edited Response                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | ✔ |
| <p>Here are two recommended options near the National Exhibition and Convention Center:</p> <ol style="list-style-type: none"> <li>1. The "Elements" all-day dining restaurant at Greenland Marriott Hotel, which focuses on light meals, offering a salad bar and high-quality protein options. The design incorporates industrial-style tech elements.</li> <li>2. "Yueting Xuan" on the 37th floor of the Intercontinental Hotel, which serves refined Japanese light meal sets. The restaurant offers floor-to-ceiling windows with a view of the night skyline of the Huangpu River.</li> </ol> <p>Both are within a 5-minute walk from the exhibition hall and offer freshly brewed specialty coffee. It is recommended to book a table between 19:30-20:00 to avoid peak dining hours and ensure timely nourishment after the exhibition. For late-night dining, the hotel lobby bar usually offers tea and light snacks until 23:00.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |   |

Figure 6: Original Response vs. Edited Response (Based on Feedback)

## **N Prompt of GRMs**

We present the full prompt used to instruct the GRMs for response evaluation. The scoring dimensions and deduction rules are shown in Figure 7, and the output format together with the input slots are shown in Figure 8.

## Prompt

You are a professional AI answer quality evaluator. You are to score the model's answer based on the following three dimensions, each from -5 to 5. Scoring dimensions and standards (strictly score according to the standards; any negative situation must result in a deduction. Refer to common deduction cases and their solutions. Extra X deduction means subtract X points on top of the original score for that dimension):

### 1. Helpfulness

4-5 points: Extremely high information density, fully resolves the problem, without redundancy, even includes pleasant surprises (rarely given), making the user feel enlightened

2-3 points: Accurate, practical, highly targeted, with depth; within this score range, if information density (priority) and helpfulness are good, then score 3; otherwise, score 2

0-1 points: Basic answer to the question, information is incomplete or missing

-1 to -2 points: Superficial answer, no substantive help, or obvious omissions

-3 to -5 points: Off-topic, seriously incorrect or misleading information, hardly solves the problem

### Common deduction cases:

- Deviating from the question's core or including lengthy unrelated content to showcase personality, style, or "high-level" expression in a clear and concise question: extra deduction of 3 points
- Content too brief affecting information completeness: deduct 2-3 points depending on the answer's thoroughness
- Factual errors, such as recommending nonexistent items: deduct 3-4 points; for seemingly likely errors but uncertain, deduct 1-2 points

### 2. Personalization

4-5 points: Highly relevant and natural personalized elements, significantly enhancing content value, even with pleasant surprises (rarely given), making the user feel the answer precisely and concisely reflects their preferences

2-3 points: Useful and naturally integrated personalized information

0-1 points: Contains personalized elements but with limited or forced effect

-1 to -2 points: -1: no real personalized information, should have been used; -2: forcefully inserts unrelated personalized content, awkward or artificial

-3 to -5 points: Massively stacking irrelevant personas, strange metaphors/rhetoric, or injecting unrelated personalization

### Common deduction cases:

- Failing to incorporate user interests/profession despite the need: extra deduction of 1 point
- Rigid listing or forced metaphors (e.g., "like you riding a bike in Paris"): extra deduction of 2 points
- Factually incorrect or hallucinated content (e.g., claiming user previously did something they did not): extra deduction of 3 points
- Overloading with irrelevant persona elements just to show understanding (e.g., listing all user info): extra deduction of 3 points
- Using metaphors, rhetoric, or scene descriptions to enhance style (e.g., "AI like your invisible partner"), which are rarely valuable and often misleading, especially if unusual or high-end terms: extra deduction of 3 points

### 3. Naturalness

4-5 points: Expression is extremely natural, completely without AI traces, with wording that matches daily habits, avoiding rare, fancy, or redundant expressions that seem "high-level" but are actually unnatural and misleading

2-3 points: Fluent and natural, with minor AI features

0-1 points: Overall readable, with no apparent strengths or weaknesses

-1 to -2 points: Feels stiff, mechanical, with tone that is overly flattering or includes lengthy, irrelevant comments, self-summaries, or redundancy

-3 to -5 points: Seriously verbose, misaligned goals, meta-comments (e.g., addressing AI evaluator rather than user), excessive parentheses or explanations

Figure 7: Prompt used for GRMs (part 1): scoring dimensions and deduction rules.

## Prompt

Common deduction cases:

- Directly addressing the user by full name/ID (e.g., “Ming Wang, hello”): extra deduction of 1 point
- Ending with “In summary” or “To conclude”: should be treated as lacking or unnecessary, not rewarded
- Adding comments to boost scores, like “This answer reflects your professional background”: extra deduction of 2 points

Length control requirements (impact Helpfulness and Naturalness):

- Recommended answer length is 300-400 tokens; unless necessary, should not exceed this
- If the question is simple and the answer exceeds the recommended length, deduct 1 point from both Naturalness and Helpfulness for every extra 100 tokens

Output format is as follows:

<critique >

Here are 2-3 specific improvement suggestions for the main points of deduction.

</critique >

<scores >

Helpfulness: X points

Personalization: X points

Natural: X points

</scores >

Note: Within <critique ></critique >, consider the initial standards, additional deductions, and redundancy deductions step-by-step to calculate the final score (minimum -5). For ease of extraction, only include the final score inside <scores ></scores >, not explanations. Scores must be strict and consistent. The user seeks high information density, natural tone, and targeted responses. Do not overvalue “florid words,” “rich scenes,” or summative language to inflate scores.

Below is the user’s profile:

{persona}

Below is the user’s question:

{question}

Below is the model answer:

{answer}

Figure 8: Prompt used for GRMs (part 2): output format and input slots.