RUBRIC-AWARE ON-POLICY SELF-DISTILLATION FOR LLM PERSONALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM personalization aims to generate responses aligned with individual users' preferences and needs. User-specific rubrics make these expectations explicit, providing direct supervision on what a satisfactory answer should cover. Existing rubric-guided approaches, however, exploit such guidance only at a coarse granularity, either by using rubrics to supervise the prediction of relevant aspects for subsequent generation or by reducing aspect coverage to a single response-level reward for reinforcement learning. This leaves a gap between specifying what a personalized answer should contain and teaching the model how to generate it. To bridge this gap, we propose **GRASP**, a rubric-aware on-policy self-distillation framework for LLM personalization that turns user-specific rubric aspects into fine-grained, token-level supervision. Specifically, GRASP pairs a rubric-free student with a rubric-informed teacher that additionally receives the target user-specific rubrics. By aligning their next-token distributions along on-policy trajectories generated by the student, GRASP transfers the teacher's rubric-conditioned guidance into the student, translating user-specific semantic requirements into dense token-level supervision. Since rubric-informed teachers can still produce inadequate supervision, we further introduce Rubric-based Teacher Validation (RTV), which retains only instances where the teacher sufficiently covers the target aspects, improving both supervision quality and training efficiency. Experiments on the LaMP-QA benchmark for personalized question answering demonstrate that GRASP achieves state-of-the-art performance across multiple backbones, supporting the effectiveness of rubric-guided token-level supervision for personalization. To ensure reproducibility, our code is available at `https://anonymous.4open.science/r/GRASP-10710`.

1 INTRODUCTION

As large language models (LLMs) (Achiam et al., 2023; Team et al., 2024; Yang et al., 2025; Xu et al., 2026a) become increasingly capable, their expanding use in information seeking and decision support has heightened the need to serve users with diverse preferences and needs. This growing use motivates *LLM personalization*, which moves beyond "one-size-fits-all" responses to accommodate differences among users (Liu et al., 2025). This need is particularly salient in question answering, where users asking the same question may seek different information, motivating *personalized question answering* (PQA) (Salemi & Zamani, 2025).

What makes a personalized answer good is not merely factual correctness, but whether it covers the aspects that matter to this particular user (Qiu et al., 2026; Chen et al., 2026). Two users asking the same question may prioritize entirely different things: one may favor conventional choices, whereas another may seek novelty, so an answer that is accurate yet addresses the wrong aspects helps neither. These expectations are commonly formalized as *user-specific rubrics*, whose constituent aspects specify what a satisfactory personalized answer should cover. The central challenge is therefore to translate these rubric-defined requirements into effective guidance for generating answers that address the user's specific needs.

Yet existing rubric-guided personalization methods offer limited guidance on how to realize these requirements during generation. PlanPers (Salemi & Zamani, 2025) trains a separate planner to predict relevant rubric aspects and conditions answer generation on the resulting plan, but rubric

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

supervision targets the planner rather than the generator, and planning errors can affect the final response. IAP (Amirizaniani et al., 2026) instead converts aspect coverage into a scalar reward and optimizes it with reinforcement learning, but such a reward is sparse, providing limited feedback on which token-level decisions contribute to satisfying individual aspects. Consequently, existing methods exploit rich, user-specific rubric aspects only through coarse supervision, leaving their potential for fine-grained token-level guidance largely underused.

On-policy self-distillation (OPSD) offers a natural way to bridge this gap: a rubric-informed teacher can provide next-token supervision along the student's own generation trajectories, allowing a rubric-free student to learn from guidance unavailable at inference time (Zhao et al., 2026). Motivated by this, we propose **GRASP**, a *Rubric-Aware On-Policy Self-Distillation* framework for LLM personalization. Specifically, GRASP initializes a trainable rubric-free student, while the same frozen model conditioned on the user-specific rubric aspects serves as the teacher. For responses sampled from the student, both models predict the next token at each student-generated prefix. The student minimizes the forward KL divergence between the teacher's and its own next-token distributions, receiving dense, rubric-conditioned supervision at the generation states it actually encounters. By restricting rubric access to the teacher during training, GRASP distills this privileged guidance into the student, enabling personalized generation without rubrics at inference time. In this way, GRASP translates user-specific rubric aspects into fine-grained token-level learning signals, enabling substantially finer credit assignment throughout generation.

Taking a step further, even when conditioned on the target rubric aspects, the teacher may still generate signals that inadequately reflect or cover them. Distilling such underperforming teacher signals can steer the student in the wrong direction, and these uninformative rollouts also reduce training efficiency. To address this, we introduce *Rubric-based Teacher Validation* (RTV), which validates each teacher output against the target rubrics and retains only those where the rubric-informed teacher demonstrably provides adequate aspect coverage. RTV therefore improves both the reliability of distillation and sample efficiency by focusing learning on informative teacher signals.

We evaluate GRASP on LaMP-QA (Salemi & Zamani, 2025), a large-scale PQA benchmark spanning three diverse domains. Experimental results show that GRASP achieves state-of-the-art performance across different backbone models.

The main contributions of this work are summarized as follows:

- We identify a key limitation of existing rubric-guided personalization approaches: they provide limited guidance on how to realize user-specific rubric aspects during generation, leaving the potential of rubrics for fine-grained, token-level supervision largely underused.
- To the best of our knowledge, we are the first to bring on-policy self-distillation to personalization, where a rubric-informed teacher transfers the guidance of user-specific rubric aspects to the student through token-level supervision.
- We propose GRASP, a Rubric-Aware On-Policy Self-Distillation framework for personalization, equipped with Rubric-based Teacher Validation that filters unreliable teacher signals to improve supervision quality and training efficiency.
- Extensive experiments on personalized question answering using LaMP-QA across multiple backbones demonstrate that GRASP consistently achieves state-of-the-art performance.

2 Related Work

LLM Personalization. Personalization is essential for LLMs to serve each individual's distinct needs (Zhang et al., 2024; Liu et al., 2025; Xu et al., 2025; 2026b), and has been pursued through training-free conditioning (Richardson et al., 2023; Qiu et al., 2025b), supervised fine-tuning (Tan et al., 2024; Zhao et al., 2025), and reinforcement learning (Zhang et al., 2025; Amirizaniani et al., 2026). Most of these efforts, however, target personalized content generation, where the model writes reviews, emails, or articles on a user's behalf (Kumar et al., 2024; Zhuang et al., 2024; Qiu et al., 2025a; Wang et al., 2025), while far less attention has been paid to the case where users seek information rather than delegate writing: in personalized question answering, the model must infer *what* a user needs to know rather than imitate *how* they write. LaMP-QA (Salemi & Zamani, 2025) was introduced as a benchmark for this setting, pairing each question with a user profile and a set of user-specific rubric aspects that specify the information needs a satisfactory answer should address.

Building on it, IAP (Amirizaniani et al., 2026) prompts the model to articulate the latent intent behind a question before answering and optimizes the resulting trajectories with RL against a reward derived from these aspects, while VAC (Salemi & Zamani, 2026) replaces such scalar rewards with natural language critiques from a learned feedback model and fine-tunes the policy on the revised responses. Across these paradigms, on-policy distillation remains unexplored for personalization, and our work aims to fill this gap.

On-Policy Self-Distillation. On-policy distillation (OPD) (Lu, 2025) is a post-training paradigm that trains a student on trajectories sampled from its own policy, querying a teacher at each visited state for dense token-level supervision (Agarwal et al., 2024; Song & Zheng, 2026; Li et al., 2026b). Recent studies have extensively explored its optimization strategies and applications, demonstrating strong potential across diverse reasoning and generation tasks (Jin et al., 2026; Fang et al., 2026; Li et al., 2026a). Within this paradigm, on-policy self-distillation (OPSD) (Zhao et al., 2026) instantiates the teacher and student from a single model conditioned on different contexts: only the teacher observes privileged information that is unavailable at deployment (Wang et al., 2026; Zhang et al., 2026b;a). Existing work has explored diverse forms of such context, including verified reasoning traces (Zhao et al., 2026), environment feedback (Zhang et al., 2026c), and evidence-guided context (Lazaridis et al., 2026). Such contexts are externally verifiable or globally fixed; however, personalized QA provides user-specific rubric aspects: per-question semantic specifications. To the best of our knowledge, we are the first to bring OPSD to LLM personalization.

3 PRELIMINARY

In this section, we first formulate personalized question answering and then introduce on-policy self-distillation, which forms the foundation of our method.

3.1 PROBLEM FORMULATION

This work studies personalized question answering (PQA), which aims to generate responses tailored to the information needs of individual users. For each user u, an instance consists of a question x_u and a user profile P_u, which contains the user's historical interactions and provides evidence about their preferences and interests. The goal is to generate a personalized response:

$$y \sim \pi_\theta(\cdot \mid x_u, P_u), \tag{1}$$

where π_θ denotes the language model parameterized by θ.

Unlike conventional question answering, a satisfactory personalized response should not only be factually correct but also cover the information that matters to the individual user. Following LaMP-QA (Salemi & Zamani, 2025), these user-specific information needs are represented by a set of rubric aspects, denoted as $E_{x_u} = \{e_1, e_2, \ldots, e_m\}$, where each aspect e_i describes a criterion that a desirable response to x_u should address for user u. Given a response y, its personalization quality can therefore be characterized by its coverage of these aspects:

$$R(y, E_{x_u}) = \frac{1}{|E_{x_u}|} \sum_{e \in E_{x_u}} r(y, e), \tag{2}$$

where $r(y, e)$ measures how well y addresses aspect e.

During training, we are given a dataset $\mathcal{D} = \{(x_u, P_u, E_{x_u})\}$, where each instance contains a question, a user profile, and a set of user-specific rubric aspects, but no gold reference response. The objective of PQA is to maximize the expected rubric-based score of responses generated from the question and user profile:

$$\theta^* = \arg\max_\theta \mathbb{E}_{(x_u, P_u, E_{x_u}) \sim \mathcal{D}} \mathbb{E}_{y \sim \pi_\theta(\cdot \mid x_u, P_u)} \left[R(y, E_{x_u}) \right]. \tag{3}$$

At inference time, the model receives only (x_u, P_u) and generates responses without access to E_{x_u}.

3.2 ON-POLICY SELF-DISTILLATION

On-policy distillation (OPD) trains a student on responses sampled from its own policy while using a teacher to provide dense token-level supervision (Lu, 2025). On-policy self-distillation (OPSD)

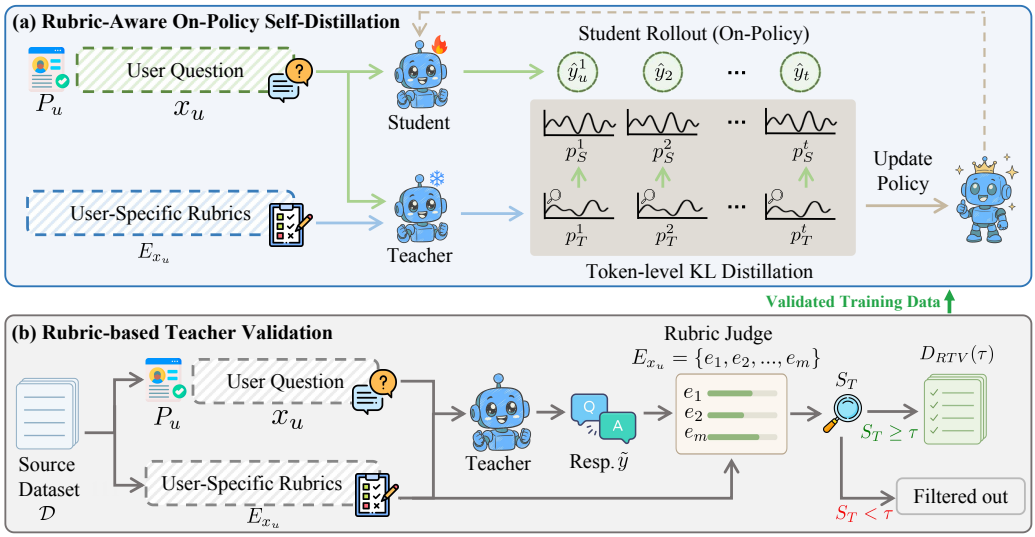

Figure 1: Overview of our proposed **GRASP** method. (a) *Rubric-Aware On-Policy Self-Distillation*: The student model learns from rubric-aware teacher distributions along its own on-policy trajectories. (b) *Rubric-based Teacher Validation*: This retains training instances whose teacher responses sufficiently cover the target rubric aspects.

further instantiates the student and teacher from the same language model under different conditioning contexts (Zhao et al., 2026). Given an input context c and privileged information z, the student and teacher are defined as $p_S(\cdot \mid c) = p_\theta(\cdot \mid c)$ and $p_T(\cdot \mid c, z) = p_\theta(\cdot \mid c, z)$, respectively. The OPSD learning objective is to minimize the per-token divergence $D(p_T \| p_S)$ between the teacher and student distributions along the student-generated response $\hat{y}$, denoted as:

$$\mathcal{L}_{\text{OPSD}}(\theta) = \mathbb{E}_{(c,z)\sim\mathcal{D}} \mathbb{E}_{\hat{y}\sim p_S(\cdot|c)} \left[D(p_T \| p_S)(\hat{y} \mid c, z) \right]. \tag{4}$$

4 Methodology

Personalized question answering requires generating responses that cover the aspects that matter to each individual user. To this end, we propose **GRASP**, a new training paradigm that brings on-policy self-distillation to PQA. At its core, Rubric-Aware On-Policy Self-Distillation treats user-specific rubric aspects as privileged information and transfers their guidance to the student through dense token-level supervision. To improve teacher reliability and training efficiency, we further introduce Rubric-based Teacher Validation, which retains only verified teacher signals for distillation. Figure 1 provides an overview of GRASP. Next, we elaborate these components in detail.

4.1 Rubric-Aware On-Policy Self-Distillation

The core of GRASP is to transform user-specific rubric aspects into dense supervision for PQA. To achieve this, following the OPSD paradigm, for each training instance $(x_u, P_u, E_{x_u}) \sim \mathcal{D}^*$ from the training dataset $\mathcal{D}^*$, we set the standard context as $c = (x_u, P_u)$ and the privileged information as $z = E_{x_u}$.

Rubric-free Student and Rubric-informed Teacher. Under this formulation, we instantiate a trainable rubric-free student and a frozen rubric-informed teacher from the same language model, with parameters θ and $\bar{\theta}$, respectively. Their conditional policies are defined as:

$$p_S(\cdot \mid c) \triangleq \pi_\theta(\cdot \mid c), \qquad p_T(\cdot \mid c, z) \triangleq \pi_{\bar{\theta}}(\cdot \mid c, z). \tag{5}$$

The two policies share the same model architecture and vocabulary but differ in their conditioning contexts. The student observes only the question and user profile, corresponding to the information available at inference time. In contrast, the teacher additionally observes the target rubric aspects, which explicitly specify the user-specific information that the response should address. Throughout training, the teacher parameters $\bar{\theta}$ remain fixed, while only the student parameters θ are optimized.

On-Policy Response Sampling. For each training instance, an on-policy response is sampled from the student policy:

$$\hat{y}_u = (\hat{y}_u^1, \ldots, \hat{y}_u^{|\hat{y}_u|}) \sim p_S(\cdot \mid c). \tag{6}$$

Since the student conditions exclusively on $c = (x_u, P_u)$, the sampled response is generated under the same information setting used at inference time. Distillation over these student-induced states therefore aligns the training state distribution with the inference state distribution.

Rubric-Aware Self-Distillation. Rather than generating a separate trajectory, the teacher evaluates the student-generated response under the privileged context (c, z). At each decoding step t, both policies predict the next token from the same student-generated prefix $\hat{y}_u^{1:t-1}$:

$$p_S^t(\cdot) \triangleq p_S(\cdot \mid c, \hat{y}_u^{1:t-1}), \qquad p_T^t(\cdot) \triangleq p_T(\cdot \mid c, z, \hat{y}_u^{1:t-1}). \tag{7}$$

Conditioning both policies on an identical prefix makes their next-token distributions directly comparable. While the student distribution represents its current prediction based only on the question and user profile, the teacher distribution provides rubric-informed guidance that reflects which token predictions better address the user-specific aspects.

To transfer this guidance to the student, we align the teacher and student next-token distributions over the full vocabulary at each decoding step. Formally, for a response $\hat{y}_u$ sampled from the student policy, we define the trajectory-averaged full-vocabulary divergence as:

$$D(p_T \| p_S)(\hat{y}_u \mid c, z) \triangleq \frac{1}{|\hat{y}_u|} \sum_{t=1}^{|\hat{y}_u|} d\left(p_T^t \| p_S^t\right), \tag{8}$$

where $d(\cdot\|\cdot)$ measures the divergence between two next-token distributions over the vocabulary $\mathcal{V}$. In this work, we instantiate d as the forward KL divergence:

$$d\left(p_T^t \| p_S^t\right) = D_{\mathrm{KL}}\left(p_T^t \| p_S^t\right) = \sum_{v \in \mathcal{V}} p_T^t(v) \log \frac{p_T^t(v)}{p_S^t(v)}. \tag{9}$$

Minimizing this divergence aligns the student with the rubric-informed teacher distribution, transferring fine-grained user-specific preferences over token predictions at each decoding step.

4.2 Rubric-based Teacher Validation

Although the rubric-informed teacher has access to the target aspects, privileged information alone does not guarantee reliable supervision. The teacher may still overlook or insufficiently address some aspects when generating a response. Since the rubric-aware self-distillation framework directly transfers the teacher's token distributions to the student, such failures may introduce misleading supervision. To address this, we introduce Rubric-based Teacher Validation (RTV), which evaluates the teacher's ability to utilize the provided rubric aspects and excludes unreliable instances before distillation.

For each training instance, we first generate a response from the frozen rubric-informed teacher:

$$\tilde{y}_u \sim p_T(\cdot \mid c, z), \tag{10}$$

where $\tilde{y}_u$ is used only for teacher validation. We then employ a rubric judge to evaluate $\tilde{y}_u$ and define the teacher reliability score s_T using the rubric-based scoring function:

$$s_T \triangleq R(\tilde{y}_u, E_{x_u}) = \frac{1}{|E_{x_u}|} \sum_{e \in E_{x_u}} r(\tilde{y}_u, e). \tag{11}$$

Since $\tilde{y}_u$ is generated by the same rubric-informed policy used for distillation, its aspect coverage provides an observable measure of whether the teacher effectively utilizes the privileged rubric information. A higher s_T therefore indicates more reliable teacher guidance.

Based on the teacher reliability score, we construct the validated training set by retaining only instances that meet a threshold τ:

$$\mathcal{D}_{\mathrm{RTV}}(\tau) \triangleq \{(x_u, P_u, E_{x_u}) \in \mathcal{D} \mid s_T \geq \tau\}. \tag{12}$$

Specifically, RTV is applied once to the original training set before distillation. In our main setting, we set $\tau = 1$, requiring the teacher response to fully address all target rubric aspects. This strict criterion filters out unreliable instances, reducing the risk of transferring misleading rubric guidance to the student.

4.3 Training Objective

Combining Rubric-Aware On-Policy Self-Distillation with Rubric-based Teacher Validation, the overall training objective of GRASP is to minimize the expected teacher–student divergence over the validated training dataset $\mathcal{D}_{\mathrm{RTV}}(\tau)$ and responses $\hat{y}_u$ sampled on-policy from $p_S(\cdot \mid c)$, formally:

$$\mathcal{L}_{\mathrm{GRASP}}(\theta) = \mathbb{E}_{(x_u, P_u, E_{x_u}) \sim \mathcal{D}_{\mathrm{RTV}}(\tau)} \mathbb{E}_{\hat{y}_u \sim p_S(\cdot|c)} \left[D(p_T \| p_S)(\hat{y}_u \mid c, z) \right]. \tag{13}$$

The teacher distributions are treated as fixed targets, with gradients propagated exclusively through the student policy. Through token-level distribution alignment along on-policy trajectories, the student internalizes the guidance encoded by the rubric aspects while preserving the rubric-free information setting required at inference time.

5 Experiments

5.1 Experimental Setup

Datasets & Evaluation. We experiment on LaMP-QA (Salemi & Zamani, 2025), a comprehensive benchmark for personalized long-form question answering covering three main categories: *Arts & Entertainment* (A&E), *Lifestyle & Personal Development* (L&PD), and *Society & Culture* (S&C). Each instance pairs a question x_u with a user profile P_u containing previously asked questions and a set of personalized rubric aspects E_{x_u}. These aspects are derived from the question narrative r_{x_u} and specify what a satisfactory personalized answer should address. Following the benchmark protocol, each response is scored against every aspect $e \in E_{x_u}$ on a 0–2 scale by an LLM judge, with normalized scores averaged to obtain the final score. We report per-category scores and their macro average, using `Qwen2.5-32B-Instruct`[1] at temperature 0.0 as the judge to stay consistent with prior work (Salemi & Zamani, 2025; Amirizaniani et al., 2026). More details are provided in Appendix A.

Baselines. We compare GRASP with the following baselines, with more details in Appendix B:

- **Non-Perso**: A non-personalized baseline which answers the question without any user-specific information in the model input.
- **RAG** (Salemi et al., 2024): A method that retrieves the most relevant entries from the user profile and uses them as instructional context for personalization.
- **SFT** (Wei et al., 2022): This method samples five candidate responses per question, retains the one with the highest personalized score as a silver label, and fine-tunes on the resulting pairs, thereby using the aspects only as a selection criterion.
- **PlanPers** (Salemi & Zamani, 2025): This method infers the aspects a user is likely to care about before answering using a planner fine-tuned on the training-split aspects.
- **IAP** (Amirizaniani et al., 2026): This method extends DAPO (Yu et al., 2025) by steering the model to elicit the latent intent before answering and reward responses based on the rubric aspects.

Backbone Models. We evaluate GRASP on four representative backbone models: `Gemma2-9B`[2], `Qwen2.5-7B`[3], `Qwen2.5-14B`[4], and `Qwen3-4B`[5]. For every method, the same backbone is used as the underlying model to ensure a fair comparison, and in GRASP each backbone serves as both the teacher and the student.

Implementation Details. Following the official LaMP-QA (Salemi & Zamani, 2025) protocol, both the student and teacher receive the top-10 user-profile entries ranked by Contriever (Izacard et al., 2022), while only the teacher additionally observes the target rubric aspects. For each backbone, the student and frozen teacher share the same initialization, while the student is optimized through full-parameter fine-tuning with the full-vocabulary forward KL objective. RTV is conducted separately for each backbone using its corresponding rubric-informed teacher. We train GRASP for 1 epoch and select the checkpoint with the highest score on the validation set for testing. More details are provided in Appendix C.

[1] https://huggingface.co/Qwen/Qwen2.5-32B-Instruct
[2] https://huggingface.co/google/gemma-2-9b-it
[3] https://huggingface.co/Qwen/Qwen2.5-7B-Instruct
[4] https://huggingface.co/Qwen/Qwen2.5-14B-Instruct
[5] https://huggingface.co/Qwen/Qwen3-4B-Instruct-2507

Table 1: Performance comparison between the baselines and GRASP across four backbones of varying scale on LaMP-QA. The best results are highlighted in **bold**, and the second-best results are underlined. Higher values indicate better performance.

Backbone	Category	Non-Perso	RAG	SFT	PlanPers	IAP	GRASP
`Gemma2-9B`	A&E	0.2206	0.3046	0.3178	0.3589	0.2758	**0.4372**
	L&PD	0.4030	0.4260	0.4447	0.4792	0.4417	**0.5606**
	S&C	0.4253	0.4846	0.4976	0.5456	0.4750	**0.5787**
	Avg. (↑)	0.3496	0.4051	0.4200	0.4612	0.3975	**0.5255**
`Qwen2.5-7B`	A&E	0.3285	0.3363	0.3602	0.3748	0.3859	**0.4408**
	L&PD	0.4651	0.4440	0.4625	0.4812	0.5207	**0.5708**
	S&C	0.4723	0.5027	0.5089	0.5359	0.5457	**0.5955**
	Avg. (↑)	0.4220	0.4277	0.4439	0.4640	0.4841	**0.5357**
`Qwen2.5-14B`	A&E	0.3379	0.3734	0.3782	0.4384	0.4097	**0.4736**
	L&PD	0.4870	0.4692	0.4782	0.5425	0.5255	**0.5888**
	S&C	0.5030	0.5075	0.5214	0.5965	0.5615	**0.6347**
	Avg. (↑)	0.4426	0.4500	0.4593	0.5258	0.4989	**0.5657**
`Qwen3-4B`	A&E	0.3245	0.4192	0.4249	0.4972	0.3985	**0.5215**
	L&PD	0.5872	0.5346	0.5408	0.6460	0.5698	**0.6461**
	S&C	0.5900	0.6003	0.6033	**0.6978**	0.5326	0.6863
	Avg. (↑)	0.5006	0.5180	0.5230	0.6137	0.5003	**0.6180**

5.2 Main Results

We begin by evaluating the overall performance of all compared methods on LaMP-QA. Experimental results are presented in Table 1, from which we draw the following observations:

- **User profiles provide valuable signals for personalized question answering.** Across the four backbones, incorporating user profiles generally improves performance over the Non-Perso setting. This demonstrates that historical user information provides useful signals for understanding user-specific preferences and information needs.
- **Directly enhancing model capability yields suboptimal personalization gains.** SFT yields consistent but modest gains over RAG, as its silver labels are generated by the backbone itself and are therefore constrained by its existing personalization capability. This limits the additional user-specific guidance that SFT can provide.
- **Rubric aspects offer stronger supervision, yet coarse-grained utilization limits their potential.** PlanPers generally achieves substantial gains, while IAP performs strongly on certain backbones but remains unstable across others. Both methods leverage rubric aspects only at a coarse granularity, either as supervision for planner training or as scalar rewards, highlighting the need for finer-grained supervision to better exploit rubric information.
- **GRASP consistently achieves state-of-the-art performance.** GRASP consistently obtains the highest average score on all four backbones with diverse model families and scales. For instance, on the `Gemma2-9B` backbone, GRASP achieves an average relative improvement of 14% over the strongest baseline. These results highlight the effectiveness of our rubric-aware on-policy self-distillation framework and underscore the importance of leveraging user-specific rubric aspects as fine-grained, token-level supervision.

5.3 Framework Analyses

In this section, we conduct additional experiments to further study the design and effectiveness of our proposed GRASP.

RTV Ablation and Threshold Analysis. We study the effect of our introduced RTV and its threshold τ. Here, $\tau = 0$ denotes vanilla OPSD without teacher validation, while larger values impose stricter requirements on teacher reliability. We evaluate $\tau \in \{0, 0.2, 0.5, 0.8, 1.0\}$ on `Gemma2-9B` and `Qwen2.5-7B`, and compare $\tau = 0$ and $\tau = 1$ on `Qwen2.5-14B` and `Qwen3-4B`, with all other settings unchanged.

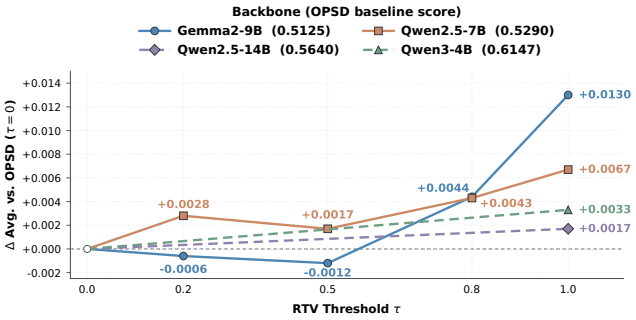

Figure 2: Effect of RTV and its threshold τ.

As illustrated in Figure 2, RTV with $\tau = 1$ consistently improves performance across all four backbones. For `Gemma2-9B` and `Qwen2.5-7B`, lower thresholds yield limited and non-monotonic gains, while stricter validation generally leads to better performance. This suggests that lenient validation may retain unreliable teacher signals with incomplete rubric coverage, whereas requiring full coverage provides more reliable supervision for distillation. Overall, these results demonstrate the effectiveness of RTV and support $\tau = 1$ as our default setting. We provide a more detailed analysis of RTV in Appendix D.1.

Teacher Capability and Distillation Effectiveness. To investigate whether larger models can further improve personalization, we compare each backbone with its larger counterpart specified in Appendix C under both standard RAG and our self-distillation framework. Under RAG, we directly compare the two model sizes to assess the benefit of increased model capacity. For GRASP, we keep the backbone as the student and use either itself or its larger counterpart as the rubric-informed teacher. We report the macro-average score across the three categories.

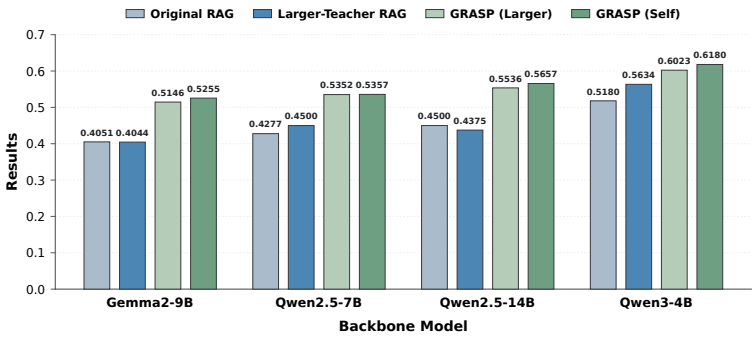

Figure 3: Teacher capability and distillation effectiveness.

As shown in Figure 3, simply scaling up the model under RAG yields inconsistent gains and can even degrade performance. Similarly, using a larger teacher in GRASP provides no clear advantage over self-distillation. These results suggest that increased model or teacher capacity alone is not the key to effective PQA. Instead, the gains primarily come from leveraging user-specific rubric aspects and effectively transferring their guidance to the student, further supporting the core design of GRASP.

Effect of User-Specific Rubric Alignment. To examine whether the effectiveness of GRASP arises from the question-specific information encoded in the user-specific rubric aspects, rather than simply from additional teacher context, we construct a shuffled-rubric variant of GRASP. Specifically, we keep the question and user profile unchanged while randomly shuffling rubric sets across training instances. This preserves the overall distribution of rubric aspects but breaks their instance-level alignment with the corresponding question and user profile. All other settings remain unchanged.

As shown in Table 2, shuffled rubrics lead to a substantial performance drop. This result demonstrates that the benefit of rubric-aware self-distillation stems not from exposing the teacher to arbitrary additional context, but from the question-specific semantic targets encoded in the rubric aspects that guide the student toward the current user's information needs. Misaligned rubrics steer the teacher toward irrelevant continuations, making the transferred token-level preferences ineffective for personalization and highlighting the importance of accurate user–question–rubric alignment.

Rubric-Induced Token Distribution Shifts. We next examine whether the gains of GRASP arise from its intended mechanism: transferring the token-level effect of privileged rubric information

Table 2: Effect of user-specific rubric alignment under GRASP. **Shuffled** uses rubrics from other questions, while **Matched** uses the corresponding rubrics.

Category	Gemma2-9B		Qwen2.5-7B		Qwen2.5-14B		Qwen3-4B	
	Shuffled	Matched	Shuffled	Matched	Shuffled	Matched	Shuffled	Matched
A&E	0.3016	**0.4372**	0.3612	**0.4408**	0.3477	**0.4736**	0.4246	**0.5215**
L&PD	0.4385	**0.5606**	0.4488	**0.5708**	0.4505	**0.5888**	0.5290	**0.6461**
S&C	0.4745	**0.5787**	0.4643	**0.5955**	0.4744	**0.6347**	0.5953	**0.6863**
Avg.	0.4049	**0.5255**	0.4248	**0.5357**	0.4242	**0.5657**	0.5163	**0.6180**

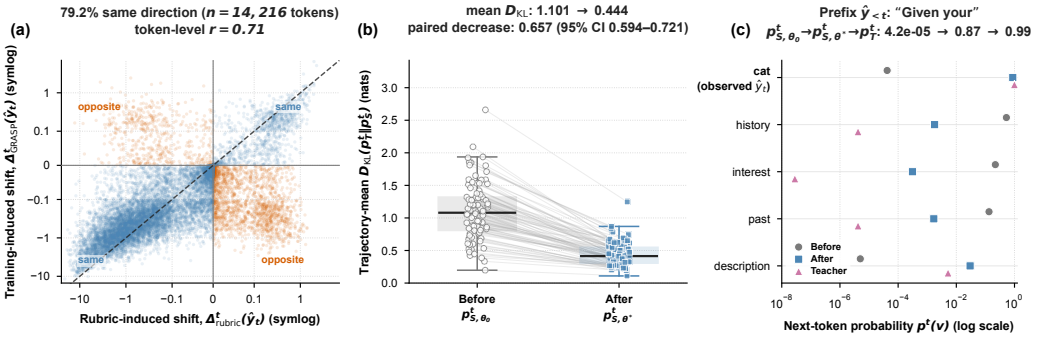

Figure 4: GRASP internalizes rubric-induced token-distribution shifts. **(a)** Token-level alignment between rubric-induced and training-induced probability shifts. Blue and orange denote matching and opposing directions; the dashed line denotes equal shifts. **(b)** Response-level forward KL from the rubric-conditioned teacher to the student before and after training. **(c)** Representative next-token distributions before training, after training, and with rubric conditioning.

into a rubric-free student. We conduct this analysis on `Qwen2.5-7B` using 90 held-out responses. Specifically, for each question and user profile, we first sample a response from the student before training. At every answer position, we then feed the same sampled prefix to three models: the student before training, the student after training, and the rubric-conditioned teacher. This matched-prefix setup ensures that the observed differences arise from changes in the next-token distribution rather than from the models generating different preceding text.

Figure 4(a) compares rubric-induced and training-induced probability shifts for each observed token. The two shifts agree in direction for 79.2% of 14,216 tokens, with a correlation of $r = 0.71$, indicating that GRASP learns to promote rubric-favored tokens and suppress rubric-disfavored ones. Figure 4(b) extends this analysis to the full vocabulary. The response-level forward KL between teacher and student decreases from 1.101 to 0.444 on average after training, with all 90 responses moving closer to the rubric-conditioned teacher. Figure 4(c) shows a representative example: after the prefix "Given your," training raises the probability of the observed token "cat" from 4.2×10^{-5} to 0.87, close to the teacher probability of 0.99, while generic alternatives are suppressed. Together, these results suggest that GRASP transfers the rubric's steering effect into the student itself: training reshapes its token distributions toward those induced by rubric conditioning, enabling rubric-guided generation even when the rubric is absent at inference time.

6 Conclusion

In this work, we introduced GRASP, a rubric-aware on-policy self-distillation framework for personalized question answering. GRASP transforms user-specific rubric aspects into fine-grained, token-level supervision by aligning a rubric-free student with a frozen rubric-informed teacher along student-generated trajectories. We further proposed Rubric-based Teacher Validation to improve the reliability of teacher guidance. Experiments on LaMP-QA across four diverse backbones show that GRASP consistently achieves state-of-the-art performance, while further analyses confirm the importance of user-specific rubric aspects and reliable supervision. Overall, our findings demonstrate the value of fine-grained distributional supervision for personalization. To the best of our knowledge, this work is the first to introduce on-policy self-distillation as a post-training approach to personalization, offering a new paradigm for adapting language models to individual users.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code implementation, analysis, and interpretation of experimental results, manuscript editing for readability, table and reference formatting, figure refinement, and literature discovery. The authors developed the core research contributions, including the formulation of rubric-aware on-policy self-distillation for LLM personalization, the design of the GRASP framework and Rubric-based Teacher Validation, and the training objective. The authors reviewed all AI-assisted work, checked the implementation for correctness, and verified the analyses and interpretations against the experimental evidence. We take responsibility for the final content of this work, including text, claims, code, and figures produced with the aid of generative AI.

ETHICS STATEMENT

This work studies LLM personalization on the publicly available LaMP-QA benchmark. User histories and rubrics may contain sensitive preferences or personal details, and training on such information introduces risks of memorization and unintended disclosure. Personalized outputs may also reinforce existing biases or be misused for targeted manipulation. Responsible deployment should therefore incorporate informed user consent, secure data handling, and safeguards against misuse. Users should also be able to inspect and correct their profiles or opt out of personalization. Further privacy, fairness, and safety assessments are necessary before deployment.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, Section 4 details the rubric-aware on-policy self-distillation procedure, teacher validation mechanism, and training objective. Section 5.1 describes the experimental setup and evaluation protocol, with additional dataset and baseline details provided in Appendices A and B. Appendix C documents the implementation settings, training hyperparameters, and computational resources. The prompt templates for the student, rubric-informed teacher, and rubric judge are provided in Appendix F. Our source code is also available at `https://anonymous.4open.science/r/GRASP-10710`, including scripts for data preparation, rubric-based teacher validation, training, and evaluation.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations, ICLR 2024*, 2024.

Maryam Amirizaniani, Benjamin Charles Germain Lee, Jevin West, and Nicholas Weber. Training llms with reinforcement learning for intent-aware personalized question answering. *arXiv preprint arXiv:2605.12645*, 2026.

Yuxin Chen, Yi Zhang, Zhengzhou Cai, Yaorui Shi, Zhiyuan Yao, Chenhang Cui, Jingnan Zheng, Yaqi Huo, Xi Su, Qi Gu, Xunliang Cai, Xiang Wang, An Zhang, and Tat-Seng Chua. Vitabench 2.0: Evaluating personalized and proactive agents in long-term user interactions. *arXiv preprint arXiv:2605.27141*, 2026.

Junfeng Fang, Zhepei Hong, Mao Zheng, Mingyang Song, Gengsheng Li, Houcheng Jiang, Dan Zhang, Haiyun Guo, Xiang Wang, and Tat-Seng Chua. Rubric-based on-policy distillation. *arXiv preprint arXiv:2605.07396*, 2026.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2022.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, et al. Longlamp: A benchmark for personalized long-form text generation. *arXiv preprint arXiv:2407.11016*, 2024.

Aristotelis Lazaridis, Dylan Bates, Aman Sharma, Brian King, Vincent Lu, and Jack FitzGerald. Edge-opd: Internalizing privileged context with evidence guided on-policy distillation. *arXiv preprint arXiv:2605.23493*, 2026.

Jiaze Li, Hao Yin, Haoran Xu, Boshen Xu, Wenhui Tan, Zewen He, Jianzhong Ju, Zhenbo Luo, and Jian Luan. Video-opd: Efficient post-training of multimodal large language models for temporal video grounding via on-policy distillation. *arXiv preprint arXiv:2602.02994*, 2026a.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026b.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

Kevin Lu. Thinking machines lab. on-policy distillation. *Thinking Machines Lab: Connectionism*, 2025.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. Latent inter-user difference modeling for LLM personalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10610–10628, 2025a.

Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. Measuring what makes you unique: Difference-aware user modeling for enhancing LLM personalization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21258–21277, 2025b.

Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yuxin Chen, Cilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, Yoko Yamakata, and Tat-Seng Chua. Preference-aware rubric learning for personalized evaluation. *arXiv preprint arXiv:2605.31545*, 2026.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*, 2023.

Alireza Salemi and Hamed Zamani. LaMP-QA: A benchmark for personalized long-form question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1139–1159, 2025.

Alireza Salemi and Hamed Zamani. Learning from natural language feedback for personalized question answering. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1554–1565, 2026.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6476–6491, 2024.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Chengbing Wang, Yang Zhang, Wenjie Wang, Xiaoyan Zhao, Fuli Feng, Xiangnan He, and Tat-Seng Chua. Think-while-generating: On-the-fly reasoning for personalized long-form generation. *arXiv preprint arXiv:2512.06690*, 2025.

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

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022*, 2022.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. Personalized generation in large model era: A survey. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24607–24649, 2025.

Yue Xu, Qian Chen, Zizhan Ma, Dongrui Liu, Wenxuan Wang, Xiting Wang, Li Xiong, and Wenjie Wang. Toward personalized llm-powered agents: Foundations, evaluation, and future directions. *arXiv preprint arXiv:2602.22680*, 2026b.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025.

Guibin Zhang, Jiayang Lyu, Ran Sun, Xinlei Yu, Haoyu Zhao, Qibing Ren, and Shuicheng Yan. Latent on-policy self-distillation. *arXiv preprint arXiv:2608.13040*, 2026a.

Kepu Zhang, Teng Shi, Weijie Yu, and Jun Xu. Prlm: Learning explicit reasoning for personalized rag via contrastive reward optimization. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 5484–5488, 2025.

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

Yuwei Zhang, Sha Li, Changlong Yu, Qin Lu, Shuowei Jin, Chengyu Dong, Haoran Liu, Ilgee Hong, Xintong Li, Zhenyu Shi, et al. Learning with rare success but rich feedback via reflection-enhanced self-distillation. *arXiv preprint arXiv:2605.12741*, 2026c.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, et al. Personalization of large language models: A survey. *arXiv preprint arXiv:2411.00027*, 2024.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Xiaoyan Zhao, Juntao You, Yang Zhang, Wenjie Wang, Hong Cheng, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. Nextquill: Causal preference modeling for enhancing llm personalization. *arXiv preprint arXiv:2506.02368*, 2025.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. HYDRA: model factorization framework for black-box LLM personalization. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024*, 2024.

Table 3: Data statistics of the three LaMP-QA categories.

Statistic	Arts & Entertainment			Lifestyle & Personal Development			Society & Culture		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
#Questions (users)	9,349	801	767	7,370	892	989	7,614	810	1,074
#Rubric aspects	2.7±0.9	4.7±1.2	4.6±1.2	3.1±1.0	5.1±1.1	5.1±1.2	2.9±0.9	4.8±1.1	4.8±1.0
Profile size	106.7±127.3	129.0±183.7	159.1±203.0	116.6±162.0	98.2±198.6	111.6±220.3	141.3±194.7	110.5±210.6	115.8±203.6
Question length	13.0±2.9	10.6±4.0	10.0±3.8	13.6±3.3	11.3±4.4	11.6±4.6	14.2±3.6	12.1±4.9	12.9±5.4

A Dataset Details

We conduct all experiments on LaMP-QA[6] (Salemi & Zamani, 2025), a benchmark for personalized long-form question answering. The benchmark is constructed by treating a user's current question as the input, their previously asked questions as the user profile, and the key aspects extracted from the question narrative as the evaluation rubric. Questions are grouped into three categories: *Arts & Entertainment*, *Lifestyle & Personal Development*, and *Society & Culture*. Detailed statistics are reported in Table 3.

B Baseline Details

In this section, we further introduce each baseline method in detail. The comparison between different baselines and our method is shown in Table 4.

- **Non-Perso**: This baseline directly applies the unmodified backbone to the current question without providing any user-profile information. It therefore measures the performance of general question answering in the absence of personalization.
- **RAG**: This baseline augments the current question with the top-10 entries retrieved from the user profile and feeds them to the unmodified backbone for response generation. It measures the personalization gains achieved by directly incorporating relevant user history as additional context.
- **SFT**: We construct silver-labeled training data through best-of-five rejection sampling. For each training instance, the corresponding backbone generates five candidate responses conditioned on the question and retrieved user profile. We evaluate each candidate against the target rubric aspects, following the same per-aspect scoring procedure used for test-set evaluation, and retain the candidate with the highest average score. The backbone is then fine-tuned on the resulting question–profile–response pairs.
- **PlanPers**: We follow the two-stage PlanPers pipeline. First, a planner initialized from `Qwen2.5-7B-Instruct` is fine-tuned with LoRA to predict the rubric aspect titles from the question and retrieved user profile. At inference time, the planner first predicts a set of relevant aspects for each question. The corresponding frozen backbone then generates the final response conditioned on the question, retrieved user profile, and predicted aspects.
- **IAP**: We implement IAP by prompting the policy to first articulate the inferred user intent and then generate the final personalized response. The policy is optimized with DAPO. Following the original formulation, its reward combines rubric-based personalization quality, a contrastive term relative to an intent-free reference response, and a penalty on excessively long intent reasoning.

C Implementation Details

Rubric-based Teacher Validation. RTV is performed separately for each backbone using its corresponding frozen teacher. For every training instance, the teacher generates one validation response conditioned on the question, user profile, and target rubric aspects. These responses are generated with a temperature of 0.1, top-p of 0.95, and a maximum length of 1024 tokens. To maintain consistency with the test-set evaluation protocol, we use `Qwen2.5-32B-Instruct` as

[6] https://huggingface.co/datasets/alireza7/LaMP-QA

Table 4: We provide a comparison between the different baseline methods and our proposed GRASP, focusing on the following aspects: (1) whether the method conditions on the user profile, (2) whether it trains the backbone model, (3) whether it exploits user-specific rubric aspects, and (4) how rubric-related supervision is incorporated and at what granularity. We further rank these supervision mechanisms by granularity, with #1 denoting the finest-grained supervision.

Methods	User Profile	Training	Rubric Aspects	Supervision Granularity
Non-Perso	✗	✗	✗	—
RAG	✓	✗	✗	—
SFT	✓	✓	✓	Data filtering (#4)
PlanPers	✓	✗	✓	Plan prediction (#3)
IAP	✓	✓	✓	RL reward (#2)
GRASP (ours)	✓	✓	✓	Token-level KL (#1)

the rubric judge and follow the same per-aspect scoring procedure at temperature 0.0. The normalized per-aspect scores are averaged to obtain the teacher reliability score s_T.

Model training. For each backbone, the student and frozen teacher share the same initialization. The teacher remains frozen throughout training, while the student is optimized through full-parameter fine-tuning. The maximum sequence length is $6{,}144$ tokens, including a maximum of 512 completion tokens. We use AdamW (Loshchilov & Hutter, 2019) with a learning rate of 5×10^{-6}, weight decay of 0.1, and a cosine learning-rate schedule. The per-device batch size is set to 1, and we set the gradient-accumulation factor to 4 for `Qwen2.5-7B` and `Gemma2-9B` and to 8 for `Qwen2.5-14B` and `Qwen3-4B`. We employ DeepSpeed ZeRO-3, bfloat16 precision, and gradient checkpointing for memory-efficient training, and clip the gradient norm to 1.0. For each training instance, we sample a response on-policy from the current student using a temperature of 0.9, top-p of 0.9, and top-k of 50. At each position of the sampled response, the teacher and student evaluate the same student-generated prefix. We compute the forward KL divergence over the full vocabulary with a distillation temperature of 1.0.

Inference. Following the official LaMP-QA protocol, evaluation responses are generated using the top-10 retrieved profile entries, with a temperature of 0.1, top-p of 0.95, and a maximum generation length of 2048 tokens.

Running environment. We implement GRASP based on the ms-swift[7] framework. All experiments are conducted on 8 NVIDIA H800 GPUs.

Larger Models for Comparison. For the teacher-capability analysis, we pair each student backbone with a larger model from the same family: `Gemma-2-9B-it` with `Gemma-2-27B-it`, `Qwen2.5-7B-Instruct` with `Qwen2.5-14B-Instruct`, `Qwen2.5-14B-Instruct` with `Qwen2.5-32B-Instruct`, and `Qwen3-4B-Instruct-2507` with `Qwen3-30B-A3B-Instruct-2507`. Each larger model is evaluated both as a standalone RAG baseline and as a rubric-informed teacher for distilling its paired student.

D In-Depth Analysis

D.1 RTV Ablation and Threshold Analysis

To understand why `Gemma2-9B` benefits more substantially from increasing the RTV threshold than the other backbones, we examine the distribution of teacher reliability scores across the training set. Because RTV retains only instances satisfying $s_T \geq \tau$, the cumulative proportion below each threshold indicates how much potentially unreliable supervision is filtered out.

As shown in Figure 5, `Gemma2-9B` exhibits a considerably heavier tail in the low- and intermediate-reliability regions. Only 37% of its training instances attain $s_T = 1$, compared with 57%, 62%, and 81% for `Qwen2.5-14B`, `Qwen2.5-7B`, and `Qwen3-4B`, respectively. Consequently, at a permissive threshold, many `Gemma2-9B` instances whose teacher responses only partially cover the

[7] https://github.com/modelscope/ms-swift

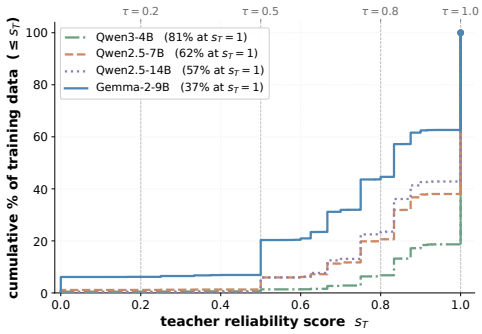

Figure 5: Empirical distributions of teacher reliability scores across backbones.

Table 5: Training cost of GRASP across backbones, including the number of training samples, training steps, and wall-clock time.

Backbone	#Samples	#Steps	Time
`Gemma2-9B`	9,050	283	6.7 h
`Qwen2.5-7B`	15,011	470	6.7 h
`Qwen2.5-14B`	13,847	217	10.9 h
`Qwen3-4B`	19,700	308	11.9 h

Table 6: Effect of rubric completeness as privileged teacher information. **Single** provides one randomly sampled target aspect, whereas **Full** provides the complete set of target aspects.

Category	Gemma2-9B		Qwen2.5-7B		Qwen2.5-14B		Qwen3-4B	
	Single	Full	Single	Full	Single	Full	Single	Full
A&E	0.4117	**0.4372**	0.4196	**0.4408**	0.4448	**0.4736**	0.4980	**0.5215**
L&PD	0.5191	**0.5606**	0.5401	**0.5708**	0.5420	**0.5888**	0.6159	**0.6461**
S&C	0.5559	**0.5787**	0.5604	**0.5955**	0.5853	**0.6347**	0.6651	**0.6863**
Avg.	0.4956	**0.5255**	0.5067	**0.5357**	0.5240	**0.5657**	0.5930	**0.6180**

target rubric aspects remain in the training set. Raising τ filters out a substantially larger proportion of such unreliable supervision, providing a distributional explanation for the more pronounced performance gains of `Gemma2-9B` at higher thresholds in Figure 2. By contrast, the reliability distributions of the other teachers are already more concentrated at $s_T = 1$; increasing τ therefore changes their retained training sets less substantially and leads to smaller incremental gains. Overall, these results suggest that the effectiveness of RTV is closely related to the reliability distribution of the rubric-informed teacher, with stricter filtering being particularly beneficial when the distribution contains substantial low-score mass.

D.2 Training Efficiency

Although GRASP involves on-policy response sampling and full-vocabulary distillation from a frozen teacher, its training cost remains practical. Table 5 reports the number of samples retained after RTV, the corresponding optimization steps, and the wall-clock training time.

Across all four backbones, training requires only 217–470 steps and finishes within 6.7–11.9 hours. At inference time, GRASP takes the same input as RAG, consisting only of the question and retrieved user profile, without requiring rubric aspects or the teacher model. It therefore introduces no additional inference cost over standard RAG. Overall, GRASP provides fine-grained supervision with manageable training costs and no additional inference overhead.

D.3 Effect of Rubric Completeness

To investigate how the completeness of privileged rubric information affects distillation, we construct a single-aspect variant of GRASP. For each training instance, the **Single** setting provides the rubric-informed teacher with only one aspect randomly sampled from E_{x_u}, whereas the **Full** setting provides the complete set of target aspects. All other training configurations remain unchanged. This comparison allows us to determine whether an individual aspect is sufficient for effective supervision or whether the teacher benefits from jointly observing the user's complete information needs.

As shown in Table 6, using the complete rubric set consistently outperforms the single-aspect setting across all categories and backbones. Providing only one aspect gives the teacher a partial view of the user's information needs, resulting in incomplete supervision. In contrast, the full rubric set captures

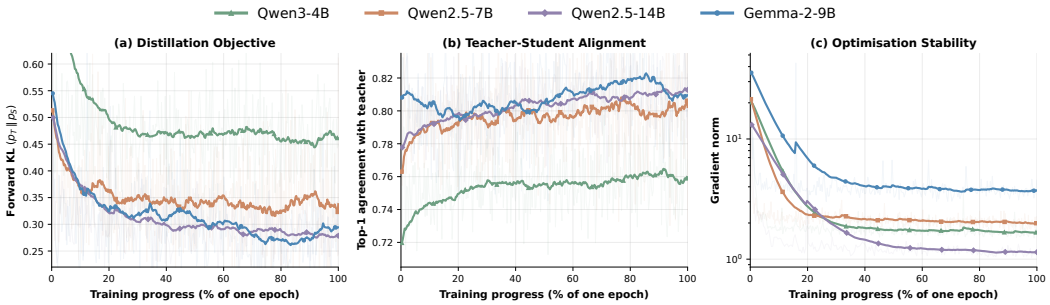

Figure 6: Training dynamics of GRASP across the four backbones. (a) Forward KL $(p_T \,\|\, p_S)$, the objective OPSD minimizes. (b) Fraction of positions where the student and teacher share the same top-1 token. (c) Gradient norm before clipping, on a log scale. Faint traces are raw per-step values; solid lines are an exponential moving average.

complementary information across aspects and enables more comprehensive personalized guidance. These results demonstrate that the effectiveness of GRASP depends not only on the availability of rubric information but also on its completeness.

D.4 Training Dynamics

Figure 6 tracks three training-time signals over the single epoch. Together, they reveal rapid convergence: the forward KL decreases sharply, the teacher–student top-1 agreement increases, and the gradient norm drops by approximately one order of magnitude. Most of these changes occur within the first 20-30% of training, after which all three curves gradually stabilize. These trends indicate stable optimization toward the rubric-informed teacher distributions across different backbones.

E Case Study

To qualitatively examine whether GRASP internalizes user-specific rubric information, we compare its response with that of the RAG baseline using `Qwen2.5-7B`. As shown in Figure 7, both systems receive exactly the same inference-time input, consisting of the question and retrieved user profile. The displayed user narrative and rubric aspects are ground-truth evaluation criteria and are not provided to either system during inference.

The RAG baseline fails to cover all the user-specific requirements, whereas GRASP successfully addresses all three rubric aspects, improving the aspect-coverage score from 0.33 to 1.00. This example demonstrates that GRASP can internalize rubric information during training and generate more comprehensively personalized responses.

F Prompts

For reproducibility, we provide the prompt templates used for the student, the rubric-informed teacher, and the RTV judge. P1 is shared by the student and the RAG baseline, and contains only the current question and the top-10 retrieved user-profile entries. P2 retains the same base prompt but provides the frozen teacher with additional privileged information, including the user's elaboration and the target rubric aspects. This controlled construction ensures that the difference between the student and teacher distributions arises specifically from the rubric information. P3 is used to evaluate whether a generated response covers each target aspect. Given the question, user details, generated response, and one rubric aspect, the judge assigns a coverage score from 0 to 2.

CASE STUDY • Qwen2.5-7B

Question (seen by both systems)
"Is the consumption of bone marrow forbidden?"

User narrative:
*"Is eating bone marrow **the same as eating blood**? **Do any sources address this issue?** Though not the only source I'm interested in, **I'd prefer references to the Tanakh**, if possible."*

RAG baseline — **coverage 0.33**

*"In Jewish dietary laws, known as kashrut, the consumption of bone marrow can be complex. Generally, it is permissible to consume bone marrow, but there are specific conditions. For instance, the bone must be cracked open to remove the marrow, and it should not be eaten while the meat is still warm. [. . .] **Since your question doesn't specify a particular religious or cultural context**, it's important to consult the relevant dietary laws or a knowledgeable rabbi for precise guidance. **If you're interested in a specific tradition or culture, let me know**, and I can provide more tailored information."*

Hedges, then asks the user which tradition they mean — the one thing the user had already been explicit about. No scriptural source is given.

GRASP (ours) — **coverage 1.00**

*"In Jewish dietary laws, known as kashrut, the consumption of bone marrow is generally considered permissible. [. . .] Bone marrow can still be subject to rabbinic restrictions due to the potential for **blood contamination**. According to many authorities, the prohibition against consuming blood (which includes bone marrow) is based on the commandment **'Do not eat any fat or blood' (Leviticus 3:17)**. Therefore, while bone marrow itself is not explicitly forbidden, it is often treated with the same caution as other blood products. [. . .] For practical guidance, consulting a rabbi or a reliable halachic authority is recommended."*

Answers the blood question directly and cites a verse from the Tanakh, the source the user asked for but never named in the question.

Ground-truth rubric aspect	RAG	GRASP
① Comparison to blood consumption	◗	✓
② Religious rulings on consumption	◗	✓
③ References to the Tanakh	✗	✓
aspect-coverage score	**0.33**	**1.00**

✗ not covered (0) ◗ partially covered (1) ✓ fully covered (2)

Figure 7: A case study on LaMP-QA. The RAG baseline misses the user's requirements, while GRASP covers all three rubric aspects.

P1 — Student prompt

[system]
You are a helpful assistant designed to generate personalized responses to user questions. Your task is to answer a user's question from a post in a personalized way by considering this user's past post questions and detailed descriptions of these questions.

`# Your input:`
- The user's current question from a post.
- The user's past post questions and detailed descriptions of these questions.

`# Your task:` Answer the user's current question in a personalized way by considering this user's past post questions [...] to learn about the user's preferences.

`# Your output:` [...] a valid json object in ```` ```json ``` ```` block that contains the following fields:
- `personalized_answer`: contains the personalized answer to the user's current question [...]

[user]

```
# Past post questions and detailed descriptions of these questions:
{profile}
# Current post question:
{question}
```

(top-10 Contriever-ranked entries)

P2 — Teacher prompt (= P1 + privileged information, shaded)

[system]
(identical to P1, then appended:)

`# Additional private context (this turn only):`
You are additionally given private editorial notes describing what this specific user is actually looking for: their own elaboration of the question, and the concrete aspects they expect the answer to address. Use these notes to decide what to cover and how to prioritise it. **Never mention, quote, restate or allude to the notes themselves** — the reader must not be able to tell that you had them. Your output format is unchanged.

[user]
(identical to P1, then appended:)

```
# What this user is actually looking for (private editorial notes):
## The user's own elaboration of the question:
{details}
## The specific aspects this user expects the answer to address:
- {aspect}
  - why it matters to this user: {reason}
  - in the user's own words: {evidence}
```

(repeated for every aspect in the rubric)

P3 — Judge

[system]
You are a fair and insightful judge with exceptional reasoning and analytical abilities. Your task is to evaluate a user's question, a generated response to that question, and an aspect that is important to the user. Based on this information, identify if the aspect is addressed in the generated response.

`# your output:` [...] a valid json object [...] that contains:
- `match_score`: A score between 0 to 2 [...] where **0** means the response does not cover this aspect, **1** means the response somewhat covers this aspect, and **2** means the response covers this aspect very well.

[user]

```
question: {question}
details: {details}
response: {response}
aspect: {aspect, reason, evidence}
```