



TagPR: Tag-Guided Process Supervision for Personalization Reasoning in Large Language Models

Song Jin¹, Juntian Zhang¹, Ruyu Lyu¹, Yong Liu¹,
Xun Zhang², Yufei Zhang², Fei Jiang², Guojun Yin², Wei Lin², Rui Yan^{3*}

¹Renmin University of China, ²Meituan, ³Wuhan University

{jinsong8, zhangjuntian, ruyu1027, liuyonggsai}@ruc.edu.cn,

{zhangxun12, zhangyufei08, jiangfei05, yinguojun02, linwei31}@meituan.com,
rui.yan@whu.edu.cn

Abstract

Recent advancements have endowed Large Language Models with impressive general reasoning capabilities. However, these reasoning models often perform worse than non-reasoning models on personalization tasks. While some methods use outcome-based RL to improve personalization reasoning, they fail to supervise the reasoning process. As a result, models may reach correct answers through flawed reasoning chains, limiting further improvement. To address this, we propose TagPR, a novel framework that adds semantic tags to the reasoning process for step-by-step guidance. TagPR first automatically generates a structured, tagged dataset for Supervised Fine-Tuning. It then employs a multi-stage RL process guided by a composite reward signal, which integrates tag-based process supervision with a novel Personalization Reward Model with User Embeddings to achieve fine-grained alignment with user-specific logic. Extensive experiments on public LaMP, LongLaMP, PGraphRAG, and a self-constructed dataset demonstrate that our approach achieves state-of-the-art results, delivering an average improvement of 32.65% over the base model across all LaMP benchmark tasks. Our work demonstrates that tag-guided process supervision is an effective approach for personalization reasoning.

1 Introduction

Research on large-model reasoning now spans mathematics, coding, and multimodal domains (Guo et al., 2025; Yu et al., 2026; Jin et al., 2026b; Zhang et al., 2026), yet these advances do not readily translate to personalization tasks. This domain is increasingly critical, powering a broad range of real-world applications, including personalized recommendation, intelligent writing assistance, targeted advertising, and user-tailored conversational agents. Effective personalization

demands far more than generic logic. It requires personalization reasoning: the ability to carefully analyze a user’s historical data, infer their unique preferences and thought patterns, and synthesize this understanding to generate responses that are precisely tailored to each user.

Surprisingly, even state-of-the-art reasoning-centric LLMs struggle in this domain, frequently failing to surpass non-reasoning baselines on personalization benchmarks. This performance deficit stems from a fundamental misalignment: models optimized for general-purpose reasoning inherently prioritize their internal, generalized logic over the user’s profile. Consequently, they generate responses that are either overly generic or directly contradict established user preferences. Recent pioneering works, such as R2P (Luo et al., 2025) and RPM (Kim et al., 2026), have identified this limitation. Although these approaches mitigate the issue by employing prompt templates or pre-constructed reasoning paths, they function merely as external scaffolds rather than fundamentally improving the model’s intrinsic capacity for user-centric reasoning.

Our core motivation stems from the observation that personalization reasoning is not a monolithic act of intuition but a structured process with multiple steps involving the analysis of user history, the identification of recurring patterns, and the application of those patterns to new contexts. The opaque and unconstrained reasoning of standard LLMs is poorly suited for this task. Furthermore, existing RL-based approaches for eliciting personalization reasoning are often hampered by their reliance on outcome-based rewards (Zhang et al., 2025; Li et al., 2025). Such signals are inherently sparse because a correct final answer does not necessarily imply a valid reasoning process, which limits further improvement. The ability to provide structured guidance signals within the model’s internal reasoning logic therefore becomes highly

* Corresponding author.

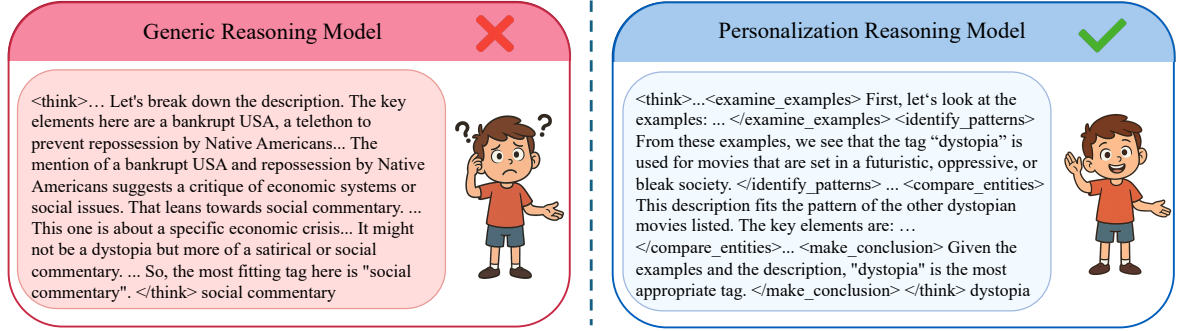


Figure 1: A comparison of reasoning paths. Left: The Generic Reasoning Model (Qwen3-8B) uses free-form logic, leading to an incorrect tag (“social commentary”). Right: Our Personalization Reasoning Model follows a structured path to correctly infer the user-specific tag (“dystopia”).

valuable. To this end, we introduce TagPR. Instead of allowing the model to reason implicitly, we compel it to externalize its logic into a sequence of discrete, interpretable steps, each marked with a semantic tag (e.g., `<examine_examples>`, `<identify_patterns>`). These tags act as cognitive waypoints, transforming the complex task of personalization into a manageable, explicit procedure that the model can learn to execute robustly, as illustrated in Figure 1. By transforming implicit reasoning steps into an explicit structure, our approach enables precise process-level supervision based on tags embedded in the reasoning trajectory during the RL stage.

This is achieved through a synergistic training strategy. First, we pioneer a data-driven pipeline to automatically generate a new dataset of reasoning chains labeled with these semantic tags. We use this dataset for Supervised Fine-Tuning (SFT) to instill the foundational grammar of structured and personalized thought. Following this, we employ a multi-stage RL process to refine this capability. This RL phase is guided by a novel composite reward that combines tag-based structural constraints with a fine-grained signal from our new Personalization Reward Model with User Embeddings (PRMU), which explicitly aligns the model’s reasoning with user-specific logic. Our key contributions are threefold:

I. We formulate personalization reasoning as a sequence of semantically tagged actions and introduce a data-driven pipeline to automatically generate and label reasoning chains with semantic tags.

II. We introduce a synergistic SFT and multi-stage RL training framework. This process is guided by a unique composite reward signal that integrates tag-based constraints and our novel Per-

sonalization Reward Model with User Embeddings (PRMU) for fine-grained alignment with user logic.

III. We demonstrate through extensive experiments on public LaMP, LongLaMP, PGraphRAG and a self-constructed dataset that **TagPR** achieves state-of-the-art results.

2 Related Work

Large Language Model Personalization LLM personalization has evolved rapidly. A dominant approach is retrieval-augmented generation, with innovations including feedback-optimized retrieval (Salemi et al., 2024a) and generation-calibrated retrievers (Mysore et al., 2024). PAG (Richardson et al., 2023) enhances retrieval by integrating user history summarization. Beyond retrieval, research has explored core personalization components (Wu et al., 2024). DPL (Qiu et al., 2025) models inter-user differences to capture unique preferences. Parameter-efficient approaches include OPPU (Tan et al., 2024b) with user-specific lightweight modules, PER-PCS (Tan et al., 2024a) for collaborative PEFT sharing, plug-and-play user embeddings (PPlug) (Liu et al., 2025), and HYDRA (Zhuang et al., 2024) for black-box personalization. Additional methods include multi-stage decomposition (Li et al., 2023) and multi-objective parameter merging (P-Soups) (Jang et al., 2023). Recent work has further expanded personalization research toward personalized user-agent simulation (Jin et al., 2025) and user-specific evaluation (Qiu et al., 2026).

Personalization Reasoning Personalization reasoning represents an emerging intersection of reasoning capabilities and personalization tasks. Early approaches primarily use prompting strategies for black-box models: RPM (Kim et al., 2026) con-

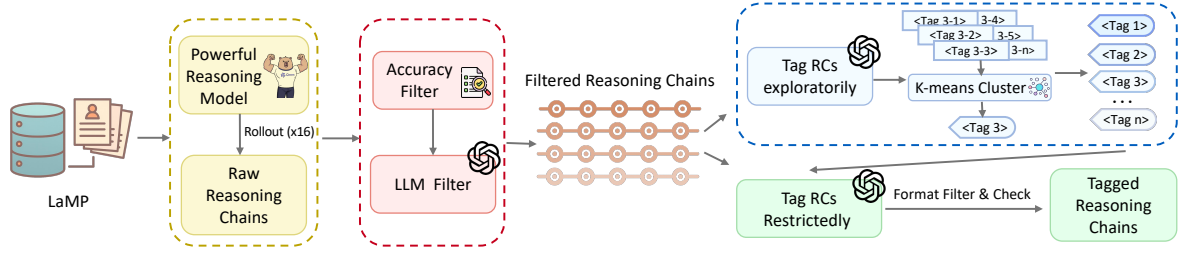


Figure 2: The pipeline for constructing our Tagged Reasoning Chains dataset. The process includes raw chains generation from LaMP, a two-stage quality filter, and a two-phase tagging procedure where primary tags are first defined via clustering and then applied in a restricted final annotation.

structs individualized reasoning paths from user history, while R2P (Luo et al., 2025) employs hierarchical reasoning templates. Fine-tuning approaches include generating reasoning paths followed by iterative self-training (Salemi et al., 2025), and reinforcement learning for preference inference through extended inductive reasoning (Li et al., 2025). Most closely related to our work, PrLM (Zhang et al., 2025) uses contrastive reward models with reinforcement learning for reasoning in personalization generation tasks. While these methods have made notable progress, they typically address personalization reasoning through either template-guided generation or reward-based optimization without fundamentally restructuring how models approach the multi-faceted nature of personalization tasks.

3 Methodology

This section presents the methodology for **TagPR**. We formulate the task and overview the framework in Section 3.1, detail the reasoning structure initialization in Section 3.2, and present the Tag-Guided Reinforcement Learning in Section 3.3.

3.1 Task Formulation & Overview

We define personalized reasoning as the task of generating a user-specific response y to a query x , conditioned on the user’s profile $P_u = \{(x_i, y_i)\}_{i=1}^{N_u}$, where N_u is the total number of historical records available for user u , and each record (x_i, y_i) is a past input–output pair. To incorporate reasoning into this process, the task is extended to first generate an explicit reasoning chain c before producing the final response y . Conditioned on the query x and a relevant profile subset $p_u \subseteq P_u$, the objective for a model parameterized by θ is to maximize the joint likelihood of the reasoning chain and the

response:

$$p(c, y|x, p_u; \theta) = p(c|x, p_u; \theta) \cdot p(y|c, x, p_u; \theta). \quad (1)$$

The core challenge is to ensure that the reasoning chain c is coherent and faithful to the user’s profile p_u , while the response y remains consistent with this explicit reasoning.

The overall **TagPR** framework addresses this challenge through two sequential phases. First, a lightweight *Reasoning Structure Initialization* stage injects the syntactic grammar of tagged reasoning into the model via a small amount of high-quality supervised data. Second, a *Tag-Guided Reinforcement Learning* stage refines the model’s personalization reasoning through tag-based process supervision and fine-grained user-aware reward signals.

3.2 Reasoning Structure Initialization

Tagged Reasoning Chains Construction Pipeline

Automated construction pipelines are increasingly used to build training resources (Jin et al., 2026a). To equip the model with a structured reasoning paradigm, we design a multi-stage pipeline to construct a tagged reasoning dataset (Figure 2). We randomly sample 1,000 instances from each of the six training tasks in the LaMP dataset (Salemi et al., 2024b), employing Qwen3-235B-A22B-Thinking-2507 (Yang et al., 2025) to generate 16 candidate reasoning chains per instance via rollout.

To ensure dataset integrity, we applied a rigorous two-stage filter. First, an *accuracy filter* retained only correctly answered samples. Second, an *LLM filter* used GPT-4o (Hurst et al., 2024) to score these reasoning chains on logic, factual accuracy, completeness, and conciseness. Only instances scoring >15 were retained for subsequent annotation (Prompt in Appendix E.1.1).

The filtered reasoning chains then underwent a two-phase tagging procedure. In the first

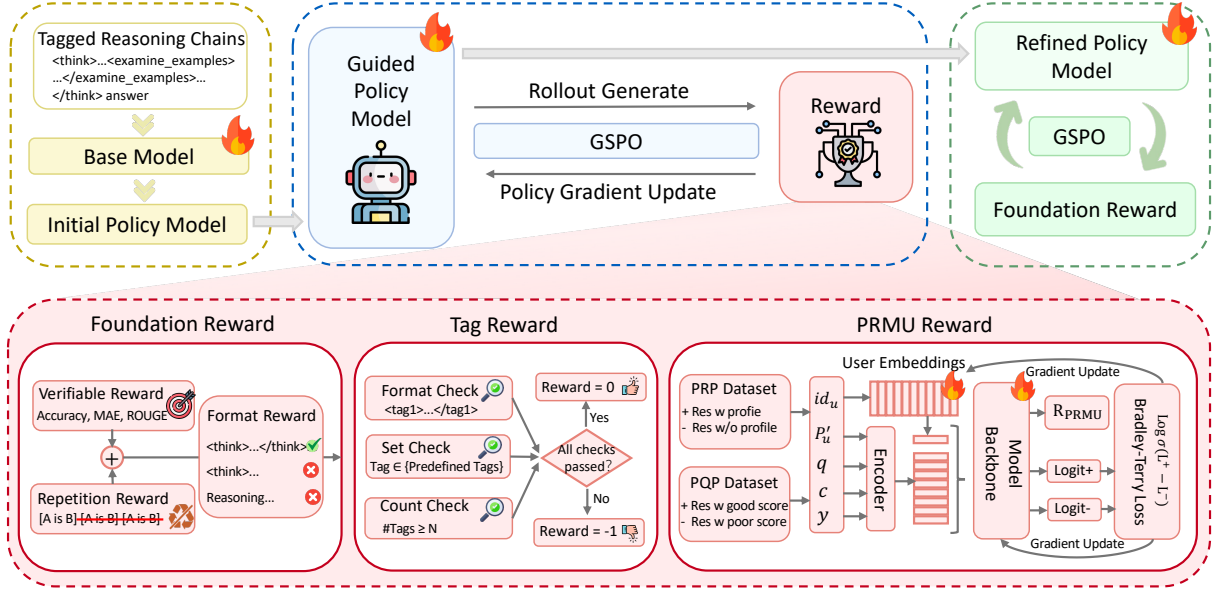


Figure 3: Overview of our proposed multi-stage training framework. An initial policy model is obtained via SFT on tagged reasoning chains. The model is then refined through two sequential RL phases: (1) a **Guided RL** stage using a complex, multi-component reward (including Tag and PRMU rewards) to learn structured reasoning, and (2) an **Exploratory RL** stage with a Foundation reward to further boost performance.

phase, *exploratory tagging* used GPT-4o to generate unrestricted tags based on the reasoning content, which were then grouped via K-means clustering (MacQueen, 1967) into 9 primary tags: <analyze_input>, <examine_examples>, <identify_patterns>, <evaluate_reference>, <compare_entities>, <synthesize_findings>, <make_decision>, <verify_conclusion> and <formulate_conclusion>. In the second phase, *restricted tagging* constrained GPT-4o to re-annotate all reasoning chains using only these 9 tags for consistency. Detailed tagging prompts are provided in Appendix E.1.2 and Appendix E.1.3. **Foundational SFT for Syntax Bootstrapping** Our framework employs a compact but high-quality dataset at the SFT stage, retaining approximately 2,000 instances. Rather than encoding personalization knowledge, the objective of this stage is to instill the syntactic grammar of structured reasoning, specifically the capacity to produce well-formed tagged thought steps such as <analyze_input> and <identify_patterns>. Without this initialization, tag-based process supervision in the subsequent RL stage would be infeasible. We minimize the negative log-likelihood of generating reasoning chain c and answer y given query q and user profile p_u :

$$\mathcal{J}_{\text{SFT}}(\theta) = - \sum_{(q, p_u, c, y) \in \mathcal{D}} \log P_{\theta}(c, y | q, p_u), \quad (2)$$

where $\mathcal{D}$ is the labeled dataset and θ are the model parameters.

3.3 Tag-Guided Reinforcement Learning

3.3.1 Tag-Based Process Supervision

By compelling the model to externalize its logic into discrete, semantically tagged steps, we transform the opaque internal reasoning process into an observable sequence that can be directly supervised. This resolves the degeneracy of outcome-only rewards, which cannot distinguish between a model that carefully analyzes user history and one that ignores it but guesses correctly.

The Tag Reward R_{tag} enforces both the structural and semantic correctness of the tagged reasoning chain. It operates as a penalty-based signal that applies a set of logical checks to the generated chain c :

$$R_{\text{tag}}(c, y) = \begin{cases} 0 & \text{if all logical checks pass} \\ -1 & \text{otherwise} \end{cases}. \quad (3)$$

The logical checks include three verifications: (1) *format verification*, ensuring that each tag is syntactically well-formed with correct opening and closing delimiters; (2) *vocabulary verification*, ensuring that every tag used belongs to the predefined set of 9 primary tags established during data construction; and (3) *coverage verification*, ensuring that the reasoning chain meets a minimum tag

count, preventing degenerate solutions that satisfy format requirements with a single trivial tag. Together, these checks enforce a cognitive structure on the model’s thought process, guiding it along a principled chain of thought.

3.3.2 Fine-Grained Alignment via PRMU

While tag-based supervision effectively guides the model toward a principled personalization reasoning paradigm, fine-grained alignment with individual user preferences requires an additional user-aware signal. To provide this signal, we introduce the Personalization Reward Model with User Embeddings (PRMU).

Unlike standard reward model architectures, PRMU incorporates learnable user embeddings E_u to capture individual preferences. This architectural modification enables it to provide a granular reward signal that prioritizes reasoning which is not only accurate but also highly tailored to the user’s profile. PRMU is trained on two preference datasets ($\sim 10k$ samples each). The Profile-Reasoning Preference (PRP) dataset contrasts responses generated with a user profile (preferred) against those generated without (rejected), teaching the model to value profile utilization. The Personalized-Quality Preference (PQP) dataset contains pairs of personalized responses where preference is determined by correctness or ROUGE score, thereby training the model to discern reasoning quality among personalized outputs.

Initialized from Skywork-Reward-V2-Qwen3-0.6B (Liu et al., 2026), our PRMU architecture first maps a user ID id_u to its corresponding embedding E_u . This embedding, along with the query, profile, and reasoning chain, is processed to produce a scalar logit. Both the base reward model parameters θ_r and the user embeddings E are jointly optimized by minimizing the Bradley-Terry (Bradley and Terry, 1952) preference loss:

$$\mathcal{J}(\theta_r, E) = -\mathbb{E}_{(x^+, x^-) \sim \mathcal{D}} \left[\log \sigma \left(f_{\text{PRMU}}(x^+) - f_{\text{PRMU}}(x^-) \right) \right], \quad (4)$$

where x^+ and x^- represent the preferred and rejected input tuples from our preference dataset $\mathcal{D}$. The final normalized reward score R_{PRMU} for the RL phase is obtained via a sigmoid function:

$$R_{\text{PRMU}} = \sigma(f_{\text{PRMU}}(id_u, q, p_u, c, y | E_u; \theta_r)). \quad (5)$$

3.3.3 Multi-Stage Policy Optimization

With the process supervision signal from R_{tag} and the user-aware signal from R_{PRMU} established, we

now describe how these components are integrated into a complete multi-stage policy optimization strategy, as illustrated in Figure 3.

Composite Reward Design The full reward function used in the first RL stage combines five distinct signals. Beyond R_{tag} and R_{PRMU} , we define three additional components. The Verifiable Reward R_v measures the factual correctness of the response y against a ground-truth reference y^* :

$$R_v(y, y^*) = \begin{cases} \text{Accuracy}(y, y^*) & \text{classification} \\ \text{ROUGE}(y, y^*) & \text{generation} \end{cases}. \quad (6)$$

The Format Reward R_f provides a binary signal to enforce the overall structural integrity of the output:

$$R_f(c, y) = \begin{cases} 1 & \text{if } c, y \text{ match expected format} \\ 0 & \text{otherwise} \end{cases}. \quad (7)$$

The Repetition Reward R_{rep} penalizes textual redundancy to improve fluency:

$$R_{\text{rep}}(c, y) = -\frac{|T_n(c, y)| - |U_n(c, y)|}{|T_n(c, y)| + \delta}, \quad (8)$$

where T_n and U_n are the multiset and set of n -grams in the generation respectively, and δ is a small constant for numerical stability.

These components are integrated into the composite reward R as follows:

$$R = \alpha \cdot (R_v + R_{\text{rep}}) \cdot R_f + \beta \cdot R_{\text{tag}} + \gamma \cdot R_{\text{PRMU}}, \quad (9)$$

where we set the balancing hyperparameters $\alpha = \beta = 0.8$ and $\gamma = 0.2$. These values deliberately prioritize logical correctness over personalization to prevent factuality degradation and reward hacking. We adopt a multiplicative form for R_f to condition other signals on format compliance, ensuring that rewards are only granted for structurally valid outputs. A sensitivity analysis in Appendix B.3 confirms robustness across a reasonable range of hyperparameter variations.

Guided RL for Personalization Reasoning In the first RL stage, the SFT-initialized model is refined using the full composite reward R . This stage teaches the model to simultaneously satisfy structural constraints, factual correctness, and user-specific alignment. For policy optimization, we utilize the GSPO algorithm (Zheng et al., 2025), which offers greater training stability by operating at the sequence level. The GSPO objective is:

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}_{\substack{q \sim \mathcal{D} \\ \{c_i, y_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)}} \left[\frac{1}{G} \sum_{i=1}^G \min \left(s_i(\theta) \hat{A}_i, \text{clip}(s_i(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i \right) \right], \quad (10)$$

Dataset →		LaMP-1		LaMP-2		LaMP-3		LaMP-4		LaMP-5		LaMP-7	
Method	R	ACC ↑	F1 ↑	ACC ↑	F1 ↑	MAE ↓	RMSE ↓	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑
<i>Previous Method</i>													
Zero-shot	✗	0.498	0.470	0.318	0.244	0.639	0.983	0.144	0.125	0.417	0.351	0.465	0.413
Zero-shot-R	✓	0.477	0.483	0.389	0.347	0.416	0.778	0.131	0.115	0.354	0.306	0.431	0.383
RAG	✗	0.668	0.645	0.414	0.361	0.354	0.710	0.158	0.139	0.453	0.384	0.473	0.419
RAG-R (Base)	✓	0.717	0.722	0.453	0.413	0.291	0.645	0.152	0.137	0.434	0.365	0.439	0.391
PAG	✓	0.677	0.649	0.420	0.367	0.337	0.675	0.167	0.148	0.452	0.385	0.479	0.426
PAG-R	✓	0.731	0.736	0.470	0.417	0.289	0.627	0.160	0.142	0.408	0.349	0.428	0.380
SFT	✗	0.670	0.654	0.511	0.461	0.273	0.569	0.196	0.178	0.455	0.393	0.498	0.445
SFT-R	✓	0.720	0.723	0.460	0.419	0.343	0.889	0.161	0.147	0.438	0.377	0.439	0.389
SFT-Ind	✗	0.717	0.717	0.532	0.488	0.269	0.568	0.207	0.187	0.463	0.411	0.507	0.454
SFT-Ind-R	✓	0.729	0.731	0.463	0.419	0.366	1.001	0.151	0.138	0.432	0.374	0.433	0.383
PPlug	✗	0.698	0.699	0.535	0.489	0.261	0.532	0.213	0.195	0.486	0.434	0.521	0.465
HYDRA-Adapter	✗	0.692	0.692	0.482	0.455	0.320	0.663	0.159	0.138	0.457	0.395	0.483	0.423
R2P	✓	0.729	0.730	0.487	0.459	0.267	0.557	0.176	0.155	0.459	0.396	0.489	0.426
PrLM	✓	0.731	0.731	0.534	0.504	0.288	0.635	0.183	0.169	0.499	0.438	0.513	0.459
<i>State-of-the-Art LLMs</i>													
GPT-4o	✗	0.733	0.733	0.542	0.512	0.254	0.554	0.191	0.175	0.470	0.407	0.475	0.419
Qwen3-235B-A22B	✓	0.715	0.720	0.511	0.488	0.280	0.633	0.177	0.158	0.450	0.396	0.455	0.409
DeepSeek-R1	✓	0.740	0.744	0.522	0.488	0.280	0.644	0.181	0.166	0.451	0.399	0.447	0.397
Gemini-2.5-Pro	✓	0.761	0.761	0.582	0.548	0.271	0.594	0.222	0.202	0.495	0.438	0.480	0.425
Gemini-3-Pro	✓	0.780	0.780	<u>0.595</u>	<u>0.560</u>	0.260	0.565	<u>0.230</u>	<u>0.210</u>	0.505	0.445	0.494	0.436
<i>Our Method</i>													
TagPR w/o RL	✓	0.722	0.724	0.456	0.416	0.339	0.878	0.159	0.145	0.440	0.378	0.437	0.386
TagPR w/o SFT	✓	0.747	0.747	0.543	0.510	0.271	0.593	0.194	0.181	0.502	0.441	0.525	0.469
TagPR w/o Tagging	✓	0.748	0.749	0.546	0.511	0.264	0.447	0.199	0.185	0.518	0.448	0.528	0.471
TagPR w/o Tag-R	✓	0.749	0.749	0.545	0.511	0.272	0.595	0.197	0.183	0.506	0.441	0.524	0.469
TagPR w/o PRMU	✓	0.768	0.769	0.557	0.514	0.246	0.393	0.205	0.190	0.522	0.453	0.545	0.490
TagPR w/o Exp-RL	✓	0.789	0.789	0.582	0.543	<u>0.233</u>	<u>0.302</u>	0.224	0.204	<u>0.536</u>	<u>0.467</u>	<u>0.560</u>	<u>0.501</u>
TagPR	✓	0.803	0.803	0.598	0.557	0.218	0.263	0.234	0.213	0.542	0.471	0.565	0.507

Table 1: Main results on the LaMP benchmark, comparing TagPR against a wide range of baselines, including previous methods, state-of-the-art LLMs, and our own ablation studies. **Bold** indicates the best performance, and underline indicates the second-best. The “R” column denotes whether a reasoning step is used (✓).

where $s_i(\theta)$ is the sequence-level importance sampling ratio and $\hat{A}_i$ is the standardized advantage for each response in a generated group of size G .

Exploratory RL for Performance Refinement In the final stage, we address performance plateaus that arise from the constraints imposed by the composite reward. Since the model has internalized the reasoning structure, we drop the scaffolding of R_{tag} and R_{PRMU} and employ a simplified Foundation Reward targeting fundamental quality metrics:

$$R_{\text{foundation}} = (R_v + R_{\text{rep}}) \cdot R_f. \quad (11)$$

Removing these constraints allows the model to explore the policy space more freely, optimizing core performance without structural penalties. Ablation studies confirm the superiority of this two-stage design over a single continuous RL stage.

4 Experiments

4.1 Experimental Setup

Implementation Details We employ Qwen3-8B as our base model. Our training process consists

of SFT on the dataset described in Section 3.2, followed by a two-stage RL phase using data sampled from the LaMP training set. We evaluate our model on the LaMP benchmark, a standard for assessing personalization, reporting results on its validation set as the test set is not public.

Baselines We conduct a comprehensive comparison against a wide spectrum of baselines. These include: (1) standard methodologies such as Zero-shot, RAG, PAG (Richardson et al., 2023), SFT, SFT-Ind, and their reasoning-enhanced variants (-R); (2) advanced personalization (PPlug (Liu et al., 2025), HYDRA-Adapter (Zhuang et al., 2024)) and reasoning-focused techniques (R2P (Luo et al., 2025), PrLM (Zhang et al., 2025)); and (3) state-of-the-art large language models like Gemini-2.5-Pro (Comanici et al., 2025). One primary baseline is the RAG-R (Base) method, which shares our configuration with the original Qwen3-8B model.

More detailed descriptions of all baselines, hyperparameters, benchmark, evaluation metrics, and experimental configurations are in the Appendix A.

4.2 Main Results

The results, presented in Table 1, demonstrate that **TagPR** establishes a new state-of-the-art across all six tasks of the LaMP benchmark. It consistently outperforms a suite of baselines, including prior personalization methods, reasoning-focused models, and larger proprietary LLMs used off-the-shelf without task-specific training.

To isolate the efficacy of our framework, we first conduct an ablation study comparing **TagPR** against a RAG-R (Base) method. This baseline shares an identical configuration but utilizes the original Qwen3-8B model. The performance gains are substantial: **TagPR** achieves a 55.5% relative improvement in ROUGE-L on the LaMP-4 generation task, boosts the F1-score by 34.9% on the challenging LaMP-2 classification task, and reduces the MAE by 25.1% on the LaMP-3 task. These results underscore that our synergistic training paradigm significantly enhances the model’s personalization reasoning capabilities.

Notably, our fine-tuned 8B parameter model outperforms larger proprietary models evaluated without task-specific training. For instance, on the LaMP-1 task, **TagPR**’s accuracy of 0.803 surpasses both Gemini-3-Pro (0.780) and GPT-4o (0.733).

4.3 Ablation Study

To dissect the contribution of each component within our framework, we conducted a comprehensive ablation study, with results presented in Table 1. Our analysis reveals a clear incremental contribution from each module.

The results first highlight the critical roles of the foundational training stages. The initial SFT phase is essential for bootstrapping the model with our tagged reasoning syntax, and its removal (TagPR w/o SFT) leads to a noticeable performance drop. Building upon this, the multi-stage RL is vital for refining this structure into high-quality personalized logic, and the SFT-only model (TagPR w/o RL) leaves a considerable gap to the full model. Comparable results between TagPR w/o RL and SFT-R indicate that tags yield no measurable gain during SFT alone, but become effective as process supervision during RL.

Furthermore, our novel reward signals prove effective. The PRMU reward provides a crucial user-aware signal, and its removal (TagPR w/o PRMU) leads to a consistent decline across all tasks. Regarding the tag-based reward, note that TagPR w/o

Tagging replaces the 9 personalization-specific tags with generic unstructured reasoning, while TagPR w/o Tag-R retains the tags but removes their associated reward. Both variants underperform the full model, confirming that the semantic content of the personalization-specific tags and their process supervision together drive consistent gains. Finally, our two-stage training design is validated as superior to a single, continuous RL stage (TagPR w/o Exp-RL). Collectively, these findings affirm that the synergistic integration of each carefully designed component is the key to TagPR’s success.

4.4 Generalization Assessment

To evaluate whether **TagPR** learns generalizable personalization capabilities, we assess its zero-shot generalization performance on two other public benchmarks and a new self-constructed benchmark.

We constructed a benchmark from Dianping¹, a prominent Chinese user-generated content platform. This setup poses a stringent test involving unseen domains, task formats, and a different language. A more detailed benchmark introduction is provided in the Appendix F. As shown in Table 2, **TagPR** demonstrates exceptional generalization capabilities. It achieves state-of-the-art results across the benchmark, securing the top score on the majority of metrics and leading proprietary models like GPT-4o evaluated without task-specific training. **TagPR** creates a highly generalizable personalization reasoning model effective across diverse domains, tasks, and languages.

Dataset →	Dianping-Content		Dianping-Title		Dianping-Paraph	
Method	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑
RAG	0.200	0.151	0.209	0.184	0.598	0.568
RAG-R (Base)	0.183	0.144	0.197	0.173	0.517	0.461
SFT	0.189	0.123	0.228	0.210	0.603	0.571
SFT-R	0.187	0.145	0.198	0.177	0.498	0.423
GPT-4o	0.207	0.168	<u>0.236</u>	<u>0.211</u>	<u>0.606</u>	<u>0.573</u>
Gemini-2.5-Pro	0.217	<u>0.170</u>	0.215	0.195	0.564	0.475
TagPR	<u>0.216</u>	0.171	0.240	0.218	0.617	0.583

Table 2: Zero-shot cross-lingual generalization performance on the three Dianping datasets. The best results are in **bold**, and the second-best are underlined. Our TagPR demonstrates superior performance.

To further assess the generalization capabilities of TagPR, we evaluated the model on partial test sets of LongLaMP (Kumar et al., 2024) and PGraphRAG (Au et al., 2025) datasets in the Appendix C. These benchmarks feature ultra-long user histories and sparse user profiles, respectively, pro-

¹<https://www.dianping.com/>.

viding complementary tests of personalization reasoning under challenging conditions. As shown in Table 8 in Appendix, our model demonstrated superior performance, verifying the robust generalization of its learned personalized reasoning capability.

4.5 Robustness Assessment

We evaluate the robustness of **TagPR** against baselines SFT and Base by varying two key factors: user profile length and profile retrieval method. Figure 6 in Appendix D presents the results on the representative LaMP-2 and LaMP-4 tasks.

First, we analyze the effect of profile length by varying the number of historical interactions from 2 to 32. The top row of Figure 6 shows that **TagPR** consistently outperforms the baselines across all lengths. Notably, TagPR’s performance improves rapidly and starts to plateau with just 8 interactions, indicating its high data efficiency in distilling user preferences. In contrast, the baselines show more gradual improvements and maintain a significant performance gap.

Second, we assess the model’s sensitivity to the profile retrieval method. We compare our default Recency-based retriever with three alternatives: a sparse retriever (BM25 (Robertson and Zaragoza, 2009)), a dense retriever (Contriever (Izacard et al., 2021)), and Random selection. As shown in the bottom row, **TagPR** demonstrates remarkable stability and maintains its superior performance across all retrieval strategies. Even with randomly selected profiles, TagPR’s performance degradation is minimal, suggesting its reasoning process can effectively identify and utilize relevant information regardless of the profile quality.

4.6 Further Analysis

This section validates the PRMU design and analyzes tagged reasoning chains, with further case studies and semantic consistency verification in Appendix B.

4.6.1 Personalization Reward Model Design

To validate our proposed PRMU, we conducted a comprehensive ablation study to assess the contribution of its core components. Table 3 compares four configurations: our full PRMU, PRMU without user embeddings (w/o UE), a baseline using an untrained reward model (Untrained RM), and a baseline with no reward model (w/o RM). Our findings first reveal that employing an off-

the-shelf reward model offers no consistent advantage over having no reward model at all. In fact, it proved detrimental in certain cases, yielding a noisy and misaligned signal. This underscores the necessity of fine-tuning the reward model on domain-specific data. Next, training the reward model on our personalization dataset, even without user-specific information (PRMU w/o UE), yields improvements across all metrics. The most significant performance gains are realized with the full PRMU model. By integrating user embeddings to provide a user-aware reward, PRMU consistently outperforms all other variants.

4.6.2 Tagged Reasoning Chains Analysis

Reasoning Length To assess reasoning efficiency, we compare the average token count of reasoning chains generated by our trained model against the original Qwen3-8B on the LaMP validation set. As illustrated in Figure 4 (Left), **TagPR** consistently produces more concise reasoning chains, achieving an average token reduction of over 50%. For instance, significant reductions of 59% on LaMP-2 and 62% on LaMP-4 are observed. Unlike the Base model’s tendency toward verbose reasoning, our framework guides the model along a direct logical path, effectively pruning irrelevant steps. This demonstrates that effective personalization relies on the precision of reasoning, not its length, leading to enhanced computational efficiency.

Reasoning Tags As shown in Figure 4 (Right), the model relies heavily on contextual grounding (<analyze_input> 38.2%, <examine_examples> 27.7%). Synthesis and decision tags (<identify_patterns> 12.4%, <compare_entities> 4.7%, <make_decision> 17.1%) occupy a smaller proportion. This indicates a highly evidence-driven approach, prioritizing input understanding over immediate conclusions.

4.6.3 Reasoning Content Analysis

To further investigate the reasoning processes qualitatively, we generated word clouds from the reasoning chains produced by the baseline Qwen3-8B and our **TagPR** model on the LaMP validation set, as shown in Figure 5 in Appendix B. The visualization reveals a stark contrast in their reasoning styles.

The word cloud for the baseline model is populated by general, conversational terms such as “maybe”, “think”, “example”, and “review”. This indicates a descriptive, narrative-style reasoning

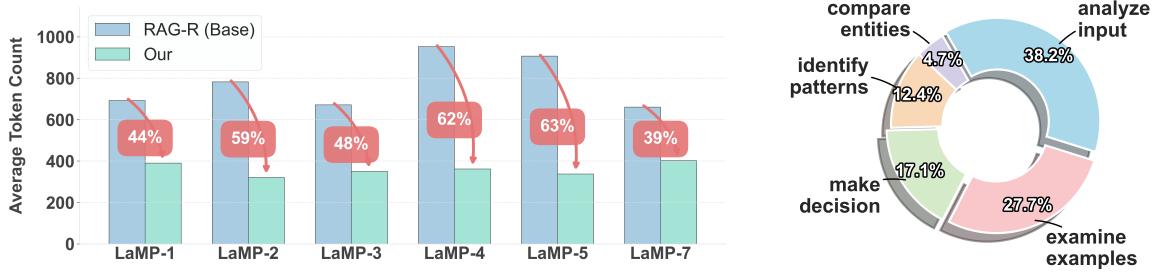


Figure 4: **Left:** Comparison of reasoning chain length between TagPR and Base on the LaMP validation set. Our model achieves substantial length reductions across all tasks, indicating greater efficiency. **Right:** Frequency distribution of the five core reasoning tags generated by our model.

Dataset →	LaMP-1		LaMP-2		LaMP-3		LaMP-4		LaMP-5		LaMP-7	
Method	ACC ↑	F1 ↑	ACC ↑	F1 ↑	MAE ↓	RMSE ↓	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑
w/o RM	0.768	0.769	0.557	0.514	0.246	0.393	0.205	0.190	0.522	0.453	0.545	0.490
Untrained RM	0.771	0.772	0.533	0.495	0.246	0.361	0.207	0.195	<u>0.536</u>	0.459	0.545	0.487
PRMU w/o UE	<u>0.784</u>	<u>0.784</u>	<u>0.581</u>	<u>0.541</u>	<u>0.231</u>	<u>0.299</u>	<u>0.215</u>	<u>0.197</u>	<u>0.536</u>	<u>0.467</u>	<u>0.558</u>	<u>0.501</u>
PRMU	0.803	0.803	0.598	0.557	0.218	0.263	0.234	0.213	0.542	0.471	0.565	0.507

Table 3: Ablation study of PRMU components across LaMP benchmarks. Best and second-best results are in **bold** and underlined.

process, where the model verbalizes a general thought process rather than executing a structured plan. In sharp contrast, the **TagPR** word cloud prominently features action-oriented keywords like “examine_examples”, “analyze_input”, “identify_patterns”, and “make_decision”, which are the core components of the functional tags introduced in our framework. This shift demonstrates that **TagPR** successfully learns to adopt an explicit, structured, and interpretable reasoning schema. Instead of merely describing its thought process, the model actively executes a sequence of defined logical steps, confirming a more efficient and targeted approach to personalization reasoning.

5 Conclusion

In this work, we presented TagPR, a novel framework designed to bridge the fundamental gap between general reasoning and personalization reasoning in LLMs. To achieve this, we developed a synergistic training paradigm starting with a data-driven pipeline for SFT to instill tagged reasoning syntax. This is followed by a multi-stage RL process. The initial guided phase utilizes a composite reward that features tag-based process supervision and our novel Personalization Reward Model with User Embeddings. A subsequent exploratory phase is then applied to refine core performance. Extensive evaluations across LaMP, LongLaMP, PGraphRAG, and a self-constructed benchmark

demonstrate that TagPR achieves state-of-the-art performance. Ultimately, our work proves that explicit tag-guided process supervision is an effective strategy for eliciting genuine personalization reasoning in large language models.

6 Limitations & Potential Risks

Despite TagPR’s promising results, several limitations and risks remain. Our nine-tag vocabulary is derived from reasoning chains in LaMP and may therefore reflect benchmark-specific reasoning patterns. Although it generalizes in our experiments, its fixed scope may limit the model’s flexibility in handling highly complex or open-ended reasoning scenarios that require dynamic structure adaptation. The use of learnable user embeddings and deep personalization introduces potential privacy concerns regarding user data retention and the risk of reinforcing existing user biases or creating echo chambers, necessitating further research into privacy-preserving and alignment-focused personalization.

Acknowledgments

This work is supported by Meituan. This work is also supported by the Public Computing Cloud, Renmin University of China and by fund for building worldclass universities (disciplines) of Renmin University of China.

References

- Steven Au, Cameron Dimacali, Ojasmitha Pedirappagari, Namyong Park, Franck Dernoncourt, Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan A Rossi, and Nesreen K Ahmed. 2025. Personalized graph-based retrieval for large language models. In *Proceedings of the 39th Pacific Asia Conference on Language, Information and Computation*, pages 678–698.
- Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.
- Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.
- 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*.
- Song Jin, Shuqi Li, Shukun Zhang, and Rui Yan. 2026a. Finrpt: Dataset, evaluation system and llm-based multi-agent framework for equity research report generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 507–515.
- Song Jin, Juntian Zhang, Yuhao Liu, Xun Zhang, Yufei Zhang, Guojun Yin, Fei Jiang, Wei Lin, and Rui Yan. 2025. Beyond static testbeds: An interaction-centric agent simulation platform for dynamic recommender systems. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18914–18931.
- Song Jin, Juntian Zhang, Xun Zhang, Zeyang Tian, Fei Jiang, Guojun Yin, Wei Lin, Yong Liu, and Rui Yan. 2026b. Diningbench: A hierarchical multi-view benchmark for perception and reasoning in the dietary domain. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 35289–35310.
- Jieyong Kim, Tongyoung Kim, Soojin Yoon, Jaehyung Kim, and Dongha Lee. 2026. Rpm: Reasoning-level personalization for black-box large language models. In *International Conference on Learning Representations*, volume 2026, pages 84214–84251.
- Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, and 1 others. 2024. Longlamp: A benchmark for personalized long-form text generation. *arXiv preprint arXiv:2407.11016*.
- Cheng Li, Mingyang Zhang, Qiaozhu Mei, Yaqing Wang, Spurthi Amba Hombaiah, Yi Liang, and Michael Bendersky. 2023. Teach llms to personalize—an approach inspired by writing education. *arXiv preprint arXiv:2308.07968*.
- Jia-Nan Li, Jian Guan, Wei Wu, and Rui Yan. 2025. Extended inductive reasoning for personalized preference inference from behavioral signals. *arXiv preprint arXiv:2505.18071*.
- Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025. Llm+ persona-plug= personalized llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9373–9385.
- Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jia-cai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and 1 others. 2026. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. In *International Conference on Learning Representations*, volume 2026, pages 133805–133838.
- Sichun Luo, Guanzhi Deng, Jian Xu, Xiaojie Zhang, Hanxu Hou, and Linqi Song. 2025. Reasoning meets personalization: Unleashing the potential of large reasoning model for personalized generation. *arXiv preprint arXiv:2505.17571*.
- J MacQueen. 1967. Multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. University of California press Oakland, CA, USA.
- Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2024. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain*,

- Application, Group, or Individual (CustomNLP4U)*, pages 198–219.
- Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025. Measuring what makes you unique: Difference-aware user modeling for enhancing llm personalization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21258–21277.
- Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yuxin Chen, Cilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, Yoko Yamakata, and Tat-Seng Chua. 2026. Preference-aware rubric learning for personalized evaluation. *arXiv preprint arXiv:2605.31545*.
- Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174.
- Alireza Salemi, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 752–762.
- Alireza Salemi, Cheng Li, Mingyang Zhang, Qiaozhu Mei, Weize Kong, Tao Chen, Zhuowan Li, Michael Bendersky, and Hamed Zamani. 2025. Reasoning-enhanced self-training for long-form personalized text generation. *arXiv preprint arXiv:2501.04167*.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024b. 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.
- Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6459–6475.
- Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6476–6491.
- Bin Wu, Zhengyan Shi, Hossein A Rahmani, Varsha Ramineni, and Emine Yilmaz. 2024. Understanding the role of user profile in the personalization of large language models. *arXiv preprint arXiv:2406.17803*.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.
- Juntian Zhang, Song Jin, Chuanqi Cheng, Yuhan Liu, Yankai Lin, Xun Zhang, Fei Jiang, Guojun Yin, Wei Lin, Rui Yan, and 1 others. 2026. Viper: Empowering the self-evolution of visual perception abilities in vision-language models. In *International Conference on Learning Representations*, volume 2026, pages 98012–98039.
- Kepu Zhang, Teng Shi, Weijie Yu, and Jun Xu. 2025. Prlm: Learning explicit reasoning for personalized rag via contrastive reward optimization. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pages 5484–5488.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. [Group sequence policy optimization](#). *Preprint*, arXiv:2507.18071.
- Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. 2024. Hydra: Model factorization framework for black-box llm personalization. *Advances in Neural Information Processing Systems*, 37:100783–100815.

A Detailed Experimental Setup

This section provides a detailed description of our experimental setup, including implementation details, benchmark information, and baseline configurations.

A.1 Implementation Details

Backbone Model We use Qwen3-8B (Yang et al., 2025) as our base model for all experiments unless otherwise specified.

Supervised Fine-Tuning (SFT) The SFT stage was conducted on 8 A100 GPUs. We used a learning rate of $1e-5$ and a global batch size of 64. The model was trained for 2 epochs on the dataset described in Section 3.2.

Reinforcement Learning (RL) Data Sampling: We sampled data from the LaMP training set for RL. Specifically, we randomly sampled 1,024 examples for each of the LaMP-1, LaMP-3, LaMP-4, LaMP-5, and LaMP-7 tasks. For the more challenging LaMP-2 task, we sampled 3,200 examples. Training Parameters: The first RL stage was trained for 13 epochs, and the second stage was trained for 2 epochs. Both stages were conducted on 8 A100 GPUs with a global batch size of 128 and a learning rate of $1e-6$. Policy Rollout: During the policy rollout stage, we set the temperature to 1.0 and top-p to 1.0, generating 5 responses for each prompt. Other Hyperparameters: The low and high clip ratios for the GSPO algorithm were set to 0.0003 and 0.0004, respectively. For the repetition penalty reward, we used n-grams of size 4. For the tag reward, the minimum required number of tags was set to 3.

A.2 Benchmark Details

Dataset We use the LaMP benchmark, a widely-adopted benchmark for evaluating the personalization capabilities of LLMs. It requires models to analyze user historical profiles to answer current queries. Since the official test set is not publicly available, all our evaluations are conducted on the official validation set. LaMP-6 was excluded from our evaluation due to its unavailability. We evaluated on the complete validation dataset for all other tasks. The detailed data statistics of LaMP are shown in Table 4.

Evaluation Metrics Following the original LaMP benchmark, we employ the following metrics:

Classification Tasks (LaMP-1 & LaMP-2). Task Descriptions: LaMP-1 (Personalized Citation): A binary classification task where the model deter-

Task	Task Type	#Train	#Val	#Classes
LaMP-1	Binary classification	6,542	1,500	2
LaMP-2	Categorical classification	5,073	1,410	15
LaMP-3	Ordinal classification	20,000	2,500	5
LaMP-4	Text generation	12,500	1,500	-
LaMP-5	Text generation	14,682	1,500	-
LaMP-7	Text generation	13,437	1,498	-

Table 4: Data statistics of the LaMP benchmark.

mines which of two candidate references an author is likely to cite based on their profile. LaMP-2 (Personalized Movie Tagging): A multi-class classification task requiring the model to select the most appropriate tag from 15 candidates for a movie, conditioned on the user’s tagging history. Metrics & Calculation: We report Accuracy and F1-Score. For Accuracy, we calculate the ratio of correctly predicted samples to the total samples. For F1-Score: In LaMP-1 (binary), we report the standard F1 for the positive class. In LaMP-2 (multi-class), we follow the benchmark standard and report Macro-F1.

Rating Task (LaMP-3). Task Description: LaMP-3 (Personalized Product Rating) is an ordinal classification task. The model predicts a specific integer rating (1–5 stars) for a product based on the user’s historical reviews. Metrics: We employ MAE (Mean Absolute Error) and RMSE (Root Mean Square Error). These metrics are standard for quantifying the deviation between predicted scores and ground-truth user ratings.

Text Generation Tasks (LaMP-4, 5, 7). Task Descriptions: These tasks involve generating text tailored to a user’s specific style: LaMP-4: Personalized news headline generation. LaMP-5: Personalized scholarly title generation. LaMP-7: Personalized tweet paraphrasing. Metrics: We utilize ROUGE-1 and ROUGE-L. ROUGE-1: measures the overlap of unigrams to assess content adequacy. ROUGE-L: is based on the Longest Common Subsequence, evaluating fluency and structural similarity between the generated text and the reference.

A.3 Baselines and Comparison Setup

To rigorously evaluate our proposed method, we benchmark it against a wide spectrum of baselines. For a fair comparison, all methods are built upon the Qwen3-8B base model and utilize the user’s 8 most recent profiles as input, unless specified otherwise (e.g., proprietary models like GPT-4o).

The baselines are categorized as follows. Standard methodologies include Zero-shot, which gen-

erates responses without user profiles as a non-personalized lower bound; standard Retrieval-Augmented Generation (RAG); Personalization-Augmented Generation (PAG) (Richardson et al., 2023), which enhances RAG with user history summaries; Supervised Fine-Tuning (SFT) on the full dataset; and SFT-Ind, which is fine-tuned only on individual task data. Reasoning-enhanced variants of these methods, denoted with a ‘-R’ suffix, are also included. We further compare against advanced techniques. Personalization-focused methods include PPlug (Liu et al., 2025), a plug-and-play approach using specialized user embeddings, and HYDRA-Adapter (Zhuang et al., 2024), for which we use only its adapter version to maintain a consistent retrieval method for fairness. Reasoning-focused baselines include R2P (Luo et al., 2025), which employs hierarchical reasoning templates, and PrLM (Zhang et al., 2025), which uses a contrastive reward model with reinforcement learning. To situate our method’s performance against the frontier of language models, we also include several leading state-of-the-art LLMs: GPT-4o (Hurst et al., 2024), Gemini-2.5-Pro (Comanici et al., 2025), Qwen3-235B-A22B (Yang et al., 2025), and DeepSeek-R1 (Guo et al., 2025).

We deliberately exclude methods centered on optimizing the retrieval module, as improving retrieval is not the focus of our research. Additionally, we do not compare against OPPU (Tan et al., 2024b), as its approach requires fine-tuning a unique module for every user and presupposes the availability of extensive user-specific profiles, rendering it infeasible to implement across the full validation set.

B Additional Further Analysis

B.1 Model Training Cost

The additional training cost relative to SFT-R arises primarily from multi-stage RL optimization. As shown in Table 5, TagPR requires 169.7 A100 GPU-hours, compared with 177.7 A100 GPU-hours for the untagged RL baseline and 8.0 A100 GPU-hours for SFT-R. Thus, TagPR requires more computation than SFT alone but remains comparable in cost to an otherwise equivalent RL-based method.

B.2 PRMU Reward Model Evaluation

We evaluate the standalone preference prediction accuracy of PRMU variants on a held-out set. As

shown in Table 6, the full PRMU with user embeddings achieves the best accuracy (80.45%), substantially outperforming an off-the-shelf reward model (69.25%), confirming the value of both domain-specific fine-tuning and user-aware modeling.

B.3 Hyperparameter Sensitivity Analysis of Reward Function

We conducted a sensitivity analysis on the reward function hyperparameters by varying the weights of α , β , and γ while keeping all other experimental conditions constant. The results, presented in Table 7, indicate that the model remains robust within a reasonable range of parameter variations, with our current configuration yielding the most balanced performance.

B.4 Semantic Consistency of Generated Tags

To verify that generated tags reflect genuine semantic content rather than memorized syntactic templates, we randomly sampled 500 tag-content pairs from the model’s responses on the final test set. We employed Gemini-3-Flash-Preview as an independent judge with the following prompt:

Prompt for Semantic Consistency Evaluation

Judge whether the tag name and the content below are semantically consistent, meaning whether the content matches what the tag name expresses. Reply with exactly one word, consistent or inconsistent.

Tag name: {tag_name}

Content: {content}

Answer:

The evaluation yielded a 100% consistency rate. This strong semantic alignment indicates that the tags effectively externalize the model’s intermediate cognitive steps in a human-interpretable manner, rather than serving as empty structural placeholders.

B.5 Case Study

We present qualitative case studies to illustrate the advanced personalization reasoning of **TagPR**. The primary case is shown in Figure 1. Additional cases are presented below.

Method	SFT	PRMU Training	RL	Total Wall-Clock Time	Total A100 GPU-Hours
SFT-R	1 h $\times$ 8 A100	–	–	1 h	8.0
TagPR w/o Tagging	1 h $\times$ 8 A100	1 h 40 min $\times$ 1 A100	21 h $\times$ 8 A100	23 h 40 min	177.7
TagPR	1 h $\times$ 8 A100	1 h 40 min $\times$ 1 A100	20 h $\times$ 8 A100	22 h 40 min	169.7

Table 5: Training cost comparison. Total wall-clock time assumes sequential execution of SFT, PRMU training, and RL.

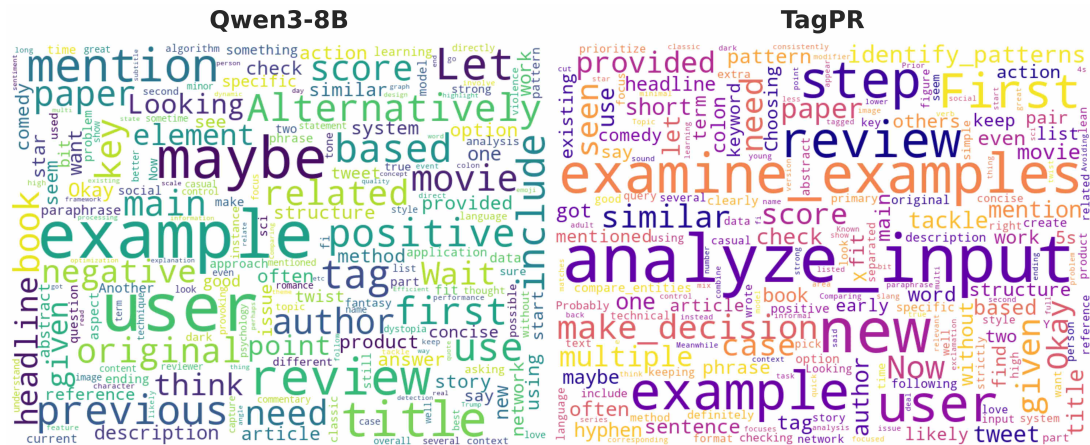


Figure 5: Word cloud comparison of reasoning chains from the baseline Qwen3-8B (left) and our TagPR model (right) on the LaMP validation set. TagPR’s reasoning is dominated by action-oriented keywords derived from our functional tags.

Method	Accuracy
Off-the-shelf Base RM	69.25%
Base RM fine-tuned on our dataset	75.40%
PRMU (full)	80.45%

Table 6: Preference prediction accuracy of PRMU variants on a held-out set.

Response from Qwen3-8B for Case 1

<think>

Okay, let's see. The user provided a list of papers by an author and is asking which of the two given references is related. The options are [1] "Compressed Linear Algebra for Large-Scale Machine Learning" and [2] "Database and information-retrieval methods for knowledge discovery".

First, I need to look at the titles of the author's papers to see which topic they're focusing on. The author's papers include things like Bayesian specification learning for API errors, distributed Bayesian machine learning, complex object implementations for big data, distributed algorithms for covariance matrices, grading in MOOCs,

learning to grade student programs, comparing machine learning platforms, simulating database-valued Markov chains, and a model-agnostic framework for spatial anomaly detection.

Looking at these, several themes emerge: ... Now, the two options are about compressed linear algebra for machine learning and database/information retrieval methods for knowledge discovery. ... Wait, the user's question is: "which reference is related?" given the list of the author's papers and the two options. So the user is asking which of the two options is related to the author's work.

So, the author's papers include topics like distributed Bayesian machine learning, big data systems, database simulations (SimSQL), and machine learning platforms. The first option, [1], is about compressed linear algebra for large-scale ML, which is a machine learning technique. The second option, [2], is about database and information retrieval methods for knowledge discovery. ... But the user is asking which of the two references is related. The answer is likely

Dataset →	LaMP-1		LaMP-2		LaMP-3		LaMP-4		LaMP-5		LaMP-7	
Method	ACC ↑	F1 ↑	ACC ↑	F1 ↑	MAE ↓	RMSE ↓	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑
$\alpha = 1.0, \beta = 1.0, \gamma = 1.0$	0.798	0.796	0.581	0.553	0.226	0.271	0.227	0.209	0.540	0.469	0.564	0.505
$\alpha = 0.2, \beta = 0.2, \gamma = 0.8$	0.791	0.788	0.573	0.533	0.229	0.272	0.226	0.208	0.533	0.459	0.561	0.501
$\alpha = 0.8, \beta = 0.2, \gamma = 0.2$	0.792	0.790	0.576	0.538	0.228	0.272	0.228	0.210	0.540	0.467	0.565	0.506
$\alpha = 0.2, \beta = 0.8, \gamma = 0.2$	0.787	0.786	0.574	0.536	0.234	0.274	0.225	0.207	0.535	0.460	0.560	0.499
$\alpha = 0.8, \beta = 0.8, \gamma = 0.4$	0.803	0.801	0.598	0.559	0.215	0.262	0.232	0.215	0.543	0.471	0.565	0.507
$\alpha = 0.8, \beta = 0.8, \gamma = 0.2$ (Ours)	0.805	0.804	0.599	0.558	0.218	0.263	0.234	0.214	0.542	0.472	0.566	0.506

Table 7: Hyperparameter sensitivity analysis on a held-out validation set (500 instances per task, sampled from unused training data). Our default configuration achieves optimal or near-optimal performance. Best results in **bold**.

[2], but I’m not 100% certain. Alternatively, maybe [1] is related because the author’s work on large-scale machine learning (like the paper on distributed algorithms) could be connected to compressed linear algebra. But the user’s options are two references, and the answer is to choose one. Given that the author has a paper on SimSQL (database-related), [2] is about databases and information retrieval, so [2] is more directly related. Therefore, the answer is [2].
</think>
[2]

Response from TagPR for Case 1

<think>
<analyze_input>
Okay, let me try to figure out which reference is related to the author’s papers. The user provided a list of papers the author has written and two options for references: [1] “Compressed Linear Algebra for Large-Scale Machine Learning” and [2] “Database and information-retrieval methods for knowledge discovery”.
</analyze_input>
<examine_examples>
First, I need to look at the titles of the author’s papers to see what themes they’re working on. Let me list them out again:
1. “Bayesian specification learning for finding API usage errors”
...
9. “A Model-Agnostic Framework for Fast Spatial Anomaly Detection”
</examine_examples>
<identify_patterns>
Looking at these, several themes emerge. There’s a strong focus on machine learn-

ing, especially distributed and large-scale aspects. Papers like “The BUDS Language for Distributed Bayesian Machine Learning” and “A comparison of platforms for implementing and running very large scale machine learning algorithms” clearly relate to machine learning systems. Also, “Distributed Algorithms for Computing Very Large Thresholded Covariance Matrices” and “Compressed Linear Algebra for Large-Scale Machine Learning” (which is reference [1]) might be connected because they both deal with large-scale computations in machine learning.

</identify_patterns>
<evaluate_reference>
Then there’s the database-related paper: ...
</evaluate_reference>
<compare_entities>
Now, comparing the two references: [1] is about compressed linear algebra for machine learning, which aligns with papers on large-scale ML algorithms and distributed computing. [2] is about database and information-retrieval methods for knowledge discovery, which connects to the database-valued Markov chains paper.
</compare_entities>
<synthesize_findings>
But the author has multiple papers on machine learning, especially distributed and large-scale. ...
</synthesize_findings>
<make_decision>
So, reference [1] seems more broadly related to the author’s work in machine learning systems and large-scale algorithms. Reference [2] is only tangentially related through one paper, while [1] connects to multiple papers.

</make_decision>
</think>
[1]

Thanks, @Ylsoccermomof3, and I'll savor my double cheeseburger.

Response from Qwen3-8B for Case 2

<think>
Okay, let's see. The user wants me to paraphrase the tweet "@Ylsoccermomof3, I appreciate it and I'll enjoy my double cheeseburger." without any explanation before or after.
First, I need to understand the original message. The person is thanking @Ylsoccermomof3 and mentioning they'll enjoy a double cheeseburger. The key elements are gratitude and the mention of the food.
Now, paraphrasing means rewording it while keeping the same meaning. Let me think of synonyms. "Appreciate" could be "thank" or "grateful for." "Enjoy" might be "savor" or "look forward to." "Double cheeseburger" is specific, so maybe keep it as is or use "big cheeseburger" for a slight variation.
Also, the structure: the original starts with the handle, then the message. The user's example response uses "Thanks, @Ylsoccermomof3, and I'll savor my double cheeseburger." So they changed "appreciate" to "Thanks" and "enjoy" to "savor." That's a good approach.
I should check if there are other ways to phrase it. Maybe "I'm grateful, @Ylsoccermomof3, and I'll relish my double cheeseburger." But "relish" might be a bit more formal. The original is casual, so "savor" fits better.
Another option: "Thanks, @Ylsoccermomof3, and I'll enjoy my big cheeseburger." But "big" might not be the same as "double." The user's example used "double," so maybe stick with that.
I think the user's example is good, but maybe check for any other possible variations. However, the user wants a paraphrase, not necessarily the same as the example. Let me make sure the structure is correct. The handle is included, then the message.
...

Response from TagPR for Case 2

<think>
<analyze_input>
Okay, the user has given me a task to paraphrase a specific tweet based on patterns observed from several example tweets they provided. Let me break this down carefully.
</analyze_input>
<examine_examples>
First, I need to analyze the patterns in the example tweets they shared. Looking at these:
- They all start with "@username" followed by a casual greeting or statement
- The language is very conversational with contractions ("you're", "it's")
- There's a mix of personal updates ("my cat is having kittens") and light-hearted interactions
- Some include multiple @mentions
- The tone is friendly and informal
</examine_examples>
<identify_patterns>
The key pattern I notice is the casual, conversational style with:
- Use of "I" statements
- Simple sentence structures
- No formal punctuation beyond basic commas
- Often includes personal context ("I just woke up", "I think everyone has")
</identify_patterns>
<analyze_input>
Now for the tweet to paraphrase: "@Ylsoccermomof3, I appreciate it and I'll enjoy my double cheeseburger."
</analyze_input>
<identify_patterns>
Hmm... the original has:
- Formal "I'll" instead of "I'm gonna"
- "double cheeseburger" which is slightly more specific than typical casual phrasing
- No contractions in "I'll" (though it's common in speech)
</identify_patterns>

...

<verify_conclusion>

So "I'm gonna" is better than "I'll" for that casual vibe. And "enjoy" could become "devour" like in the @Jintanut example ("very hungry") but "enjoy" is already good.

</verify_conclusion>

<formulate_conclusion>

Final decision: Keep the structure but make it sound like a real person talking. Add "thanks" to match the gratitude pattern, use "gonna" for casualness, and keep the @mention.

</formulate_conclusion>

</think>

@Ylsoccermomof3, thanks for the encouragement and I'm gonna enjoy my double cheeseburger.

C Supplement for Generalization Assessment

C.1 Generalization Assessment on LongLaMP

We further evaluated the generalization capabilities of our model using the LongLaMP dataset. Characterized by extensive user history profiles and a requirement for long-form text generation, LongLaMP poses a significant challenge to personalized reasoning. We focused on three tasks: TopicWriting, ProductReview, and AbstractReviewTitle. Due to computational constraints, we randomly sampled 10% of the test set for each task and reported performance using ROUGE-1 (R-1) and ROUGE-L (R-L).

We assessed zero-shot generalization by directly applying the model trained on the LaMP dataset to the LongLaMP dataset. As shown in Table 8, our model demonstrated robust zero-shot generalization capabilities.

C.2 Generalization Assessment on PGraphRAG

We further evaluated the model's generalization capabilities using the PGraphRAG dataset. This dataset is characterized by extreme sparsity, where the majority of user profiles contain either zero or only one, posing a significant challenge for personalized reasoning. We focused specifically on the AmazonReviewTitle subtask. Since historical

profiles are the foundation of personalization, we restricted our evaluation to test users with a profile history length of at least one, excluding neighbor profiles information. By directly applying our trained model, we demonstrated its robust personalized reasoning capabilities even in data-sparse scenarios, as shown in Table 8.

D Supplement for Robustness Assessment

This section provides supplementary results for the robustness assessment discussed in the main paper. Figure 7 and Figure 8 illustrate the performance of **TagPR** against the SFT and Base baselines on the LaMP-1, LaMP-3, LaMP-5, and LaMP-7 tasks, complementing the results for LaMP-2 and LaMP-4 shown in Figure 6.

As demonstrated in the figure, the conclusions from the main text hold true across these additional datasets. **TagPR** consistently achieves superior performance, showcasing high data efficiency by reaching a strong performance level with only a few user interactions. Furthermore, its advantage is maintained across all profile retrieval methods, including random selection, which underscores the robustness of our framework.

E Supplement for Tagged Reasoning Chains Construction

E.1 Prompts Details

E.1.1 The Prompt for LLM Filter

Prompt for LLM Filter

Role

You are an AI expert specializing in evaluating the Chain-of-Thought (CoT) quality of large language models. Your task is to provide a comprehensive and objective evaluation of the model's Chain-of-Thought quality based on the provided question and the model's response.

Task Description

I will provide you with a "Question" and a "Model Response" generated by a large language model for that question. The "Model Response" contains detailed reasoning steps and the final answer. Please evaluate the quality of the Chain-of-Thought in this "Model Response" according to the following evaluation dimensions, and strictly

Dataset →	TopicWriting		ProductReview		AbstractGeneration		AmazonReviewTitle	
Method	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑	R-1 ↑	R-L ↑
Qwen3-8B-Instruct	0.282	0.134	0.342	0.154	0.382	0.203	0.172	0.163
Qwen3-8B-Thinking (Base)	0.271	0.124	0.321	0.150	0.351	0.182	0.179	0.165
Qwen3-32B-Instruct	0.292	0.134	0.354	0.159	0.384	0.202	0.171	0.161
Qwen3-32B-Thinking	0.275	0.119	0.320	0.144	0.346	0.177	0.198	0.191
GPT-4o	0.294	0.140	0.330	0.157	0.372	0.200	0.140	0.136
Gemini-2.5-Pro	0.270	0.132	0.372	0.171	0.398	0.215	0.204	0.196
TagPR (Our)	0.303	0.154	0.377	0.175	0.404	0.221	0.210	0.202

Table 8: Zero-shot generalization performance on partial test sets of LongLaMP and PGraphRAG. We report ROUGE-1 (R-1) and ROUGE-L (R-L) scores. The best results are in **bold**. Our TagPR demonstrates superior performance.

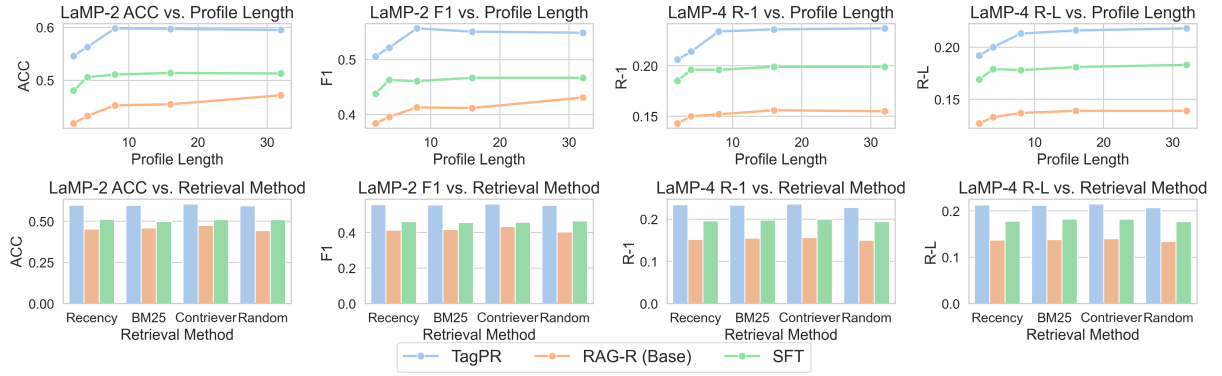


Figure 6: Robustness assessment of TagPR on LaMP-2 and LaMP-4. **Top:** Performance across varying profile lengths. **Bottom:** Performance across different retrieval methods. TagPR consistently outperforms baselines, demonstrating high data efficiency and resilience to retrieval quality.

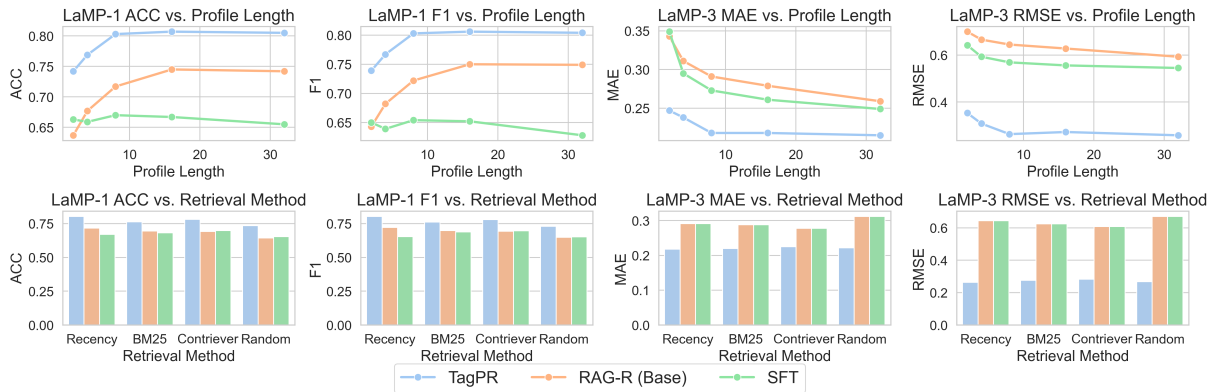


Figure 7: Robustness assessment of TagPR on LaMP-1 and LaMP-3. **Top:** Performance across varying profile lengths. **Bottom:** Performance across different retrieval methods.

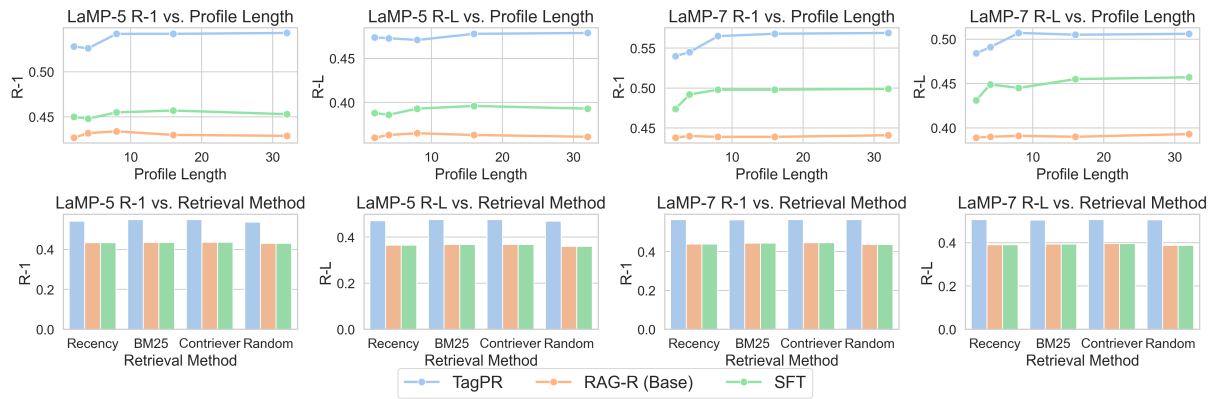


Figure 8: Robustness assessment of TagPR on LaMP-5 and LaMP-7. **Top:** Performance across varying profile lengths. **Bottom:** Performance across different retrieval methods.

output the result in the specified JSON format.

Evaluation Dimensions

1. **Logical Coherence:** Is there a clear logical connection between the reasoning steps? Are there any logical leaps or contradictions? (1-5 points)
2. **Step Accuracy:** Is every step in the reasoning chain accurate? Are there any factual errors or calculation mistakes? (1-5 points)
3. **Reasoning Completeness:** Does the Chain-of-Thought cover all the key steps required to solve the problem? Are there any omissions? (1-5 points)
4. **Relevance to the Question:** Is the entire thought process closely centered around the original question? Is there any redundant or off-topic reasoning? (1-5 points)

Input Data

[Question]

{question}

[Model Response]

{response}

Output Format

Please strictly follow the JSON format below for your evaluation results. Ensure the

output is a complete and syntactically correct JSON object. Do not add any additional explanations or text before or after the JSON code block.

```
{
  "evaluation_report": {
    "detailed_assessment": [
      {
        "dimension": "Logical Coherence",
        "reasoning": "[Provide an explanation of the pros and cons for this dimension]",
        "score": "[Enter an integer score from 1-5 here]"
      },
      {
        "dimension": "Step Accuracy",
        "reasoning": "[Provide an explanation of the pros and cons for this dimension, and explicitly point out any errors if they exist]",
        "score": "[Enter an integer score from 1-5 here]"
      },
      {
        "dimension": "Reasoning Completeness",
        "reasoning": "[Provide an explanation of the pros and cons for this dimension, and explicitly point out any omissions if they exist]",
        "score": "[Enter an integer score from 1-5 here]"
      }
    ]
  }
}
```

```

{
  "dimension": "Relevance to
    the Question",
  "reasoning": "[Provide an
    explanation of the pros
    and cons for this
    dimension, such as the
    presence of redundant
    information]",
  "score": "[Enter an
    integer score from 1-5
    here]"
}
],
"summary": {
  "total_score": "[Enter the
    total score, between 1
    and 20, which is the sum
    of the scores from each
    dimension]"
}
}
}

```

E.1.2 The Prompt for Exploratory Tagging

Prompt for Exploratory Tagging

Role: You are an expert specializing in understanding and analyzing the thought processes of AI. Your task is to carefully review a given question and the “Chain-of-Thought” generated by an AI model to answer it. You will then break down this Chain-of-Thought into meaningful segments and assign an XML-style tag to each segment that best describes its function.

Task:

Based on the user-provided [Question] and the model-generated [Chain-of-Thought], please complete the following steps:

1. **Analyze the Question and Chain-of-Thought:** Deeply understand the core requirements of the question and how the Chain-of-Thought progressively derives the final answer.
2. **Segment the Chain-of-Thought:** Break down the entire Chain-of-Thought into multiple logically coherent steps or stages. Each step should represent a distinct function in the thought process.
3. **Tag Each Segment:** For each segmented piece, create an XML-style tag

(e.g., <tag_name>...</tag_name>). The tag name should accurately summarize the role this segment plays in the overall thought process.

4. **Output the Result:** Output only the tagged Chain-of-Thought text.

Guidelines and Tag Suggestions:

- **Tags should reflect function:** The tag name should describe “what this segment is doing”, not simply restate its content.
- **Maintain consistency:** Try to use the same tags for similar functions.
- **Tags should be concise and clear:** Use short, easy-to-understand words for tag names (English is recommended for easier programmatic processing).

Example:

[Question]:

History: “When brilliant video game maker Flynn hacks the mainframe of his ex-employer. . .” → Tag: “classic” and, “When an insatiable great white shark terrorizes the townspeople. . .” → Tag: “classic” and, “Jake Blues, just released from prison, puts his old band back together. . .” → Tag: “classic” and, . . . (other examples of “classic” and “true story”). . .

Description: “In the years before the Second World War, a tomboyish postulant at an Austrian abbey is hired as a governess in the home of a widowed naval captain with seven children, and brings a new love of life and music into the home.”

Candidate Tags: [sci-fi, based on a book, comedy, action, . . . , classic, . . . , true story]

[Chain-of-Thought to be tagged]:

First, I analyzed the historical data provided by the user. This data shows descriptions of several movies and their corresponding tags.

I found that the movies tagged as “classic” (like *Tron*, *Jaws*, *The Blues Brothers*, etc.)

are all well-known and beloved films released in the 20th century. Movies tagged as “true story” (like *Apollo 13*) are based on real events. This reveals a pattern: the “classic” tag points to older films with a significant era-defining and broad cultural impact. Next, I analyzed the new movie description to be tagged: “In the years before the Second World War...”. This description points to the famous movie *The Sound of Music*. *The Sound of Music*, released in 1965, tells a household story and achieved immense success and cultural influence worldwide, perfectly matching the characteristics of a “classic” movie that I summarized from the historical data. It is not a sci-fi film, and while it might be partially based on real experiences, its core identity is its classic status.

Therefore, among all candidate tags, “classic” is the most appropriate.

[Your Output]:

```
<analyze_user_history>
First, I analyzed the historical
  data provided by the user. This
  data shows descriptions of
  several movies and their
  corresponding tags.
</analyze_user_history>
<identify_pattern>
I found that the movies tagged as
  "classic" (like *Tron*, *Jaws*,
  *The Blues Brothers*, etc.)
  are all well-known and beloved
  films released in the 20th
  century. Movies tagged as "true
  story" (like *Apollo 13*) are
  based on real events. This
  reveals a pattern: the "classic
  " tag points to older films
  with a significant era-defining
  and broad cultural impact.
</identify_pattern>
<analyze_current_query>
Next, I analyzed the new movie
  description to be tagged: "In
  the years before the Second
  World War...". This description
  points to the famous movie *
  The Sound of Music*.
</analyze_current_query>
<compare_query_with_history>
*The Sound of Music*, released in
  1965, tells a household story
  and achieved immense success
  and cultural influence
  worldwide, perfectly matching
  the characteristics of a "
  classic" movie that I
```

```
summarized from the historical
data. It is not a sci-fi film,
and while it might be partially
  based on real experiences, its
  core identity is its classic
  status.
```

```
</compare_query_with_history>
<final_conclusion>
```

```
Therefore, among all candidate
  tags, "classic" is the most
  appropriate.
```

```
</final_conclusion>
```

Now, according to the rules above, please add tags to the [Question] and [Chain-of-Thought] provided below:

[Question]:

```
{question}
```

[Chain-of-Thought to be tagged]:

```
{chain_of_thought}
```

E.1.3 The Prompt for Restricted Tagging

Prompt for Restricted Tagging

Role: You are an expert specializing in understanding and analyzing the thought processes of AI. Your task is to carefully review a given question and the “Chain-of-Thought” generated by an AI model to answer it. You will then break down this Chain-of-Thought into meaningful segments and assign an XML-style tag to each segment that best describes its function.

Task:

Based on the user-provided [Question] and the model-generated [Chain-of-Thought], please complete the following steps:

1. **Analyze the Question and Chain-of-Thought:** Deeply understand the core requirements of the question and how the Chain-of-Thought progressively derives the final answer.
2. **Segment the Chain-of-Thought:** Break down the entire Chain-of-Thought into multiple logically coherent steps or stages. Each step should represent a distinct function in the thought process.
3. **Tag Each Segment:** For each seg-

mented piece, create an XML-style tag (e.g., `<tag_name>...</tag_name>`). The tag name must be chosen exclusively from the mandatory list provided below.

4. **Output the Result:** Output only the tagged Chain-of-Thought text.

Mandatory Tag Set and Definitions:

You **must** use **only** the tags from the following list. Choose the tag that best describes the function of each segment.

- **analyze_input:** Analyzes the initial user question or task description to understand the goal.
- **examine_examples:** Examines specific, individual pieces of evidence, data points, or examples provided.
- **identify_patterns:** Summarizes findings from one or more examples to find a common rule, pattern, or theme.
- **evaluate_reference:** Assesses how the input aligns with a specific, external piece of reference material.
- **compare_entities:** Performs a direct comparison between two or more items to determine their similarities, differences, or which is superior.
- **synthesize_findings:** Consolidates all prior analysis and comparisons into a comprehensive summary before making a final choice.
- **make_decision:** Commits to a specific, final choice or action.
- **verify_conclusion:** Performs a final check on the decision to ensure it is logical, consistent, and accurate.
- **formulate_conclusion:** Constructs the final, complete answer or statement based on the decision made.

Example:
[Question]:

History: "When brilliant video game maker Flynn hacks the mainframe of his ex-employer..." → Tag: "classic" and, "When an insatiable great white shark terrorizes the townspeople..." → Tag: "classic" and, "Jake Blues, just released from prison, puts his old band back together..." → Tag: "classic" and, ... (other examples of "classic" and "true story")...

Description: "In the years before the Second World War, a tomboyish postulant at an Austrian abbey is hired as a governess in the home of a widowed naval captain with seven children, and brings a new love of life and music into the home."

Candidate Tags: [sci-fi, based on a book, comedy, action, ..., classic, ..., true story]

[Chain-of-Thought to be tagged]:

First, I analyzed the historical data provided by the user. This data shows descriptions of several movies and their corresponding tags.

I found that the movies tagged as "classic" (like *Tron*, *Jaws*, *The Blues Brothers*, etc.) are all well-known and beloved films released in the 20th century. Movies tagged as "true story" (like *Apollo 13*) are based on real events. This reveals a pattern: the "classic" tag points to older films with a significant era-defining and broad cultural impact. Next, I analyzed the new movie description to be tagged: "In the years before the Second World War...". This description points to the famous movie *The Sound of Music*. *The Sound of Music*, released in 1965, tells a household story and achieved immense success and cultural influence worldwide, perfectly matching the characteristics of a "classic" movie that I summarized from the historical data. It is not a sci-fi film, and while it might be partially based on real experiences, its core identity is its classic status.

Therefore, among all candidate tags, "classic" is the most appropriate.

[Your Output]:

```
<examine_examples>
First, I analyzed the historical
data provided by the user. This
data shows descriptions of
```

```

several movies and their
corresponding tags.
</examine_examples>
<identify_patterns>
I found that the movies tagged as
"classic" (like *Tron*, *Jaws*,
*The Blues Brothers*, etc.)
are all well-known and beloved
films released in the 20th
century. Movies tagged as "true
story" (like *Apollo 13*) are
based on real events. This
reveals a pattern: the "classic
" tag points to older films
with a significant era-defining
and broad cultural impact.
</identify_patterns>
<analyze_input>
Next, I analyzed the new movie
description to be tagged: "In
the years before the Second
World War...". This description
points to the famous movie *
The Sound of Music*.
</analyze_input>
<compare_entities>
*The Sound of Music*, released in
1965, tells a household story
and achieved immense success
and cultural influence
worldwide, perfectly matching
the characteristics of a "
classic" movie that I
summarized from the historical
data. It is not a sci-fi film,
and while it might be partially
based on real experiences, its
core identity is its classic
status.
</compare_entities>
<make_decision>
Therefore, among all candidate
tags, "classic" is the most
appropriate.
</make_decision>

```

Now, according to the rules above, please add tags to the [Question] and [Chain-of-Thought] provided below:

[Question]:

{question}

[Chain-of-Thought to be tagged]:

{chain_of_thought}

E.2 Refined Primary Tags Set

The final set of primary tags derived from our clustering procedure is listed in Figure 9. These tags were used to annotate the reasoning chains in our dataset. To determine the number of primary tags (K) during the clustering phase, we analyzed the Within-Cluster Sum of Squares (WCSS) and se-

lected $K = 9$ to balance cluster compactness with diminishing returns.

F New Constructed Personalization Benchmark

To evaluate zero-shot, cross-lingual generalization, we built a benchmark from Dianping, a prominent Chinese user-generated content platform. This appendix details its construction.

F.1 Data and User Profile Creation

We collected public posts from Dianping and applied rigorous filtering to retain high-quality content, removing short posts, duplicates, and advertisements. From this cleaned dataset, we selected 1,000 users with extensive post histories.

For each user, a profile representing their personal writing style was constructed from their 8 most recent posts (title and content). The 9th most recent post was held out as the ground truth for our evaluation tasks, ensuring a strict zero-shot setting where the test data is unseen.

F.2 Task Formulation

The benchmark consists of three distinct tasks, with one instance per user for each task, totaling 3,000 evaluation instances. All tasks are conditioned on the user’s 8-post profile. As in the LaMP dataset, we use the ROUGE-1 and ROUGE-L metrics for evaluation.

Dianping-Content (Title → Content): Given the title of the held-out post, the model must generate the full post content in the user’s specific style.

Dianping-Title (Content → Title): The inverse task, where the model generates a stylistically appropriate title from the held-out post’s content.

Dianping-Paraph (Generic → Stylized Post): This task measures stylistic transfer. For each user’s held-out post, we first used a general-purpose LLM (GPT-4o) to generate a neutral, generic version based on the original content. The model’s task is to rewrite this generic text to match the user’s unique style, with the user’s original post as the target.

F.3 Benchmark Statistics

Key statistics of the final benchmark are summarized in Table 9.

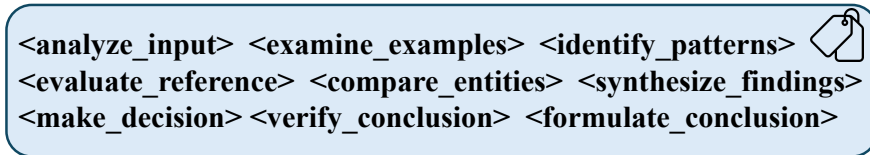


Figure 9: The refined set of nine primary tags used for annotating reasoning chains. These tags represent the most salient reasoning patterns identified through our clustering analysis.

Task	Task Type	#Test	#Classes
Dianping-Content	Text generation	1000	-
Dianping-Title	Text generation	1000	-
Dianping-Paraph	Text generation	1000	-

Table 9: Data statistics of the new constructed personalization benchmark.