Optimizing User Profiles via Contextual Bandits for Retrieval-Augmented LLM Personalization

Anonymous ACL submission

Abstract

Large Language Models (LLMs) excel at general-purpose tasks, but personalizing their responses to individual users remains challenging. Retrieval augmentation offers a lightweight alternative to fine-tuning by conditioning LLMs on user history records, yet existing strategies rely on heuristics (e.g., relevance to the query) that overlook the true contribution of records to personalization. To address these limitations, we propose PURPLE, a contextual bandit framework that oPtimizes UseR Profiles for Llm pErsonalization. PURPLE operates as a re-ranking layer over candidate records, balancing efficiency with personalization quality. Across nine real-world personalization tasks spanning classification, regression, and short- and long-text generation, PURPLE consistently outperforms strong heuristic and retrieval-augmented baselines, establishing contextual bandit retrieval as a principled and scalable solution for personalized LLMs. Our code is available at: https://anonymous.4open.science/r/ACL-2026-PURPLE-3096/.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable success in various natural language processing tasks. As these models are increasingly applied to personalized applications, tailoring responses to individuals based on their own preferences has become a crucial challenge. Existing approaches for personalizing LLMs, such as Parameter-Efficient Fine-Tuning (PEFT) (Hu et al., 2022) and Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022), generally require modifying model parameters. These approaches incur high computational costs, demand frequent updates, and are impractical for real-time personalization at scale, especially when the LLM is not fully open-sourced. Moreover, continually fine-tuning models for different individuals would complicate safety evaluation and deployment, since each personalized variant would require separate testing.

In this paper, we study a lightweight approach to LLM personalization via retrieval augmentation (Wu et al., 2025), where user profiles are formed by retrieving and injecting past user records into the prompt. Prior work has shown that incorporating such profiles can effectively steer LLM outputs toward individual preferences (Salemi et al., 2024; Jiang et al., 2025). Compared to parameter-updating methods, retrieval-augmented personalization is lightweight, transparent, and readily deployable, as users can directly inspect and edit the records guiding generation. However, simply using the full user history risks introducing redundancy and noise and may exceed the model's context window, while overly aggressive pruning may discard important personalization signals. This tension raises a central challenge: ***which* user records should be used to construct the user profile** so as to maximize *utility*, i.e., improve downstream task performance when conditioned on by the LLM?

Existing strategies for building user profiles often rely on heuristics, such as selecting records with the highest *relevance*, defined as semantic similarity to the query (Karpukhin et al., 2020). However, we argue that relevance is an unreliable proxy for personalization utility. This is because relevance-based heuristics conflate surface-level contextual similarity with the underlying preference signals that actually drive personalization. As a result, relevance scores emphasize how similar a record *looks* to the query not necessarily whether it supports the user's current intent. To illustrate this dichotomy, consider Figure 1, where the user seeks *"relaxing movies for Friday night"*. Standard relevance heuristics prioritize record h_1 due to the high lexical overlap of *"Friday night"*, ig-

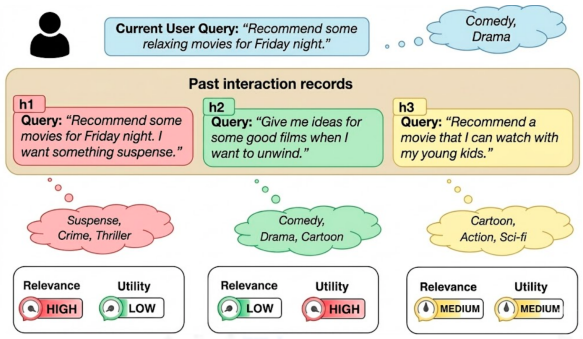

Figure 1: An illustration of the discrepancy between relevance and utility for a user's past interaction records given a query. The thought bubbles illustrate the potential movie genres that satisfy the underlying user intent for each query.

noring the latent preference for *"suspense"* which contradicts the current need for relaxation. Conversely, record h_2 offers high utility by capturing the semantic intent to *"unwind"*, despite lacking surface-level keyword matches. This highlights that relevance scores prioritize how similar a record *looks* to the query, whereas utility depends on whether it supports the underlying user intent.

Beyond evaluating individual records, constructing an effective user profile requires set-aware reasoning rather than independent selection. Even if the utility of each record could be accurately estimated in isolation, a greedy top-k selection remains suboptimal. This is because record utilities do not compose additively; the contribution of a specific record depends heavily on the context of other retrieved records. Revisiting Figure 1, while h_2 and h_3 individually appear useful by reflecting a shared preference for lighthearted content, their combination is suboptimal. Jointly prompting with $\{h_2, h_3\}$ introduces conflicting constraints—mixing *"comedy"* preferences with *"young kids"* requirements—which dilutes generation quality compared to using h_2 alone. This demonstrates that personalization utility is non-monotonic: even valid preference signals can interfere with one another, meaning that simply accumulating "useful" records can actively degrade profile performance.

The limitations highlighted above call for a reranking mechanism that is directly optimized for downstream generation performance and is explicitly aware of interactions among records. Existing approaches fall short of these requirements: heuristic retrieval relies on static proxies like similarity, while recent listwise rerankers, though capable of modeling dependencies, remain constrained by relevance-oriented supervision. To bridge this gap, we propose **PURPLE**, a framework that models user record selection as a contextual bandit problem (Langford and Zhang, 2007). In its formulation, the context consists of both the current query and the user's past records. The selection policy is guided by a reward function reflecting downstream personalized text generation performance. PURPLE outputs a *propensity score* for each user record, which is passed through a Plackett-Luce ranking model to produce the final selected user records. This formulation enables the model to capture interactions between records and adaptively select those that are most beneficial. We train PURPLE end-to-end using the policy gradient method (Sutton et al., 1999).

Our main contributions are as follows:

- We introduce PURPLE, a framework that casts retrieval-augmented LLM personalization as a contextual bandit problem, adaptively optimizing user profiles beyond static heuristics.
- We show through extensive experiments on nine personalization tasks, covering classification, regression, and short- and long-text generation, that PURPLE consistently outperforms strong baselines in both effectiveness and efficiency.
- We perform comprehensive ablation studies on PURPLE's key design choices and further validate PURPLE's effectiveness through extended analysis and human evaluation.

2 Related work

LLMs for Personalization. LLMs demonstrate strong performance across domains (OpenAI, 2024), yet their outputs often diverge from user expectations because pre-training captures general rather than individual needs. Reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) and parameter-efficient finetuning (PEFT) (e.g., LoRA (Hu et al., 2022)) can align models with user preferences, but both require model finetuning and are impractical for end users who lack access to resources. A complementary direction personalizes LLMs through user profiles (Salemi et al., 2024), built from prior user interactions or external signals. Leveraging user profiles has shown benefits across multiple tasks requiring personalization, such as recommendation (Penaloza

et al., 2025), text summarization (Zhang et al., 2024), question answering (Wu et al., 2024), content generation (Shen et al., 2024), and personalized chatbot interaction (Jiang et al., 2025). Yet it remains unclear which user history records in a profile truly drive performance improvements, particularly in retrieval-augmented generation (RAG), where performance hinges on selecting semantically relevant context. Moreover, little analysis has been conducted on how to best select and compose user records into profiles with high personalization utility. Our work addresses this gap by studying how user profiles shape personalization in retrieval-augmented LLMs, and by proposing strategies for selecting user records to maximize downstream performance.

Retrieval-Augmented Language Models. Retrieval-augmented language models (RALMs) enhance parametric LMs with external memory to improve factuality and coverage. Early work such as REALM (Guu et al., 2020) and RAG (Lewis et al., 2020) jointly trained the retriever and LM, while Re2G (Glass et al., 2022) further incorporated a reranking module into this end-to-end pipeline. To reduce training costs, subsequent methods froze the LM and applied retrieval in-context. For example, In-Context RALM (Ram et al., 2023) leveraged LLMs for reranking, while REPLUG (Shi et al., 2024) distilled retrievers from LLMs. More recently, instruction-tuned variants such as SelfRAG (Asai et al., 2024) and RankRAG (Yu et al., 2024) jointly model retrieval and generation, but their reliance on large-scale finetuning renders them impractical for personalization.

The methods most relevant to our work are In-Context RALM and REPLUG; however, both incorporate only one retrieved record at a time, a limitation our method directly addresses. Specifically, REPLUG combines multiple records by weighting generation outputs with retrieval probabilities, while In-Context RALM periodically triggers retrieval during decoding at fixed steps and replaces previously used records. These designs arise because jointly reasoning over multiple records leads to a combinatorial explosion in the number of possible profiles. In contrast, our approach is explicitly designed to overcome this limitation by modeling cross-record dependencies and directly optimizing over multi-record profiles without resorting to such approximations.

LLMs for Reranking. Reranking methods can be categorized as pointwise, pairwise, or listwise. Pointwise models such as MonoBERT (Nogueira et al., 2019) and MonoT5 (Nogueira et al., 2020) score each query–document pair independently, while pairwise models such as DuoT5 (Pradeep et al., 2021) compare candidates in pairs. In contrast, listwise approaches jointly model the full candidate set and have recently been advanced by LLMs through prompt-only ranking (RankGPT (Sun et al., 2023)), model distillation (e.g., RankVicuna, RankZephyr, Lit5Distill, FIRST (Pradeep et al., 2023a,b; Tamber et al., 2023; Gangi Reddy et al., 2024)), and inference-time relevance extraction (ICR (Chen et al., 2025)). However, these methods conflate relevance with utility, which is not ideal for personalization. In this work, we instead train rerankers using downstream generation quality as feedback, prioritizing utility over semantic similarity.

3 Methodology

We formulate retrieval-augmented LLM personalization as a contextual bandit problem (Langford and Zhang, 2007), where the goal is to learn a policy that selects informative user records. Unlike classic multi-armed bandits, contextual bandits incorporate auxiliary information (e.g., the current query and user history) before making a selection, enabling direct optimization of retrieval strategies through policy gradient reinforcement learning. This aligns the selection of user records with downstream personalization objectives.

3.1 Problem Formulation

We consider a dataset $\mathcal{D} = \{(\mathcal{H}^u, x^u, y^u)\}_{u=1}^{|\mathcal{D}|}$, where each example consists of a user's collection of history records $\mathcal{H}^u$, a query x^u, and a ground-truth personalized response y^u. Personalization is achieved by retrieving informative records from $\mathcal{H}^u$ and supplying them as context to a frozen LLM, which then generates the final response. In practice, we apply PURPLE as a re-ranking module on top of a candidate pool selected by lightweight heuristics, ensuring low-latency inference compatible with large-scale systems. In the following development, we focus on a single user and omit the superscript index for brevity.

Let $\mathcal{H} = \{h_1, \ldots, h_N\}$ denote the set of N history records for a user, where each record $h_i = (x_i, y_i)$ is an input–output pair (e.g., a query and

its answer from the user). Given a new query x, our goal is to construct a user profile from $\mathcal{H}$ to condition the LLM for generating a personalized response. Formally, a profile is an ordered tuple $\mathcal{P} = \langle p_1, \ldots, p_K \rangle \in \mathrm{Perm}_K(\mathcal{H})$, which is a K-permutation of $\mathcal{H}$. We stress that the profile is order-sensitive: different permutations of the same K records correspond to distinct profiles and thus provide different inputs to the downstream LLM.

We formulate the selection of $\mathcal{P}$ as a *contextual bandit problem*, where the context is given by the user's history $\mathcal{H}$ and the query x, and the action corresponds to selecting K records from $\mathcal{H}$ to construct a profile. Formally, this formulation consists of the following key components:

- **Context:** $\mathcal{C} = (\mathcal{H}, x)$, where $\mathcal{H}$ is the user's collection of history records and x is the query. This representation captures both past user preferences and the immediate intent.
- **Actions:** $\mathcal{P} = \langle p_1, \ldots, p_K \rangle \in \mathrm{Perm}_K(\mathcal{H})$, which corresponds to selecting K distinct records from $\mathcal{H}$ in a particular order. The action thus determines not only which records to use but also how they are arranged. The size of the action space is $N!/(N-K)!$.
- **Reward:** $R(\mathrm{LLM}(\mathcal{P}, x), y)$, a function that measures the quality of the LLM-generated response $\mathrm{LLM}(\mathcal{P}, x)$ relative to the ground-truth personalized response y.

We model the policy with a neural distribution $\pi_{\boldsymbol{\theta}}(\cdot \mid \mathcal{C})$, parameterized by $\boldsymbol{\theta}$, which assigns probabilities to candidate user profiles given the context $\mathcal{C}$. The objective is to learn parameters $\boldsymbol{\theta}$ such that the policy assigns higher probabilities to more informative profiles, which ultimately enhance personalized text generation. To this end, we maximize the expected reward over sampled user profiles, optimizing the following objective on a dataset $\mathcal{D}$ spanning multiple users, each associated with a set of history records, a query, and the corresponding ground-truth answer:

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathcal{H},x,y)\sim\mathcal{D},\mathcal{P}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathcal{C})}[R(\mathrm{LLM}(\mathcal{P}, x), y)]. \quad (1)$$

It is challenging to directly optimize Equation 1 since the reward is not differentiable. To address this, we employ the likelihood ratio gradient estimator from reinforcement learning and stochastic optimization (Williams, 1992; Sutton et al., 1999), which allows us to compute the gradient as:

$$\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{(\mathcal{H},x,y)\sim\mathcal{D},\mathcal{P}\sim\pi_{\boldsymbol{\theta}}(\cdot|\mathcal{C})} [\nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(\mathcal{P} \mid \mathcal{C}) R(\mathrm{LLM}(\mathcal{P}, x), y)]. \quad (2)$$

Since it is intractable to enumerate all profiles $\mathcal{P} \in \mathrm{Perm}_K(\mathcal{H})$ during the optimization process, we estimate Equation 2 by randomly sampling $M = 32$ profiles. To stabilize training and reduce variance in gradient estimation, we apply z-score normalization over the rewards of these 32 profiles sampled for each example. The detailed gradient estimation scheme is provided in Equation 2 in Appendix A.

3.2 Model and Function Design

Design of $\pi_{\boldsymbol{\theta}}(\cdot \mid \mathcal{C})$ Since different permutations of the selected records may lead to different final responses, we adopt the Plackett–Luce (PL) model, which assigns probabilities to profiles based on the scores of individual user records. Therefore, $\pi_{\boldsymbol{\theta}}(\cdot \mid \mathcal{C})$ defines a distribution over all $N!/(N-K)!$ permutations of length K drawn from the user's N history records. The probability assigned to a specific profile $\mathcal{P}$ is given by:

$$\pi_{\boldsymbol{\theta}}(\mathcal{P} \mid \mathcal{C}) = \prod_{k=1}^{K} \frac{f_{\boldsymbol{\theta}}(p_k; \mathcal{C})}{S - \sum_{j=1}^{k-1} f_{\boldsymbol{\theta}}(p_j; \mathcal{C}))}, \quad (3)$$

where $S = \sum_{i=1}^{N} f_{\boldsymbol{\theta}}(h_i; \mathcal{C})$, and $f_{\boldsymbol{\theta}}(\cdot)$ is the user record encoder that outputs a propensity score in $[0, 1]$ for each record, indicating the model's tendency to include that record in the user profile. During training, profiles are generated by sampling K records without replacement based on Equation 3. At inference time, the top-K records ranked by propensity scores are selected. Because our user record encoder is order-aware and rewards are assigned to ordered sets, the learned propensity score can be interpreted as each record's contribution to the selected set. We further show in Sec. 5.3 that this ordering achieves higher final utility compared with other baselines.

Design of $f_{\boldsymbol{\theta}}$ For the record encoder $f_{\boldsymbol{\theta}}$, we aim to capture the interdependencies among user records. A key design consideration is the trade-off between modeling dependencies at the token level versus the sentence level. While the former could, in principle, capture finer-grained interactions, it would quickly exceed the encoder's context length. To address this, we adopt a late

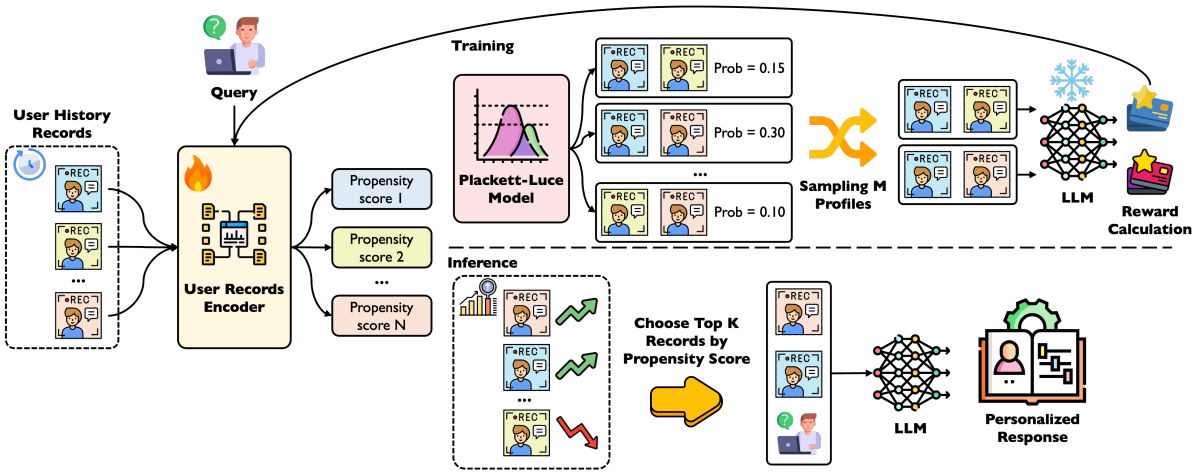

Figure 2: Workflow of the proposed PURPLE framework. User records encoder takes a user query and a list of user history records as input, outputting the propensity scores of all records. During **training**, a Plackett-Luce model is employed to convert the propensity scores to a probability distribution over all possible profiles, followed by sampling M profiles for gradient estimation. During **inference**, records with top K propensity scores are provided to the LLM along with the user query to generate a personalized response.

interaction strategy (Khattab and Zaharia, 2020), where we first obtain sentence-level embeddings with a pre-trained encoder, and then apply a Transformer encoder to model dependencies across records. Figure 2 illustrates the overall workflow of our method. Within the user record encoder, we utilize a pre-trained Contriever (Izacard et al., 2022) to obtain token embeddings for both the query and the records. Each record first cross-attends to the query at the token level, producing query-fused record embeddings that incorporate query information. A subsequent pooling operation is then applied to the updated record token embeddings to produce fixed-size sentence-level embeddings. embeddings are then processed by a Transformer encoder to model cross-record dependencies. We omit positional encodings to avoid ordering bias among records.

Design of Reward Function In this work, we propose an LLM-driven reward, where the policy is trained to maximize the log-likelihood that the LLM assigns to the target sequence. Formally, given a user profile $\mathcal{P}$, a query x, and a ground-truth personalized response y, we define the reward as:

$$R(\text{LLM}(\mathcal{P}, x), y) = \log p_\phi(y \mid \mathcal{P}, x) = \sum_{j=1}^{|y|} \log p_\phi(y_j \mid \mathcal{P}, x, y_{<j}), \quad (4)$$

where ϕ are the parameters of the LLM and $p_\phi(\cdot)$ denotes its next-token distribution. Using the log-likelihood of ground-truth sequences as the reward provides dense feedback signals, in contrast to downstream metrics such as accuracy, mean squared error, or ROUGE-1 (Liu et al., 2025). Moreover, we show in Appendix B that this objective is equivalent to maximizing the evidence lower bound (ELBO) of the marginalization-based RAG approach (Lewis et al., 2020), which becomes intractable in our setting due to the combinatorial explosion. In the next section, we empirically demonstrate that this log-likelihood-based reward is robust across diverse downstream tasks.

4 Experiments

4.1 Dataset and Evaluation

We evaluate the performance of PURPLE using `Phi-4-Mini-Instruct` (Microsoft, 2025) and `Llama-3-8B-Instruct` (Team, 2024) as the frozen LLM, and further scale to `Llama-3-70B-Instruct` (Team, 2024). Our experiments span a wide range of personalization settings, including personalized classification, regression, and both short- and long-text generation from the LaMP (Salemi et al., 2024) and LongLaMP (Kumar et al., 2024) benchmarks. We follow the prompt templates of Salemi et al. (2024) and Kumar et al. (2024) to incorporate user profiles into the original queries.

Specifically, we evaluate PURPLE on **nine personalization tasks**: two classification tasks — *Personalized Citation Identification* (Citation) and *Personalized Movie Tagging* (Movie), evalu-

Table 1: Results on the LaMP benchmark. The best and second-best results in each column are highlighted in **bold** and underlined, respectively. We report the mean and standard deviation over three runs with different random seeds for LLM generation. For `GPT-5-nano`, we report only a single run due to API cost constraints.

Task	Citation	Movie	Rating	News	Scholar	Tweet
Metric	Acc. / F1	Acc. / F1	MAE / RMSE	RG1 / RGL / MT	RG1 / RGL / MT	RG1 / RGL / MT
With Phi-4-Mini-Instruct (3.84B)						
BM25	$63.9_{0.53}$ / $63.8_{0.69}$	$34.5_{0.65}$ / $29.6_{0.61}$	$0.438_{0.00}$ / $0.852_{0.01}$	$14.4_{0.16}$ / $12.8_{0.16}$ / $11.9_{0.12}$	$39.7_{0.08}$ / $33.2_{0.04}$ / $42.2_{0.08}$	$38.3_{0.04}$ / $33.6_{0.08}$ / $35.2_{0.04}$
Contriever	$64.6_{0.33}$ / $64.5_{0.20}$	$36.0_{0.37}$ / $31.1_{0.24}$	$0.424_{0.01}$ / $0.830_{0.03}$	$14.6_{0.04}$ / $13.1_{0.00}$ / $12.2_{0.04}$	$39.7_{0.00}$ / $33.3_{0.08}$ / $42.0_{0.08}$	$38.6_{0.08}$ / $33.7_{0.08}$ / $35.8_{0.04}$
IC-RALM	$62.0_{0.20}$ / $61.7_{0.37}$	$33.6_{0.12}$ / $28.8_{0.16}$	$0.471_{0.01}$ / $0.857_{0.02}$	$13.4_{0.04}$ / $11.9_{0.04}$ / $11.0_{0.00}$	$37.6_{0.12}$ / $31.0_{0.12}$ / $41.0_{0.33}$	$38.1_{0.12}$ / $33.3_{0.16}$ / $35.2_{0.12}$
REPLUG	$61.2_{0.94}$ / $60.2_{0.78}$	$37.4_{0.49}$ / $32.1_{0.37}$	$0.486_{0.01}$ / $0.891_{0.02}$	$14.1_{0.04}$ / $12.7_{0.00}$ / $11.4_{0.08}$	$38.1_{1.63}$ / $32.8_{0.90}$ / $40.4_{1.51}$	**$42.2_{0.08}$ / $37.3_{0.00}$ / $38.6_{0.04}$**
RankGPT (Llama-3-8B-Instruct)	$63.9_{0.82}$ / $63.7_{0.65}$	$34.2_{0.94}$ / $28.9_{1.18}$	$0.446_{0.00}$ / $0.863_{0.01}$	$14.7_{0.33}$ / $13.1_{0.24}$ / $12.3_{0.24}$	$39.7_{0.00}$ / $33.3_{0.04}$ / $42.0_{0.04}$	$38.2_{0.04}$ / $33.4_{0.02}$ / $35.2_{0.08}$
RankGPT (GPT-5-nano)	65.9 / **65.6**	35.5 / 31.4	0.444 / 0.865	14.6 / 13.0 / 12.1	39.8 / 33.4 / 42.3	38.5 / 33.7 / 35.5
ICR (Llama-3-8B-Instruct)	$65.2_{0.53}$ / $65.0_{0.45}$	$34.1_{0.73}$ / $29.8_{1.02}$	$0.424_{0.00}$ / $0.830_{0.02}$	$15.0_{0.00}$ / $13.4_{0.04}$ / $12.5_{0.00}$	$39.5_{0.08}$ / $33.1_{0.08}$ / $42.0_{0.04}$	$38.6_{0.12}$ / $33.6_{0.24}$ / $35.5_{0.16}$
PURPLE (Ours)	**$66.0_{0.20}$ / $65.6_{0.16}$**	**$38.6_{0.33}$ / $34.2_{0.53}$**	**$0.419_{0.01}$ / $0.808_{0.02}$**	**$15.1_{0.04}$ / $13.5_{0.00}$ / $12.6_{0.08}$**	**$40.0_{0.04}$ / $33.5_{0.04}$ / $42.4_{0.00}$**	$39.0_{0.04}$ / $34.0_{0.04}$ / $35.9_{0.00}$
With Llama-3-8B-Instruct (8.03B)						
BM25	$56.9_{0.61}$ / $56.5_{0.53}$	$45.2_{0.37}$ / $36.5_{0.98}$	$0.328_{0.01}$ / $0.664_{0.02}$	$16.7_{0.33}$ / $15.1_{0.37}$ / $14.7_{0.33}$	$41.0_{0.02}$ / $35.2_{0.04}$ / $40.6_{0.02}$	$31.4_{0.16}$ / $26.6_{0.16}$ / $27.6_{0.24}$
Contriever	$58.5_{0.20}$ / $58.1_{0.41}$	$47.2_{0.37}$ / $39.1_{0.29}$	$0.314_{0.00}$ / $0.631_{0.01}$	$17.2_{0.04}$ / $15.6_{0.04}$ / $15.1_{0.04}$	$41.1_{0.08}$ / $35.4_{0.12}$ / $40.5_{0.04}$	$32.1_{0.16}$ / $27.2_{0.24}$ / $28.5_{0.16}$
IC-RALM	$59.3_{0.04}$ / $56.6_{0.29}$	$38.0_{0.82}$ / $29.9_{0.41}$	$0.351_{0.01}$ / $0.671_{0.01}$	$14.1_{0.24}$ / $12.5_{0.24}$ / $12.1_{0.08}$	$37.5_{0.08}$ / $31.5_{0.04}$ / $39.2_{0.04}$	$29.2_{0.69}$ / $24.6_{0.57}$ / $25.6_{0.49}$
REPLUG	$55.5_{1.06}$ / $47.5_{2.00}$	$42.0_{1.35}$ / $31.8_{1.14}$	$0.314_{0.00}$ / $0.633_{0.00}$	$14.8_{0.04}$ / $13.2_{0.00}$ / $11.8_{0.04}$	**$42.5_{0.08}$ / $37.1_{0.06}$ / $40.8_{0.12}$**	$30.4_{0.20}$ / $26.1_{0.12}$ / $25.8_{0.37}$
RankGPT (Llama-3-8B-Instruct)	$57.0_{0.20}$ / $56.3_{0.12}$	$47.2_{0.90}$ / $38.4_{0.53}$	$0.318_{0.01}$ / $0.637_{0.01}$	$16.9_{0.20}$ / $15.3_{0.16}$ / $14.8_{0.29}$	$41.0_{0.04}$ / $35.4_{0.08}$ / $40.5_{0.16}$	$31.0_{0.12}$ / $26.3_{0.08}$ / $27.3_{0.16}$
RankGPT (GPT-5-nano)	**59.5** / 58.0	45.1 / 36.2	0.321 / 0.638	17.1 / 15.4 / 15.0	41.0 / 35.3 / 40.5	31.5 / 26.5 / 27.8
ICR (Llama-3-8B-Instruct)	$58.4_{0.29}$ / $57.3_{0.37}$	$48.0_{0.37}$ / $39.3_{0.73}$	$0.312_{0.01}$ / $0.631_{0.02}$	$17.1_{0.04}$ / $15.4_{0.00}$ / $14.9_{0.00}$	$41.3_{0.16}$ / $35.5_{0.29}$ / **$40.8_{0.24}$**	$31.8_{0.33}$ / $26.8_{0.24}$ / $28.1_{0.29}$
PURPLE (Ours)	$59.2_{0.82}$ / **$58.8_{0.78}$**	**$49.6_{0.65}$ / $41.6_{0.53}$**	**$0.307_{0.01}$ / $0.624_{0.01}$**	**$17.6_{0.04}$ / $15.9_{0.00}$ / $15.3_{0.08}$**	$41.4_{0.49}$ / $35.8_{0.73}$ / $40.3_{0.37}$	**$32.5_{0.04}$ / $27.5_{0.00}$ / $28.8_{0.04}$**
With Llama-3-70B-Instruct (70.6B)						
BM25	$71.3_{0.37}$ / $70.8_{0.37}$	$54.5_{0.37}$ / $47.2_{0.41}$	$0.245_{0.01}$ / $0.544_{0.01}$	$18.1_{0.33}$ / $16.5_{0.29}$ / $14.9_{0.37}$	$43.8_{0.53}$ / $38.2_{0.41}$ / $40.5_{0.49}$	$36.3_{0.16}$ / $30.9_{0.20}$ / $33.0_{0.12}$
Contriever	$71.2_{0.78}$ / $70.8_{0.73}$	$56.6_{0.16}$ / $49.5_{0.29}$	$0.238_{0.00}$ / $0.528_{0.00}$	$18.6_{0.08}$ / $17.0_{0.08}$ / $15.6_{0.08}$	$44.1_{0.08}$ / $38.6_{0.12}$ / $41.1_{0.00}$	$36.5_{0.00}$ / $31.4_{0.00}$ / $33.3_{0.00}$
IC-RALM	$66.5_{0.04}$ / $66.4_{0.00}$	$49.0_{0.29}$ / $41.8_{0.12}$	$0.261_{0.00}$ / $0.555_{0.00}$	$14.8_{0.02}$ / $13.3_{0.04}$ / $12.2_{0.00}$	$39.7_{0.00}$ / $34.1_{0.24}$ / $38.6_{0.16}$	$32.0_{0.20}$ / $27.4_{0.00}$ / $28.8_{0.20}$
REPLUG	$65.8_{0.33}$ / $65.4_{0.41}$	$51.7_{0.04}$ / $44.0_{0.08}$	$0.258_{0.01}$ / $0.554_{0.02}$	$15.2_{0.04}$ / $13.9_{0.04}$ / $12.1_{0.04}$	$41.8_{0.12}$ / $36.6_{0.33}$ / $38.6_{0.35}$	$32.1_{0.08}$ / $27.6_{0.12}$ / $27.7_{0.08}$
RankGPT (Llama-3-8B-Instruct)	$69.7_{0.16}$ / $69.1_{0.12}$	$56.9_{0.16}$ / $48.8_{0.41}$	$0.247_{0.00}$ / $0.546_{0.01}$	$18.0_{0.24}$ / $16.5_{0.29}$ / $15.2_{0.29}$	$44.0_{0.04}$ / $38.6_{0.08}$ / $41.1_{0.12}$	$35.8_{0.00}$ / $30.6_{0.00}$ / $32.4_{0.08}$
RankGPT (GPT-5-nano)	**73.8 / 73.5**	55.3 / 48.2	0.240 / 0.523	18.7 / 17.0 / **15.8**	**44.6 / 38.8 / 41.5**	36.6 / 31.3 / 33.2
ICR (Llama-3-8B-Instruct)	$71.6_{0.16}$ / $71.0_{0.16}$	$56.8_{0.24}$ / $49.6_{0.57}$	$0.238_{0.00}$ / $0.531_{0.00}$	$18.3_{0.00}$ / $16.8_{0.04}$ / $15.3_{0.16}$	$44.3_{0.06}$ / $38.5_{0.04}$ / $41.2_{0.20}$	$36.3_{0.16}$ / $31.1_{0.24}$ / $33.2_{0.20}$
PURPLE (Ours)	$72.2_{0.49}$ / $71.8_{0.57}$	**$57.0_{0.08}$ / $49.9_{0.41}$**	**$0.236_{0.02}$ / $0.520_{0.01}$**	**$18.8_{0.04}$ / $17.1_{0.00}$** / $15.6_{0.08}$	$44.4_{0.02}$ / **$38.8_{0.06}$** / $41.0_{0.04}$	**$37.3_{0.01}$ / $32.1_{0.04}$ / $34.0_{0.06}$**

ated with Accuracy and F1; one regression task — *Personalized Product Rating* (Rating), evaluated with MAE and RMSE; and six generation tasks, evaluated with ROUGE-1 (RG1), ROUGE-L (RGL) (Lin, 2004), and METEOR (MT) (Banerjee and Lavie, 2005). The generation tasks are further divided into short-text generation — *Personalized News Headline Generation* (News), *Personalized Scholarly Title Generation* (Scholar), and *Personalized Tweet Paraphrasing* (Tweet) — and long-text generation — *Personalized Abstract Generation* (Abstract), *Personalized Topic Generation* (Topic), and *Personalized Product Review Generation* (Review). In all experiments, we first use Contriever (Izacard et al., 2022) to retrieve the top 20 records as the user history $\mathcal{H}$, and then select 5 of them to construct the user profile $\mathcal{P}$. Appendix C further elaborates implementation details.

4.2 Baseline Methods

We focus on the setting where the LLM is kept frozen and no ground-truth profile is available for training the reranker. Therefore, we compare with three categories of prior methods that, likewise, neither fine-tune the LLM nor rely on supervision from ground-truth retrieval results.

The baselines we compare with include **(i) Zero-Shot Rerankers**, represented by RankGPT (Sun et al., 2023) and ICR (Chen et al., 2025). For both methods, we adopt `Llama-3-8B-Instruct` as the reranker LLM. We also report the performance of RankGPT with `GPT-5-nano` to reflect an upper bound of the methods that distill from the ranking results of state-of-the-art proprietary LLMs (Pradeep et al., 2023a,b; Tamber et al., 2023; Gangi Reddy et al., 2024). **(ii) In-Context Retrieval-Augmented Language Models**, represented by IC-RALM (Ram et al., 2023) and REPLUG (Shi et al., 2024). Both methods consider only one record at a time when generating a response. They incorporate multiple records from the user profile either through marginalization

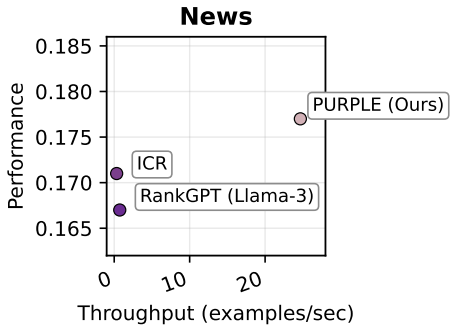

Figure 3: Performance–throughput graph on the News dataset. PURPLE is faster than LLM-based rerankers while achieving better performance.

Table 2: Ablation studies of PURPLE using `Phi-4-Mini-Instruct`. CA and RDM stand for cross-attention and record dependency modeling, respectively. MR refers to using task-specific evaluation metric as the reward.

Task	**Citation**	**Movie**	**Rating**	**News**	**Scholar**	**Tweet**
Metric	Acc. / F1	Acc. / F1	MAE / RMSE	RG1 / RGL / MT	RG1 / RGL / MT	RG1 / RGL / MT
PURPLE	**66.2 / 65.8**	**38.2 / 33.6**	**0.405 / 0.788**	**15.2 / 13.5 / 12.5**	**40.0 / 33.5 / 42.4**	**39.1** / 34.0 / 35.9
w/o CA	64.8 / 64.5	35.1 / 29.7	0.440 / 0.816	14.8 / 13.2 / 12.4	**40.0 / 33.5** / 42.2	**39.1 / 34.1** / 36.0
w/o RDM	61.3 / 60.6	35.0 / 31.1	0.449 / 0.850	14.5 / 12.8 / 11.9	39.7 / 33.1 / 41.9	39.0 / 34.0 / **36.1**
w/ MR	64.8 / 64.8	38.0 / 33.3	0.433 / 0.854	15.0 / 13.2 / 12.4	39.5 / 32.9 / 41.8	38.7 / 33.7 / 35.6

(REPLUG) or context switching (In-Context RALM). Additionally, we include **(iii) Efficient Sparse and Dense Retrievers**, applied directly as rerankers. We include BM25 (Robertson and Zaragoza, 2009) for the sparse retriever and Contriever (Izacard et al., 2022) for the dense retriever. These methods represent the efficiency-oriented side of the efficiency–performance trade-off.

5 Experiment Results

5.1 Overall Performance Comparison

Table 1 presents the results of PURPLE and baseline methods on the LaMP benchmark, while Table 3 presents the results on the LongLaMP benchmark. The main findings are as follows:

PURPLE consistently outperforms strong baselines across LLM scales Across all tasks and LLMs of varying sizes, PURPLE achieves consistent improvements over existing methods. Compared with Contriever, which is of comparable model size, our learned propensity scores provide more effective ranking signals than raw relevance. Compared with zero-shot rerankers, which use much larger backbone LLMs and incur higher inference cost, PURPLE achieves stronger personalization with a much smaller model, since reinforcement learning training allows us to better capture the utility of profiles formed by multiple records. Compared with in-context RALMs, which provide user records one at a time to the LLM and combine multiple records post hoc, our single-stage modeling more effectively captures personalized signals, highlighting the advantage of treating user profiles holistically.

PURPLE outperforms strong baselines with high computational throughput. Figure 3 shows that, on a representative LaMP dataset, PURPLE outperforms existing methods while being more efficient. PURPLE achieves higher performance than ICR and RankGPT (Llama-3) while maintaining high computational throughput.

PURPLE is effective across task types, including regression. As our reward is based on the log-probability that the LLM assigns to the ground-truth response, it does not directly reflect numerical distances between regression targets. Nevertheless, PURPLE still achieves strong gains on the regression task (Rating). This demonstrates that log-probability provides a principled and broadly applicable reward signal across diverse task formats.

5.2 Ablation Studies

Table 2 presents ablation studies of PURPLE using `Phi-4-Mini-Instruct`. Overall, we examine three key design choices. First, instead of performing token-level cross attention, we test a simplified variant that encodes the entire query into a single embedding and prepends it as an extra token to the Transformer encoder (w/o CA). This approach is less effective, indicating that fine-grained token-level interactions between the query and user records are crucial for accurate personalization. Second, we remove the Transformer encoder entirely, resulting in a point-wise scoring model where each record is processed independently (w/o RDM). This variant shows the largest performance drop across tasks. Although the model can still leverage individually informative records, it fails to model dependencies such as redundancy and complementarity among records. Third, we examine an alternative reward design that uses task-specific evaluation metrics as the reward (Accuracy, MAE, and ROUGE-1). This metric-based reward is substantially more sparse and coarse-grained, providing weaker learning signals and making optimization more challenging. Empirically, this variant consistently underperforms PURPLE across all tasks, thereby

strongly motivating our reward design choice.

These results highlight that token-level cross attention, cross-record dependency modeling, and the log-probability-based reward are all indispensable, validating our design choice of treating user profiles as structured contexts rather than isolated records, and providing denser reward signals via ground-truth log-probabilities given by the LLM.

5.3 Analysis: Selecting topK at inference

To further examine the quality of the learned propensity scores, we compare the top-5 selections of PURPLE, ICR, RankGPT, and Contriever on Citation, Rating, News, and Tweet. For each test example, we consider the top-5 records proposed by each method and randomly sample five orderings from the $5! = 120$ permutations as controls. As shown in Figure 4, orderings induced by our learned propensity scores are more frequently ranked as the best among the six candidates. This result indicates that our scoring function better captures relative dependencies between records, rather than relying on local pairwise relevance alone. These findings highlight that our method not only identifies useful records but also arranges them in an order that maximizes downstream personalization utility.

5.4 Human Evaluation

To complement our automatic evaluation and verify the practical utility of the selected profiles, we conduct a rigorous side-by-side human evaluation. While automatic metrics measure lexical overlap, they often fail to capture the subtle stylistic nuances, such as voice, stance, and informal phrasing, that define true personalization. We focus our human evaluation on *Personalized Tweet Paraphrasing* (Tweet), which requires significant stylistic adaptation to user personas. We randomly sample 250 instances. For each instance, human evaluators are presented with the input query, a snippet of the user's history, and two anonymized model outputs: one generated using one of the most recent baseline ICR and one using PURPLE. Evaluators perform a blind, forced-choice comparison to determine which output better reflected the user's specific persona while maintaining semantic fidelity. The results demonstrate that PURPLE achieves a decisive victory, surpassing ICR by a margin of $+14.4\%$ (57.2% vs. 42.8%). This confirms that our method effectively captures personal stylistic features, such as slang usage and informal tone, where heuristic relevance retrieval often fails. We provide detailed qualitative case studies analyzing these behaviors in Appendix D.

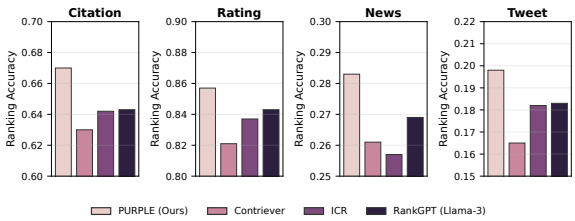

Figure 4: Ranking accuracy across various tasks. PURPLE achieves the highest accuracy on the examined datasets, consistently outperforming heuristic retrievers, LLM-based rerankers, and in-context rerankers.

6 Conclusion

In this work, we study the problem of retrieval-augmented personalization for large language models. Our key intuitions are twofold: (i) *record relevance does not reliably predict personalization utility*, and (ii) *utility is non-monotonic* across records, making greedy aggregation suboptimal. To address these limitations, we propose **PURPLE**, a contextual bandit framework that optimizes user profiles by directly leveraging downstream performance as feedback. PURPLE jointly models query–record interactions and cross-record dependencies, enabling adaptive selection of user profiles beyond static heuristics. Instead of raw task-specific metrics, PURPLE relies on LLM-based probability as the reward, providing denser reward signals that facilitate more effective training. Extensive experiments on nine real-world personalization tasks spanning classification, regression, and text generation show that PURPLE consistently outperforms heuristic retrievers, LLM-based rerankers, and in-context RALMs, while being significantly more efficient. These results establish contextual bandit-based retrieval as a principled and scalable paradigm for personalizing LLMs.

Limitations

While PURPLE demonstrates strong effectiveness across diverse personalization tasks, it has several limitations. First, the method relies on the log-likelihood of ground-truth personalized responses under a frozen LLM as the training reward, which assumes access to high-quality personalized supervision. In practical deployment, such explicit supervision may be sparse or unavailable, with user feedback often taking implicit or noisy forms (e.g., engagement signals), and extending PURPLE to these settings remains an open challenge. Second, although using log probability as the reward provides a unified and task-agnostic optimization signal across classification, regression, and generation tasks, PURPLE is still trained separately for each dataset and task in our current study. We do not explicitly evaluate the extent to which a policy learned under this unified objective can generalize across tasks or domains, and exploring more transferable training and deployment strategies is left for future work.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In *The Twelfth International Conference on Learning Representations*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Shijie Chen, Bernal Jimenez Gutierrez, and Yu Su. 2025. Attention in Large Language Models Yields Efficient Zero-Shot Re-Rankers. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Revanth Gangi Reddy, JaeHyeok Doo, Yifei Xu, Md Arafat Sultan, Deevya Swain, Avirup Sil, and Heng Ji. 2024. FIRST: Faster improved listwise reranking with single token decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8642–8652, Miami, Florida, USA. Association for Computational Linguistics.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Naik, Pengshan Cai, and Alfio Gliozzo. 2022. Re2G: Retrieve, Rerank, Generate. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2701–2715, Seattle, United States. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3929–3938. PMLR.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Edward J. Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Transactions on Machine Learning Research*.

Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. 2025. Know me, respond to me: Benchmarking llms for dynamic user profiling and personalized responses at scale. *Preprint*, arXiv:2504.14225.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense Passage Retrieval for Open-Domain Question Answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Omar Khattab and Matei Zaharia. 2020. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sigir '20, pages 39–48, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A. Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, Nedim Lipka, Chien Van Nguyen, Thien Huu Nguyen, and Hamed Zamani. 2024. LongLaMP: A Benchmark for Personalized Long-form Text Generation.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, Sosp '23, pages 611–626, New York, NY, USA. Association for Computing Machinery.

John Langford and Tong Zhang. 2007. The Epoch-Greedy algorithm for contextual multi-armed bandits. In *Proceedings of the 21st International Conference on Neural Information Processing Systems*, NIPS'07, pages 817–824, Red Hook, NY, USA. Curran Associates Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. 2025. NOVER: Incentive Training for Language Models via Verifier-Free Reinforcement Learning.

Microsoft. 2025. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-LoRAs. *Preprint*, arXiv:2503.01743.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pre-trained sequence-to-sequence model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 708–718, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with BERT. *Preprint*, arXiv:1910.14424.

OpenAI. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Emiliano Penaloza, Olivier Gouvert, Haolun Wu, and Laurent Charlin. 2025. Tears: Text representations for scrutable recommendations. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 4949–4968, New York, NY, USA. Association for Computing Machinery.

Ronak Pradeep, Rodrigo Nogueira, and Jimmy Lin. 2021. The expando-mono-duo design pattern for text ranking with pretrained sequence-to-sequence models. *Preprint*, arXiv:2101.05667.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023a. RankVicuna: Zero-shot listwise document reranking with open-source large language models. *Preprint*, arXiv:2309.15088.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023b. RankZephyr: Effective and robust zero-shot listwise reranking is a breeze! *Preprint*, arXiv:2312.02724.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-Context Retrieval-Augmented Language Models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331.

Stephen Robertson and Hugo Zaragoza. 2009. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When Large Language Models Meet Personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

Xiaoteng Shen, Rui Zhang, Xiaoyan Zhao, Jieming Zhu, and Xi Xiao. 2024. PMG : Personalized multimodal generation with large language models. In *Proceedings of the ACM Web Conference 2024*, Www '24, pages 3833–3843, New York, NY, USA. Association for Computing Machinery.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Richard James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2024. REPLUG: Retrieval-Augmented Black-Box Language Models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8371–8384, Mexico City, Mexico. Association for Computational Linguistics.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT Good at Search? Investigating Large Language Models as Re-Ranking Agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

Manveer Singh Tamber, Ronak Pradeep, and Jimmy Lin. 2023. Scaling down, LiTting up: Efficient zero-shot listwise reranking with seq2seq encoder-decoder models. *Preprint*, arXiv:2312.16098.

Llama Team. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256.

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. *Preprint*, arXiv:2406.17803.

Shangyu Wu, Ying Xiong, Yufei Cui, Haolun Wu, Can Chen, Ye Yuan, Lianming Huang, Xue Liu, Tei-Wei Kuo, Nan Guan, and Chun Jason Xue. 2025. Retrieval-augmented generation for natural language processing: A survey. *Preprint*, arXiv:2407.13193.

Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. RankRAG: Unifying Context Ranking with Retrieval-Augmented Generation in LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pages 121156–121184. Curran Associates, Inc.

Lemei Zhang, Peng Liu, Marcus Tiedemann Oekland Henriksboe, Even W. Lauvrak, Jon Atle Gulla, and Heri Ramampiaro. 2024. PersonalSum: A user-subjective guided personalized summarization dataset for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 99333–99346. Curran Associates, Inc.

A Details of Gradient Estimation

To estimate the gradient in Equation 2, we first draw a batch of examples $\{(\mathcal{H}_b, x_b, y_b)\}_{b=1}^{B}$. For each example, we sample M profiles $\mathcal{P}_b^1, \ldots, \mathcal{P}_b^M$ from $\pi_\theta(\cdot \mid \mathcal{C}_b)$, and finally compute the empirical mean. This learning procedure corresponds to the REINFORCE algorithm (Sutton et al., 1999), with gradient estimate:

$$\nabla_\theta \mathcal{J}(\theta) \approx \frac{1}{B}\sum_{b=1}^{B}\frac{1}{M}\sum_{m=1}^{M}\nabla_\theta \log \pi_\theta(\mathcal{P}_b^m \mid \mathcal{C}_b)\tilde{r}_b^m. \quad (5)$$

To reduce variance in gradient estimation, we apply reward normalization over the M profiles sampled for each example. Concretely, for each example with rewards $\boldsymbol{r}_b = [r_b^1, \ldots, r_b^M]^\top$, where $r_b^m = R(\Phi(\mathcal{P}_b^m, x_b), y_b)$, the normalized reward is computed as $\tilde{r}_b^m = \frac{r_b^m - \text{mean}(\boldsymbol{r}_b)}{\text{std}(\boldsymbol{r}_b)}$.

B Motivating our Reward

The specific choice of using the log-probability of ground truth personalized response is grounded in the generative modeling perspective of retrieval-augmented generation (RAG) (Lewis et al., 2020), where the user profile is treated as a latent variable and the response likelihood is obtained by marginalizing over all possible profile selections. Applying Jensen's inequality to the training objective in this setting gives:

$$\mathbb{E}_{(\mathcal{H},x,y)\sim\mathcal{D}}\left[\log \sum_{\mathcal{P}\in \text{Perm}_K(\mathcal{H})} \pi_\theta(\mathcal{P} \mid \mathcal{C}) p_\Phi(y \mid \mathcal{P}, x)\right] \quad (6)$$
$$\geq \mathbb{E}_{(\mathcal{H},x,y)\sim\mathcal{D},\mathcal{P}\sim\pi_\theta(\cdot|\mathcal{C})}[\log p_\Phi(y \mid \mathcal{P}, x)].$$

Therefore, maximizing the expected reward under our reinforcement learning objective is equivalent to maximizing the evidence lower bound (ELBO), with p_Φ modeled by a frozen LLM.

C Implementation Details

We employ a frozen pre-trained Contriever to first encode both queries and user history records into token embeddings. The only trainable components are the remaining modules of the user records encoder. These include a cross-attention layer that integrates query information into record embeddings, a Transformer encoder that captures inter-record dependencies, and an MLP decoder that maps the updated record encodings into scalar propensity scores. We set the number of Transformer encoder layers to $l = 12$, resulting in a parameter size roughly twice that of Contriever, while still being substantially smaller than the zero-shot rerankers and in-context RALMs used as baselines. For gradient estimation, we use a batch size of $B = 16$ and sample $M = 32$ user profiles for each example. We train the model for 10 epochs using the Adam optimizer (Kingma and Ba, 2017) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 1×10^{-4}. During training, we apply a gradient clipping norm of 1.0. The checkpoint achieving the best validation performance is selected for testing.

In all experiments, we use frozen LLMs both to generate personalized responses and to evaluate the log-probability of ground-truth responses conditioned on the query and user profiles (i.e., our reward). For generation, we set the temperature to $T = 0.7$ and employ nucleus sampling (Holtzman et al., 2020) with $top\_p = 0.8$. For `Phi-4-Mini-Instruct` and `Llama-3-8B-Instruct`, we deploy on a single NVIDIA H100 GPU. For `Llama-3-70B-Instruct`, we deploy the model across four NVIDIA H100 GPUs using vLLM (Kwon et al., 2023). All LLMs are deployed in BF16 precision. Training of PURPLE is conducted on the same GPUs used for LLM deployment.

D Qualitative Case Studies

In this section, we present detailed case studies from our human evaluation to illustrate the qualitative superiority of PURPLE over the baseline ICR. We present one representative example where the two methods diverge significantly.

In the Tweet task, preserving the logic of the original sentence is as critical as matching the user's informal voice. The example below demonstrates a failure in logical reasoning by the baseline. The input tweet contains a comparative structure ("at least we attended... unlike someone"). The baseline fails to parse this dependency, resulting in a paraphrase that states the exact opposite of the truth. PURPLE successfully disentangles the complex sentence structure and preserves the correct meaning.

Table 3: Results on the LongLaMP benchmark. The best and second-best results in each column are highlighted in **bold** and underlined, respectively. We report the mean and standard deviation over three runs with different random seeds for LLM generation. For `GPT-5-nano`, we report only a single run due to API cost constraints.

Task	**Abstract**	**Topic**	**Review**
Metric	R1 / RL / MT	R1 / RL / MT	R1 / RL / MT
With Phi-4-Mini-Instruct (3.84B)			
BM25	$38.6_{0.12}$ / $22.0_{0.16}$ / $26.2_{0.08}$	$25.2_{0.41}$ / $12.7_{0.20}$ / $\underline{17.1}_{0.16}$	$25.8_{1.39}$ / $13.2_{0.53}$ / $16.5_{0.16}$
Contriever	$38.5_{0.04}$ / $21.9_{0.12}$ / $26.1_{0.08}$	$\underline{25.7}_{1.80}$ / $\underline{13.0}_{0.73}$ / $16.6_{0.20}$	$25.2_{1.92}$ / $13.1_{0.57}$ / $15.9_{0.78}$
IC-RALM	$37.2_{0.04}$ / $21.1_{0.02}$ / $25.1_{0.04}$	$24.8_{1.47}$ / $12.4_{0.78}$ / $15.9_{0.20}$	$25.0_{1.35}$ / $12.7_{0.49}$ / $15.8_{0.16}$
REPLUG	$36.0_{0.24}$ / $21.2_{0.20}$ / $23.8_{0.20}$	$23.5_{1.06}$ / $12.1_{0.20}$ / $13.1_{0.41}$	$20.8_{2.94}$ / $11.2_{1.14}$ / $12.5_{1.63}$
RankGPT (Llama-3-8B-Instruct)	$38.6_{0.12}$ / $22.0_{0.08}$ / $26.2_{0.08}$	$25.1_{0.53}$ / $12.7_{0.29}$ / $\underline{17.1}_{0.18}$	$25.1_{1.63}$ / $12.9_{0.53}$ / $15.8_{0.45}$
RankGPT (GPT-5-nano)	**39.1 / 22.4 / 26.9**	24.9 / 12.5 / **17.5**	$\underline{27.1}$ / $\underline{13.7}$ / $\underline{16.6}$
ICR (Llama-3-8B-Instruct)	$38.5_{0.24}$ / $22.0_{0.16}$ / $26.2_{0.16}$	$\underline{25.7}_{1.71}$ / $\underline{13.0}_{0.73}$ / $16.5_{0.24}$	$25.8_{1.67}$ / $13.2_{0.53}$ / $16.4_{0.45}$
PURPLE (Ours)	$\underline{38.8}_{0.08}$ / $\underline{22.1}_{0.16}$ / $\underline{26.4}_{0.12}$	**26.2**$_{1.14}$ / **13.2**$_{0.61}$ / **17.5**$_{0.12}$	**27.9**$_{0.04}$ / **14.4**$_{0.08}$ / **17.1**$_{0.04}$
With Llama-3-8B-Instruct (8.03B)			
BM25	$42.2_{0.04}$ / $\underline{24.2}_{0.04}$ / $31.8_{0.08}$	$29.8_{0.73}$ / $14.6_{0.24}$ / $\underline{20.2}_{0.12}$	$\underline{30.9}_{2.00}$ / $\underline{15.3}_{0.82}$ / $\underline{20.8}_{0.41}$
Contriever	$\underline{42.3}_{0.24}$ / $\underline{24.2}_{0.29}$ / $31.7_{0.24}$	$\underline{30.6}_{1.39}$ / **15.2**$_{0.53}$ / $20.0_{0.32}$	$30.1_{2.49}$ / $14.9_{1.02}$ / $19.7_{0.90}$
IC-RALM	$38.3_{0.90}$ / $20.4_{0.78}$ / $29.1_{0.37}$	$25.0_{0.90}$ / $11.4_{1.06}$ / $15.5_{1.92}$	$29.8_{1.27}$ / $13.6_{0.98}$ / $18.3_{0.98}$
REPLUG	$38.2_{0.45}$ / $20.7_{0.33}$ / $28.4_{0.20}$	$20.2_{1.22}$ / $10.8_{0.57}$ / $11.9_{1.27}$	$15.5_{2.04}$ / $8.8_{0.94}$ / $8.8_{1.02}$
RankGPT (Llama-3-8B-Instruct)	$42.2_{0.04}$ / $\underline{24.2}_{0.04}$ / $31.9_{0.04}$	**30.8**$_{1.39}$ / **15.2**$_{0.57}$ / **20.4**$_{0.36}$	$30.5_{2.45}$ / $15.1_{1.02}$ / $20.4_{0.82}$
RankGPT (GPT-5-nano)	**42.5 / 24.5 / 32.1**	28.7 / 14.2 / $\underline{20.2}$	**33.6 / 16.5 / 21.4**
ICR (Llama-3-8B-Instruct)	$41.9_{0.24}$ / $23.9_{0.20}$ / $31.5_{0.16}$	$\underline{30.6}_{1.35}$ / $\underline{15.1}_{0.57}$ / $\underline{20.2}_{0.20}$	$30.7_{1.96}$ / $15.2_{0.78}$ / $20.3_{0.41}$
PURPLE (Ours)	$\underline{42.3}_{0.08}$ / $\underline{24.2}_{0.12}$ / $\underline{32.0}_{0.24}$	$30.4_{1.63}$ / $15.0_{0.73}$ / $19.8_{0.24}$	$30.2_{1.61}$ / $14.9_{0.83}$ / $20.0_{0.94}$

User Persona / History:
- @tommoger thanks tom, and glad it all went well
- @callamKAPOW I tried to change it but the pic was to big and I can't find the old one
- Thinks I can pass English, I just wish I could be as confident about geography

Input Tweet:
Paraphrase the following tweet without any explanation before or after it: @picklesticks6 and I am in the same boat, but at least we attended the math lessons unlike someone we both know who is taking their exam today and feels sad about it. =(

Ground Truth Target:
@picklesticks6 same, but at least we turned up to the maths lessons unlike someone we know whos takeing their exam today =(

ICR (Baseline) Generation:
@picklesticks6 we both missed math classes, and it's making our exam prep tougher, unlike someone who went and feels down about it today. =(

PURPLE (Ours) Generation:
@picklesticks6 feels the same about math exams as me, but at least we went to the lessons, unlike someone else who feels sad today about their exam. =(

Analysis:
This case highlights a critical semantic failure in the baseline. **ICR** generates "we both missed math classes," which directly contradicts the input stating "at least we attended." It effectively swaps the roles of the speaker and the third party. **PURPLE** demonstrates superior natural language understanding by correctly preserving the fact that the speaker *did* attend the lesson. Furthermore, PURPLE better matches the user's persona (which discusses academic confidence in English/Geography) by maintaining the conversational flow without losing the factual premise.

E LLM Usage

During the preparation of this manuscript, we employed large language models (LLMs) to assist with English writing refinement and style polishing. All technical content, including the design of PURPLE, theoretical formulations, experimental setup, and reported results, was conceived, implemented, and validated by the authors. The LLMs were used solely for linguistic improvement and did not contribute to the research methodology or experimental findings.