# Do Implicit Personalization and Explicit Styles Conflict? PsPLUG: A Lightweight Plug-in for Balancing Personalization and Style in Customized LLMs

**Yutong Song**[♣*], **Jiang Wu**[♠*], **Shaofan Yuan**[♢], **Chengze Shen**[♢], **Jian Wang**[♢], **Yu Wang**[♢], **Nikil Dutt**[♣], **Amir M. Rahmani**[♣]

♣ University of California, Irvine ♠ Independent Researcher ♢ TikTok

## Abstract

Personalized large language models are often expected to follow explicit style instructions, yet we find that such instructions can undermine the user-specific characteristics that personalization methods aim to preserve. We call this failure mode personalization collapse: explicit style control can conflict with implicit user preferences. To address this challenge, we propose PsPLUG, a lightweight plug-in that learns a user-specific residual after accounting for the requested style. PsPLUG also allows us to tune personalization strength at inference time. Our experiments show that explicit style instructions can diminish personalization in existing methods, whereas PsPLUG better preserves user preferences while providing precise control over the balance between personalization and style adherence.

## 1 Introduction

Large language models (LLMs) are increasingly deployed in interactive settings where users expect systems not only to produce factually correct content, but also to align with their individual linguistic habits, preferences, and communicative styles (Tan and Jiang, 2023; Zhang et al., 2024). This has sparked rapid progress in personalized generation, including (i) *retrieval-based* methods that fetch user histories into the context window (Salemi et al., 2024b; Kumar et al., 2024; Mysore et al., 2024; Tang et al., 2024), (ii) *per-user fine-tuning* approaches such as user-specific LoRA or adapters (Hu et al., 2021; Houlsby et al., 2019; Tan et al., 2024a,b), and (iii) lightweight plug-in mechanisms that inject user embeddings or soft prompts (Liu et al., 2024; Li and Liang, 2021; Liu et al., 2022). Together, these techniques have demonstrated that LLMs can adapt to a user given enough data or context. (Chen et al., 2024, 2025b; Zhang et al., 2025a) Despite this progress, a critical vulnerability in current personalization pipelines remains largely overlooked. Existing methods typically inject user-related signals without theoretically clarifying what constitutes the *core* personalized signal, or how it relates to the neutral behavior of the base model under the same inputs (Shenfeld et al., 2025; Kim et al., 2025). Concurrently, modern NLP applications increasingly operate under *explicit system instructions*, such as strict stylistic or tonal guidance (e.g., "respond formally", "use a concise tone") (Zhang et al., 2023; Liang et al., 2024; Shanahan et al., 2023). We empirically observe that when such explicit constraints are introduced, existing personalization methods suffer from severe persona degradation. Strong system instructions tend to dominate the generation space, effectively overriding and collapsing diverse dimensions of user-specific traits. This reveals a fundamental challenge: *style-constrained personalization*, where a model must reliably balance explicit task directives with implicit user priors.

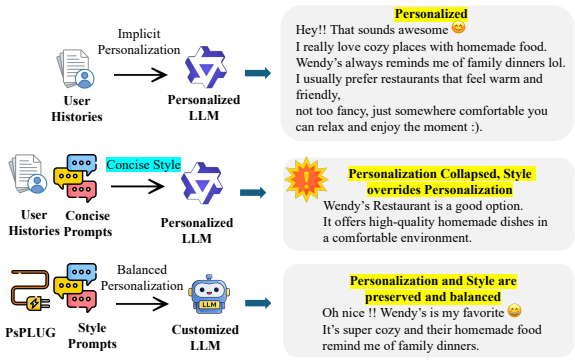


Figure 1: The style-constrained personalization challenge. While standard models capture implicit user personalization, existing methods suffer from severe personalization collapse under explicit style instructions.

To address this, we introduce a novel theoretical perspective: modeling personalization as a *distribu-*

[*]Equal contribution.

The code is available at https://github.com/RainieLLM/PsPLUG.



*Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, pages 938–957
October 24-29, 2026 ©2026 Association for Computational Linguistics

*tional residual*. Rather than learning absolute output likelihoods, we view the persona as the distinct deviation between two conditional distributions under the *identical* input: the user's true linguistic distribution and the neutral distribution of a base LLM. A user-authored response naturally encapsulates personalized lexical and structural preferences, whereas a zero-shot LLM response reflects a generic, population-level prior. Their divergence constitutes the pure personalization residual.

Building on this residual view, we design a style-aware preference objective (Bradley and Terry, 1952; Rafailov et al., 2023). By contrasting user-authored texts against style-conditioned negatives—responses following explicit instructions but lacking user nuances—the model isolates latent persona signals. Unlike standard preference optimization that merely anchors to generic base distributions (e.g., LongPO (Chen et al., 2025a)), our residual reward mathematically disentangles these competing signals to preserve fine-grained persona fidelity under strict stylistic constraints.

We instantiate this methodology as **PsPLUG**, a lightweight soft-prompt module that prepends learned prefix embeddings to a frozen LLM backbone. To achieve a controllable equilibrium between dual constraints, PsPLUG incorporates a unified inference-time scaling mechanism governed by a coefficient ($\alpha$). This grants fine-grained, dynamic adjustment over the trade-off between instruction adherence and personalization strength, enabling scalable adaptation without the prohibitive costs of per-user fine-tuning.

Our work makes three main contributions:

- We propose PsPLUG, a novel framework that effectively balances explicit system instructions with implicit user personalization, addressing the critical issue that existing methods often suffer from severe personalization degradation during text generation, as strong system instructions tend to override and collapse diverse dimensions of user-specific traits.
- We introduce a novel perspective that models personalization as a distributional residual. Building on this view, PsPLUG uses a style-conditioned preference objective to separate user-specific persona signals from dominant instruction and style effects, enabling parameter-efficient personalization without per-user fine-tuning.
- We develop a unified inference-time control mechanism with a scaling coefficient ($\alpha$), which allows fine-grained adjustment of the trade-off between instructions and personalization strength. Experiments on the LaMP benchmark show that PsPLUG consistently outperforms state-of-the-art baselines in preserving persona alignment under strong instructions.

## 2 Method

### 2.1 Task Formulation

Let $x$ denote a task input (e.g., a news headline prompt), $u$ index a user, and $y^u$ be the user-authored response. Let $s$ denote an optional style instruction, and define the full task input as $x_s \triangleq (x, s)$. We assume access to a base LLM $\pi_{\text{ref}}(y \mid x_s)$ that represents a non-personalized, population-level distribution. Our goal is to construct a personalized policy $\pi_\phi(y \mid x_s, u)$ that (i) reproduces user-specific behavior and (ii) can be balanced with explicit task-level style instruction $s$. In our experiments, we consider a fixed set of four predefined style instructions summarized in the Table 1.

### 2.2 PsPLUG Framework

We instantiate the residual view with a lightweight plug-in that injects a compact continuous prefix into a frozen LLM. PsPLUG is trained once and attached to task prompts at inference time. Crucially, because the backbone remains frozen and user signals are decoupled into a modular prefix, our approach enables rapid user context switching without the need to reload model parameters. Furthermore, scaling this prefix provides a continuous *strength control mechanism*, yielding a tunable trade-off between personalization and style instructions.

We denote by $\phi$ all trainable plug-in parameters, including the shared instruction embedding $\theta_{\text{sys}}$ and the parameters of $f_\phi^{\text{pag}}$ and $f_\phi^{\text{qry}}$. The backbone $\pi_{\text{ref}}$ (including its input embedding layer $\text{Emb}_{\text{ref}}$) remains frozen.

Formally, we inject a compact prefix $z_{u,x_s}$ at the input embedding layer. This prefix is the concatenation of three distinct vectors: (i) a trainable system instruction vector $z_{\text{sys}}$; (ii) a user vector $z_u$ derived from the user's history; (iii) an input vector $z_x$ derived from the current prompt. The personalized

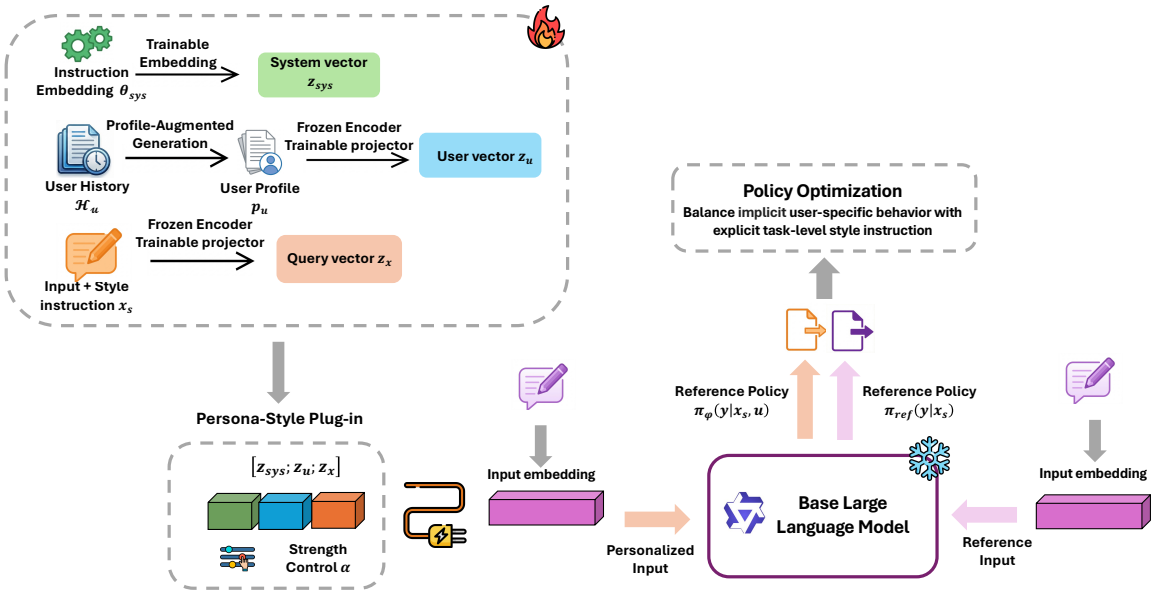


Figure 2: PsPLUG is a lightweight plug-in framework that injects a learnable user-specific prefix into a frozen LLM to enable controllable personalization. The fire logo indicates the trainable model and the snow logo indicates the frozen model.

Table 1: Predefined style instructions used in PsPLUG training and evaluation.

| Style Name | Style Instruction |
|---|---|
| Warm | Please write in a warm, humorous style that uses gentle jokes and soft, uplifting comedy. |
| Critical | Please write in a sharply critical way, directly pointing out flaws or problems and avoiding overly balanced phrasing. |
| Concise | Please write in a concise and formal way, using precise language and avoiding unnecessary elaboration. |
| Elaborative | Please write in a reflective and elaborative way, carefully explaining reasoning with detailed examples and considering multiple perspectives. |

policy is defined as:

$$z_{u,x_s} \triangleq [z_{\text{sys}};\ z_u;\ z_x],$$
$$\pi_\phi(y \mid x_s, u) = \pi_{\text{ref}}\big(y \mid [z_{u,x_s};\ \text{Emb}_{\text{ref}}(x_s)]\big). \tag{1}$$

**System Instruction Embedding.** Besides user- and input-dependent components, PsPLUG includes a shared system instruction embedding that is global across users. This design is inspired by recent studies on instruction tuning (Su et al., 2023; Zhang et al., 2023), which demonstrate that modeling instructions as continuous embeddings can improve an LLM's ability to interpret and follow task-level guidance. We parameterize this component as a trainable embedding $\theta_{\text{sys}} \in \mathbb{R}^d$ and share it across all users:

$$z_{\text{sys}} \triangleq \gamma\, \theta_{\text{sys}}, \tag{2}$$

where $\gamma$ is a constant embedding-scale factor (we set it to match the typical norm of the backbone input embeddings). The shared instruction embedding $\theta_{\text{sys}}$ is optimized jointly with the other plug-in parameters in $\phi$. At inference time, $z_{\text{sys}}$ is prepended to the input embedding sequence, serving as a shared instruction anchor across users.

**Personalization Embedding Projector.** To inject user-specific signals, we convert the user history $\mathcal{H}_u$ into a short fixed Profile-Augmented Generation (PAG) text descriptor $p_u$, and encode it with a frozen sentence encoder $E(\cdot)$:

$$p_u \triangleq \text{PAG}(\mathcal{H}_u), \qquad e_u \triangleq E(p_u). \tag{3}$$

We cache $e_u$ offline and then map $e_u$ into the LLM hidden space through a trainable multi-layer perceptron (MLP) projector:

$$z_u \triangleq \gamma\, f_\phi^{\text{pag}}(e_u), \tag{4}$$

where $\gamma$ is an embedding-scale factor; during training, $f_\phi^{\text{pag}}$ is optimized as part of the plug-in parameters $\phi$, while $E(\cdot)$ and the backbone remain frozen.

**User Query Encoder.** User preferences can manifest differently depending on the current request,

so PsPLUG further includes an input-aware component. Given the current input prompt $x_s$, we obtain a fixed input feature by pooling token embeddings from the frozen backbone:

$$h(x_s) \triangleq \mathrm{Pool}\big(\mathrm{Emb}_{\mathrm{ref}}(x_s)\big). \tag{5}$$

We then map $h(x_s)$ into the hidden space with a trainable MLP encoder:

$$z_x \triangleq \gamma\, f_\phi^{\mathrm{qry}}(h(x_s)). \tag{6}$$

Since the backbone is frozen, $h(x_s)$ is non-trainable; we only optimize the plug-in parameters $\phi$ (including $f_\phi^{\mathrm{qry}}$, $f_\phi^{\mathrm{pag}}$, and $\theta_{\mathrm{sys}}$).

### 2.3 PsPLUG Training: Persona–Style Balancing

Consider a generic input $x$ augmented with a style instruction $s$ (e.g., "write in a concise formal style"), represented as $x_s \triangleq (x, s)$.

This formulation highlights a fundamental tension and raises two questions: (i) Can the model preserve personalization without violating the style instruction? (ii) Can we explicitly *control* the intensity of personalization relative to the style?

**Construction of Style-Conditioned Pairs.** To isolate user-specific signals from generic style following, we construct preference pairs under the style-augmented context $x_s$. We first generate a *style-only* baseline from the model:

$$y^s \sim \pi_{\mathrm{ref}}(\cdot \mid x_s). \tag{7}$$

Since $y^s$ follows $s$ but contains no user-specific information, we treat the user-authored response $y^u$ as the preferred output and form the style-conditioned pair $(y^+, y^-) \triangleq (y^u, y^s)$. This construction ensures that the learning signal focuses on the user residual beyond style compliance.

**Residual Based Training Objective.** PsPLUG therefore models personalization as learning a *residual* between two conditional distributions under the same context $x_s$: the personalized policy $\pi_\phi(\cdot \mid x_s, u)$ and the style-conditioned reference $\pi_{\mathrm{ref}}(\cdot \mid x_s)$. Using the preference pair $(y^+, y^-)$ constructed above, we optimize $\phi$ via a Bradley–Terry (BT) pairwise loss (Bradley and Terry, 1952). For brevity, we omit $(x_s, u)$ in the notation and define the implicit reward score $r_\phi(y)$ as the log-likelihood ratio:

$$r_\phi(y) \triangleq \log \pi_\phi(y \mid x_s, u) - \log \pi_{\mathrm{ref}}(y \mid x_s). \tag{8}$$

The loss function is defined as:

$$\begin{aligned} \ell_{\mathrm{style}}(\phi; x_s, u, y^+, y^-) &= -\log \sigma\big(\beta\, \Delta r_\phi\big), \\ \Delta r_\phi &\triangleq r_\phi(y^+) - r_\phi(y^-). \end{aligned} \tag{9}$$

where $\sigma(\cdot)$ is the sigmoid function and $\beta > 0$ is a temperature hyperparameter. Here, the reference terms anchor the comparison to the baseline style-following behavior induced by $s$ (captured by $\pi_{\mathrm{ref}}$), while the personalized policy is encouraged to deviate only when it better captures the user's specific preferences ($y^u$) over the generic style ($y^s$).

**Inference Time Personalization Strength Control.** Because PsPLUG is modular, we can explicitly regulate the intensity of the user-specific signal during inference without affecting the system instruction or input understanding. We introduce a scaling coefficient $\alpha \geq 0$ specifically for the user vector $z_u$. Let $z_{u,x_s} = [z_{\mathrm{sys}}; z_u; z_x]$ be the learned prefix components. The control mechanism is applied as follows:

$$\begin{aligned} z_{u,x_s}^{(\alpha)} &= [z_{\mathrm{sys}};\ \alpha \cdot z_u;\ z_x], \\ \pi_\phi^{(\alpha)}(y \mid x_s, u) &= \pi_{\mathrm{ref}}\big(y \mid [z_{u,x_s}^{(\alpha)};\ \mathrm{Emb}_{\mathrm{ref}}(x_s)]\big). \end{aligned} \tag{10}$$

By scaling only the user history projector output $z_u$, $\alpha$ serves as a continuous control parameter for personalization strength: as $\alpha \to 0$, the user-specific signal is progressively suppressed, whereas $\alpha > 1$ amplifies the influence of the user's historical personalization.

### 2.4 Special Case: Personalization without Style Instructions

The style-guided formulation in Section 2.3 admits a natural special case when no explicit style instruction is provided. Formally, setting $s = \emptyset$ reduces the conditioning context $x_s = (x, s)$ to the raw input $x$, and the reference policy $\pi_{\mathrm{ref}}(\cdot \mid x_s)$ collapses to a generic baseline as $\pi_{\mathrm{ref}}(\cdot \mid x)$.

**Neutral Baseline and Preference Pairs.** Under this setting, the style-only baseline $y^s$ in Section 2.3 reduces to a *neutral baseline*:

$$y^0 \sim \pi_{\mathrm{ref}}(\cdot \mid x). \tag{11}$$

The preference pair becomes $(y^+, y^-) \triangleq (y^u, y^0)$. This construction isolates the user-specific residual relative to generic population behavior.

**Residual Objective without Style.** Substituting $s = \emptyset$ into the implicit reward score in Eq. 8, we obtain:

$$r_\phi(y) = \log \pi_\phi(y \mid x, u) - \log \pi_{\text{ref}}(y \mid x). \quad (12)$$

The corresponding Bradley–Terry loss is:

$$\begin{aligned} &\ell_{\text{neutral}}(\phi; x, u, y^u, y^0) \\ &\quad = -\log \sigma\Big(\beta\big[r_\phi(y^u) - r_\phi(y^0)\big]\Big). \end{aligned} \quad (13)$$

This special case allows PsPLUG to learn personalization with no style guidance. Gradients update the plug-in parameters $\phi$, while user-dependent information is injected only via $z_u$, which thus serves as the main carrier of the personalization residual relative to $\pi_{\text{ref}}$.

# 3 Experimental Settings

## 3.1 Datasets and Evaluation

We follow the official LaMP benchmark protocol. We report F1 and accuracy for LaMP-1, accuracy and F1 for LaMP-2, MAE and RMSE for LaMP-3, and ROUGE-1 / ROUGE-L / METEOR (Lin, 2004) for LaMP-4, LaMP-5, and LaMP-7. In the style text generation experiments, we consider a fixed set of four predefined style instructions, summarized in Table 1. More task and dataset split details are shown in Appendix A.

In addition to the task metrics, we report two auxiliary scores for controlled customization: *personalization-score* measures alignment with the user's preferences, and *style-score* measures adherence to the given style instruction. Both are computed using LLM-based judges, with human validation on a subset. Details of the judging protocol are provided in Appendix C.

## 3.2 Implementation Details

Across all tasks, we use Qwen/Qwen3-8B (Yang et al., 2025) as the backbone LLM. For each user, we construct a PAG profile via greedy decoding using vLLM (Kwon et al., 2023). The profile text is encoded by a frozen sentence encoder (BGE-base-en-v1.5) using the `[CLS]` representation with $\ell_2$ normalization, and the resulting embeddings are cached offline. For evaluation, we employ GPT-5.2 PRO as an automated LLM judge to assess generation quality. All experiments are conducted on 8 NVIDIA H100 GPUs, and full hyperparameter settings are provided in Appendix A.3.

## 3.3 Baselines

We compare PsPLUG with the following baselines (see Appendix B for implementation details):

**Non-personalized** The LLM generates outputs conditioned solely on the task input, without accessing user history.

**Naive retrieval-based personalization.** We implement a retrieval-augmented baseline that retrieves the top-$k$ user history items via BM25 (Robertson and Zaragoza, 2009) and prepends them to the input as demonstrations.

**State-of-the-Art Personalization.** We include three representative methods: **PAG** (Richardson et al., 2023), **OPPU** (Tan et al., 2024a), and **PPlug** (Persona-Plug) (Liu et al., 2024). These methods align the backbone LLM with user interests using learnable soft prompts or plug-in modules. We match the backbone and decoding configurations for a fair comparison.

# 4 Results and Analysis

**Research Questions** In this section, we present comprehensive experiments, aiming to address the following Research Questions (RQs):

**RQ1:** How does PsPLUG perform on personalization tasks without explicit style instructions compared to existing personalization baselines?

**RQ2:** Can PsPLUG maintain effective user-level personalization when explicit style constraints are introduced in text generation tasks?

**RQ3**: Can PsPLUG achieve a controllable trade-off between stylistic adherence and personalized expression?

**RQ4**: Is PsPLUG a lightweight and efficient plug-in approach, and can it adapt across base LLMs of different model sizes?

## 4.1 Main Results

To answer **RQ1**, we compare PsPLUG with other personalized baselines and results are shown in Table 2. **Consistently outperforms existing personalization baselines on tasks without explicit style instructions.** While a few tasks (e.g., LaMP-5) remain competitive with PPlug, PsPLUG maintains comparable performance overall.

[-1*]Denotes one-time pre-processing. $|P_u|$: size of user history; $H, L$: backbone hidden size and layers; $r$: LoRA rank; $d_e$: embedding dimension. Note that PsPLUG injects fixed vectors, keeping inference overhead constant and independent of $|P_u|$.

Table 2: Main personalization results on LaMP benchmarks. ↑ indicates higher is better, and ↓ indicates lower is better. Best results are in **bold**, and second-best are underlined.

| TASK/methods | Metric | Non-per. | RAG | PAG | PPlug | OPPU | PsPLUG (Ours) |
|---|---|---|---|---|---|---|---|
| **LaMP-1: CITATION ID.** | ACC↑ | 0.518 | 0.441 | 0.562 | 0.563 | 0.556 | **0.584** |
| | F1↑ | 0.448 | 0.397 | 0.481 | 0.493 | 0.556 | **0.589** |
| **LaMP-2M: MOVIE TAGGING** | F1↑ | 0.254 | 0.283 | 0.311 | 0.307 | 0.314 | **0.334** |
| | ACC↑ | 0.375 | 0.381 | 0.393 | 0.382 | **0.425** | 0.392 |
| **LaMP-3: PRODUCT RATING** | MAE↓ | 0.516 | 0.615 | 0.435 | 0.339 | 0.347 | **0.332** |
| | RMSE↓ | 0.805 | 0.981 | 0.714 | 0.583 | 0.613 | **0.464** |
| **LaMP-4: NEWS HEADLINE GEN.** | ROUGE-1↑ | 0.146 | 0.165 | 0.164 | 0.158 | 0.152 | **0.167** |
| | ROUGE-L↑ | 0.128 | 0.144 | 0.146 | 0.138 | 0.128 | **0.148** |
| | METEOR↑ | 0.107 | 0.108 | 0.098 | 0.092 | 0.079 | **0.109** |
| **LaMP-5: SCHOLARLY TITLE GEN.** | ROUGE-1↑ | 0.426 | 0.459 | 0.415 | 0.462 | 0.426 | **0.463** |
| | ROUGE-L↑ | 0.342 | 0.387 | 0.352 | 0.386 | 0.342 | **0.391** |
| | METEOR↑ | 0.360 | **0.401** | 0.375 | 0.399 | 0.393 | 0.398 |
| **LaMP-7: TWEET PARAPHRASE** | ROUGE-1↑ | 0.497 | 0.500 | 0.507 | 0.502 | 0.498 | **0.523** |
| | ROUGE-L↑ | 0.440 | 0.441 | 0.435 | 0.443 | 0.422 | **0.457** |
| | METEOR↑ | 0.324 | **0.337** | 0.329 | 0.332 | 0.327 | **0.337** |

To answer **RQ2**, we compare PsPLUG with other personalized baselines under 4 style prompt and results are shown in Table 3. We have findings as follows.

**PsPLUG maintains strong user-level personalization under explicit style constraints**. It achieves best or second-best performance in over 80% of style–task–metric settings and outperforming prior baselines by up to 2–4 ROUGE points.

**Different styles exhibit varying degrees of interference with personalization.** Tone-oriented styles such as warm and critical tend to disrupt personalization baselines more severely, often leading to noticeable performance degradation. In contrast, **Under the concise constraint, some baselines even improve score** (e.g., Non-pers. baseline R-1 on LaMP-5 increases from 0.426 to 0.441), suggesting that concise is inherently aligned with task structure and user preferences rather than conflicting with them. This behavior suggests that concise primarily functions as a structural constraint on content length and organization, rather than altering tone or sentiment.

Because LaMP benchmarks primarily rely on ROUGE-1 and ROUGE-L, which heavily depend on surface-level lexical overlap, the interaction between style control and personalization becomes inherently entangled and difficult to disentangle. This limitation highlights the necessity of adopting more diverse evaluation criteria to properly assess stylistic adherence and user-level personalization. We additionally employ an LLM-based evaluator and human evaluations to assess style score on LaMP-7 beyond overlap-based metrics, shown in Figure 3. Overall, PsPLUG achieves the highest style scores across all four styles, indicating strong and consistent style adherence. Human and LLM judgments exhibit similar relative trends across styles, particularly for warm and concise. Notably, concise receives the highest style scores for all methods, while critical and elaborative are generally harder to control. We also observe a noticeably larger gap between LLM and human judgments under the elaborative style, indicating that the two evaluators rely on different criteria when assessing this style. Compared to other baselines, **PsPLUG consistently improves style adherence without sacrificing personalization**, supporting its ability to disentangle and jointly model user preference signals and explicit style constraints. To show the difference between styles outputs, we illustrate a case study in Figure 4.

For personalization, LLM-based and human evaluations are also broadly aligned in their relative rankings, as both consistently identify PsPLUG as the strongest or near-strongest method across most styles. This suggests that LLM judgments can serve as a useful proxy for coarse-grained comparison of personalization quality. However, the alignment is not perfect. Human evaluators tend to be more sensitive to subtle affective and user-specific cues, while the LLM evaluator appears to place relatively more emphasis on surface-level stylistic patterns and overall fluency. As a result, the discrepancies become more visible under challenging styles such as *critical* and especially *elaborative*, where pre-

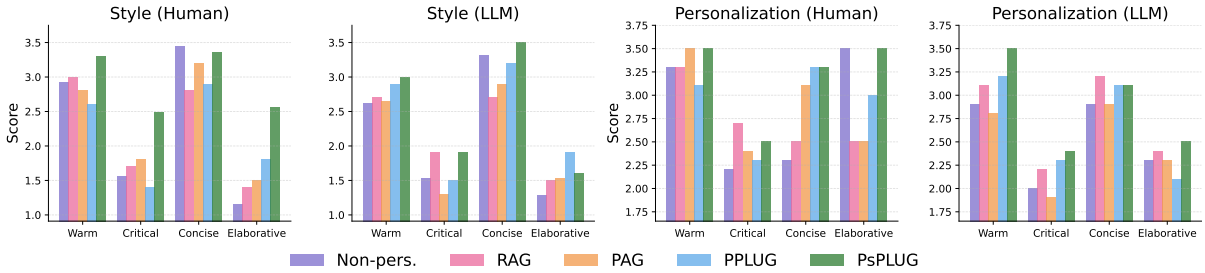


Figure 3: LLM-based judgments and human evaluations for LaMP-7.

Table 3: Personalization results on LaMP tasks under different styles. Best are in **bold** and second-best are underlined.

| Style | Metric | Non-pers. | RAG | PAG | PPlug | PsPLUG |
|---|---|---|---|---|---|---|
| **LaMP-4: News Headline Gen.** | | | | | | |
| warm | R-1 | 0.120 | 0.114 | 0.131 | 0.138 | **0.143** |
| | R-L | 0.102 | 0.099 | 0.114 | 0.119 | **0.124** |
| critical | R-1 | 0.129 | 0.138 | 0.144 | 0.150 | **0.159** |
| | R-L | 0.112 | 0.120 | 0.126 | 0.132 | **0.137** |
| concise | R-1 | 0.140 | **0.167** | 0.139 | 0.143 | 0.149 |
| | R-L | 0.123 | **0.150** | 0.121 | 0.126 | 0.129 |
| elaborative | R-1 | 0.134 | 0.147 | 0.135 | **0.147** | 0.139 |
| | R-L | 0.113 | 0.137 | 0.121 | 0.126 | **0.128** |
| **LaMP-5: Scholarly Title Gen.** | | | | | | |
| warm | R-1 | 0.248 | 0.259 | 0.365 | 0.470 | **0.476** |
| | R-L | 0.196 | 0.202 | 0.295 | 0.402 | **0.403** |
| critical | R-1 | 0.304 | 0.312 | 0.302 | 0.422 | **0.441** |
| | R-L | 0.249 | 0.255 | 0.223 | 0.352 | **0.368** |
| concise | R-1 | 0.441 | 0.464 | 0.458 | **0.471** | 0.465 |
| | R-L | 0.360 | 0.391 | 0.356 | **0.395** | 0.376 |
| elaborative | R-1 | 0.206 | 0.217 | 0.285 | 0.322 | **0.346** |
| | R-L | 0.167 | 0.178 | 0.245 | 0.312 | **0.385** |
| **LaMP-7: Tweet Paraphrasing** | | | | | | |
| warm | R-1 | 0.426 | 0.455 | 0.448 | 0.454 | **0.457** |
| | R-L | 0.368 | 0.400 | 0.392 | 0.396 | **0.402** |
| critical | R-1 | 0.472 | 0.476 | 0.466 | 0.473 | **0.478** |
| | R-L | 0.418 | **0.431** | 0.417 | 0.410 | 0.414 |
| concise | R-1 | 0.475 | 0.472 | 0.451 | **0.487** | **0.487** |
| | R-L | 0.421 | **0.434** | 0.401 | 0.423 | 0.425 |
| elaborative | R-1 | 0.461 | 0.467 | 0.455 | 0.460 | **0.470** |
| | R-L | 0.403 | 0.405 | 0.401 | 0.402 | **0.415** |

serving fine-grained persona traits is harder and the notion of style quality is inherently more subjective.

Overall, these results indicate that **LLM judgments and human judgments are directionally aligned but not fully interchangeable**. They agree well in identifying the strongest methods and the easier versus harder styles, which supports the reliability of LLM-based evaluation for scalable model comparison. At the same time, the remaining gaps suggest that human evaluation is still necessary for validating nuanced interactions between implicit personalization and explicit stylistic control.

## 4.2 Strength sensitivity test

To answer **RQ3**, we conduct experiment and PsPLUG strength sensitivity on all styles, results shown in Figure 5. **Different styles respond differently to strength scaling**, with concise remaining the most stable across strengths, while tone-oriented styles such as warm and elaborative show higher sensitivity. In particular, elaborative degrades sharply at higher strengths on LaMP-5, suggesting that excessive verbosity conflicts with task constraints such as title generation. PsPLUG provides a controllable trade-off via the strength parameter.

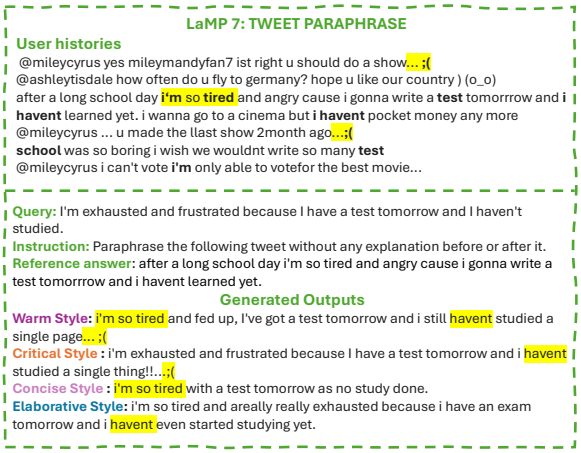


Figure 4: A case study for personalized-style outputs for LaMP-7 using PsPLUG. The yellow background highlights content overlapping with the user history.

## 4.3 Efficiency and scalability

To answer **RQ4**, we test PsPLUG across different model sizes and analyze model efficiency. From Table 4, we find that while larger models generally achieve better ROUGE scores on some tasks such as LaMP-5 and LaMP-7, medium-sized models often perform competitively and even outperform larger models on certain metrics.

Table 5 compares the personalization overhead. RAG incurs storage costs scaling with history size $O(|P_u|)$ and increased latency due to processing retrieved contexts. PEFT necessitates per-user op-

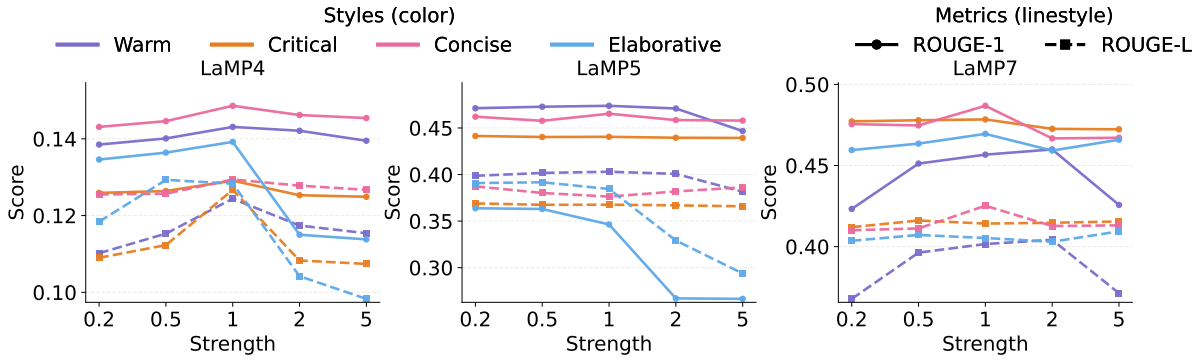


Figure 5: Personalization performance conducted at different strengths and styles.

Table 4: Effect of base model size on LaMP tasks.

| Task | Metric | 4B | 8B | 32B |
|---|---|---|---|---|
| LaMP-4 | ROUGE-1 | 0.145 | **0.162** | 0.148 |
| | ROUGE-L | 0.127 | **0.144** | 0.130 |
| | METEOR | 0.081 | **0.098** | 0.085 |
| LaMP-5 | ROUGE-1 | 0.425 | 0.463 | **0.474** |
| | ROUGE-L | 0.389 | 0.390 | **0.417** |
| | METEOR | 0.366 | 0.321 | **0.382** |
| LaMP-7 | ROUGE-1 | 0.483 | 0.513 | **0.529** |
| | ROUGE-L | 0.424 | 0.457 | **0.460** |
| | METEOR | 0.308 | **0.337** | 0.308 |

Table 5: Per-user efficiency comparison (personalization overhead) for PsPLUG.

| Metric | RAG | PEFT | PsPLUG |
|---|---|---|---|
| Training Time/User | $O(\|P_u\|)^*$ | $O(\|P_u\|)$ | $O(k)^*$ |
| Latency/Query | $O(\|P_u\|)$ | $O(\text{Load} + \text{Merge})$ | $O(H(d_e{+}H))$ |
| Storage/User | $O(\|P_u\| \cdot d_e)$ | $O(rHL)$ | $O(d_e)$ |

timization and introduces adapter-loading latency during multi-user serving. In contrast, PsPLUG employs a one-time setup to encode the user profile into a single static embedding. At inference, this vector is prepended to input embedding sequence. Consequently, PsPLUG maintains a constant prompt length, ensuring computational cost remains independent of the user history size $|P_u|$.

# 5 Related Works

## 5.1 Personalized LLMs

A direct route to personalization is to fine-tune a model on user-specific data so that user preferences are internalized in the generation distribution. This paradigm has long been explored in personalized dialogue and generation, including persona-grounded dialogue agents (Zhang et al., 2018; Mazaré et al., 2018) and user-conditioned generation tasks (Li and Tuzhilin, 2019; Majumder et al., 2019; Jaech and Ostendorf, 2018). In the LLM era, recent work revisits training-based personalization with richer user supervision signals, e.g., learning from post-deployment user feedback or edits (Madaan et al., 2022; Dalvi Mishra et al., 2022; Gao et al., 2024). Preference-optimization pipelines further reduce the reliance on expensive manual labels by mining implicit user preferences: PUGC converts user-generated content into scalable preference pairs for objectives such as DPO (Tan et al., 2025), while DPL argues that inter-user differences are the key personalization signal and explicitly learns from contrastive comparisons across users (Qiu et al., 2025). But training-based approaches often incur non-trivial per-user adaptation cost, and they typically do not characterize how latent persona should compose with explicit style instructions.

## 5.2 Retrieval-based Personalized LLMs

Retrieval-based personalization augments frozen backbones by injecting user context. This paradigm is systematized by LAMP, which validates retrieving top-$k$ history for diverse tasks, and LONGLAMP, which extends evaluation to long-form generation. Beyond standard IR primitives (Robertson and Zaragoza, 2009; Izacard et al., 2021), recent work improves context quality via downstream signal optimization (Salemi et al., 2024a) and history profiling (Zhong et al., 2022; Liu et al., 2023a). While attractive for its training-free nature, retrieval is constrained by latency and context length, motivating rigorous comparisons with parametric alternatives (Balaguer et al., 2024; Salemi and Zamani, 2025).

## 5.3 PEFT for Personalization

Parameter-efficient fine-tuning (PEFT) (Woźniak et al., 2024) personalizes shared backbones via

lightweight user-specific modules. Recent approaches scale per-user adaptation through modular composition (Zhuang et al., 2024), hierarchical grouping (Zhang et al., 2025b; Song et al., 2026), and instance-wise or continual low-rank adaptation (Zhu et al., 2024; Kong et al., 2024), alongside post-hoc model merging (Jang et al., 2023). Orthogonally, decoding-time methods steer frozen models without parameter updates. But existing formulations often under-specify the disentanglement of core personalized signals from neutral references and lack inference-time control over persona-task composition.

## 6 Conclusion

We present a unified framework that jointly models persona and style by framing personalization as a residual deviation from a neutral policy, rather than fitting a user-specific output distribution. We introduce PsPLUG, a lightweight soft-prompt plug-in enabling scalable persona injection without per-user fine-tuning. By integrating an inference scalar $\alpha$, our method balances personalization preservation and style adherence. Experiments on LaMP demonstrate superior alignment and strong compliance with minimal inference overhead.

## Limitations

In this study, we propose a framework that decouples personalization signals from raw text. We acknowledge several limitations in this work for further exploration and investigation. While our approach effectively separates style from content, we primarily construct style conditions based on four predefined categories (Warm, Critical, Concise, Elaborative). However, real-world stylistic demands are often more fine-grained, open-ended, and compositional. Furthermore, our experiments are currently limited to specific backbone models within the LaMP benchmark distribution; the robustness of our approach across different model architectures, multilingual settings, and diverse long-context domains remains to be verified. Ultimately, we regard this work as a preliminary step that poses the critical question of disentangling personalization from style. We focused on specific textual attributes, yet valid decoupling involves broader dimensions, and we hope this study inspires the community to investigate disentanglement across more complex axes in the future.

## 7 Ethical Considerations

The benchmarks utilized in this study, LaMP and LongLaMP, are publicly accessible and strictly anonymized, ensuring that no personally identifiable information (PII) is exposed. We adhered to standard data protocols, obtaining all datasets via official APIs without the use of proprietary or non-open-source data. While personalized generation paradigms inherently necessitate access to user history, our proposed framework is designed to mitigate these concerns through architectural decoupling. Specifically, our approach separates high-level personalization signals from raw textual data. This allows user representations to be constructed locally on the client side, requiring only the transmission of lightweight personalization modules rather than sensitive historical logs. Consequently, compared to retrieval-augmented generation or user-specific fine-tuning strategies, our design significantly minimizes the risk of data leakage and adheres to ethical research standards.

## References


Angels Balaguer, Vinamra Benara, Renato Luiz de Freitas Cunha, Roberto de M. Estevão Filho, Todd Hendry, Daniel Holstein, Jennifer Marsman, Nick Mecklenburg, Sara Malvar, Leonardo O. Nunes, Rafael Padilha, Morris Sharp, Bruno Silva, Swati Sharma, Vijay Aski, and Ranveer Chandra. 2024. RAG vs. fine-tuning: Pipelines, tradeoffs, and a case study on agriculture. *arXiv:2401.08406*.

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.

Guanzheng Chen, Xin Li, Michael Qizhe Shieh, and Lidong Bing. 2025a. Longpo: Long context self-evolution of large language models through short-to-long preference optimization. *Preprint*, arXiv:2502.13922.

Jiangjie Chen, Xintao Wang, Rui Xu, Siyu Yuan, Yikai Zhang, Wei Shi, Jian Xie, Shuang Li, Ruihan Yang, Tinghui Zhu, Aili Chen, Nianqi Li, Lida Chen, Caiyu Hu, Siye Wu, Scott Ren, Ziquan Fu, and Yanghua Xiao. 2024. From persona to personalization: A survey on role-playing language agents. *Preprint*, arXiv:2404.18231.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. 2025b. Pad: Personalized alignment of llms at decoding-time.

Bhavana Dalvi Mishra, Oyvind Tafjord, and Peter Clark. 2022. Towards teachable reasoning systems: using

a dynamic memory of user feedback for continual system improvement. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. 2024. Aligning llm agents by learning latent preference from user edits. *arXiv:2404.15269*.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *Preprint*, arXiv:1902.00751.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Tiancheng Hu and Nigel Collier. 2024. Quantifying the persona effect in llm simulations. *Preprint*, arXiv:2402.10811.

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*.

Aaron Jaech and Mari Ostendorf. 2018. Personalized language model for query auto-completion. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 700–705.

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:2310.11564*.

Minbeom Kim, Kang il Lee, Seongho Joo, Hwaran Lee, Thibaut Thonet, and Kyomin Jung. 2025. Drift: Decoding-time personalized alignments with implicit user preferences.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *Preprint*, arXiv:2405.01535.

Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. 2024. Customizing language models with instance-wise lora for sequential recommendation. In *NeurIPS*.

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. *arXiv:2407.11016*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention.

Pan Li and Alexander Tuzhilin. 2019. Towards controllable and personalized review generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3237–3245.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. 2024. Controllable text generation for large language models: A survey. *arXiv:2408.12599*.

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.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2024. Llms + persona-plug = personalized llms. *arXiv:2409.11901*.

Shuai Liu, Hyundong J. Cho, Marjorie Freedman, Xuezhe Ma, and Jonathan May. 2023a. Recap: Retrieval-enhanced context-aware prefix encoder for personalized dialogue response generation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: Nlg evaluation using gpt-4 with better human alignment. *Preprint*, arXiv:2303.16634.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve gpt-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. 2019. Generating personalized recipes from historical user preferences. In

*Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5976–5982.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779.

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 *CustomNLP4U*.

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 ACL*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36.

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:2310.20081*.

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, Surya Kallumadi, and Hamed Zamani. 2024a. Optimization methods for personalizing large language models through retrieval augmentation. In *SIGIR*.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024b. Lamp: When large language models meet personalization. In *ACL*.

Alireza Salemi and Hamed Zamani. 2025. Comparing retrieval-augmentation and parameter-efficient fine-tuning for privacy-preserving personalization of large language models. In *Proceedings of the 2025 International ACM SIGIR Conference on Innovative Concepts and Theories in Information Retrieval (ICTIR)*, pages 286–296.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role-play with large language models. *Preprint*, arXiv:2305.16367.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. 2025. Language model personalization via reward factorization. *arXiv:2503.06358*.

Yutong Song, Jiang Wu, Weijia Zhang, Chengze Shen, Shaofan Yuan, Weitao Lu, Jian Wang, Yu Wang, Nikil Dutt, and Amir M Rahmani. 2026. Card: Cluster-level adaptation with reward-guided decoding for personalized text generation. *arXiv preprint arXiv:2601.06352*.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1102–1121, Toronto, Canada. Association for Computational Linguistics.

Zhaoxuan Tan and Meng Jiang. 2023. User modeling in the era of large language models: Current research and future directions. *arXiv:2312.11518*.

Zhaoxuan Tan, Zheng Li, Tianyi Liu, Haodong Wang, Hyokun Yun, Ming Zeng, Pei Chen, Zhihan Zhang, Yifan Gao, Ruijie Wang, Priyanka Nigam, Bing Yin, and Meng Jiang. 2025. Aligning large language models with implicit preferences from user-generated content. In *ACL*.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024a. Democratizing large language models via personalized parameter-efficient fine-tuning. In *EMNLP*.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Personalized pieces: Efficient personalized large language models through modular and collaborative peft. In *EMNLP*.

Xiangru Tang, Xingyao Zhang, Yanjun Shao, Jie Wu, Yilun Zhao, Arman Cohan, Ming Gong, Dongmei Zhang, and Mark Gerstein. 2024. Step-back profiling: Distilling user history for personalized scientific writing. *arXiv:2406.14275*.

Stanisław Woźniak, Bartłomiej Koptyra, Arkadiusz Janz, Przemysław Kazienko, and Jan Kocoń. 2024. Personalized large language models. *arXiv:2402.09269*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jinghao Zhang, Yuting Liu, Wenjie Wang, Qiang Liu, Shu Wu, Liang Wang, and Tat-Seng Chua. 2025a. Personalized text generation with contrastive activation steering. *arXiv:2503.05213*.

Li Zhang, Haotian Liu, Yuxin Wang, and Maosong Sun. 2025b. Proper: A progressive learning framework for personalized large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*. Association for Computational Linguistics.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Zhehao Zhang, Ryan A. Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, Ruiyi Zhang, Jiuxiang Gu, Tyler Derr, Hongjie Chen, Junda Wu, Xiang Chen, Zichao Wang, Subrata Mitra, Nedim Lipka, and 2 others. 2024. Personalization of large language models: A survey. *arXiv preprint*, arXiv:2411.00027.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

Hanxun Zhong, Zhicheng Dou, Yutao Zhu, Hongjin Qian, and Ji-Rong Wen. 2022. Less is more: Learning to refine dialogue history for personalized dialogue generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5808–5820.

Jiachen Zhu, Jianghao Lin, Xinyi Dai, Bo Chen, Rong Shan, Jieming Zhu, Ruiming Tang, Yong Yu, and Weinan Zhang. 2024. Lifelong personalized low-rank adaptation of large language models for recommendation. *arXiv:2408.03533*.

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. *arXiv:2406.02888*.

## Appendix Contents

# A Dataset Statistics and Task Details

Detailed statistics for the six tasks are provided in Table 6. The formats of input, output, and user histories of the tasks are shown in Table 7.

Table 6: Data statistics of the six experimented tasks in the LaMP benchmark.

| Task | Type | Train | Val | In Len. | Out Len. | Hist. | #Cls |
|---|---|---|---|---|---|---|---|
| LaMP-1 | Binary | 6,542 | 1,500 | $51.4 \pm 5.7$ | – | $84.1 \pm 47.5$ | 2 |
| LaMP-2 | Category | 5,073 | 1,410 | $92.4 \pm 21.9$ | – | $86.8 \pm 189.5$ | 15 |
| LaMP-3 | Ordinal | 20,000 | 2,500 | $128.2 \pm 146.2$ | – | $185.4 \pm 129.3$ | 5 |
| LaMP-4 | Gen | 12,500 | 1,500 | $30.0 \pm 12.1$ | $10.1 \pm 3.1$ | $204.6 \pm 250.7$ | – |
| LaMP-5 | Gen | 14,682 | 1,500 | $162.3 \pm 65.6$ | $9.7 \pm 3.2$ | $87.9 \pm 53.6$ | – |
| LaMP-7 | Gen | 13,437 | 1,498 | $29.7 \pm 7.0$ | $17.0 \pm 5.7$ | $15.7 \pm 14.8$ | – |

Table 7: Format of input, output, and user history.

| Task | Input | Output | User History |
|---|---|---|---|
| LaMP-4 | Gen headline: {*article*} | How I Got 'Rich' | title: {*title*}<br>text: {*article*} |
| LaMP-5 | Gen title for abstract: {*abstract*} | Distributed Partial Clustering | title: {*title*}<br>text: {*abstract*} |
| LaMP-7 | Paraphrase tweet: {*tweet*} | gotta make the most of my last day | text: {*tweet*} |

## A.1 Persona Score evaluation prompt

**Persona Score evaluation prompt**

Task Description: You are given: (1) an instruction (which may include an input), (2) a response to evaluate, (3) a reference answer that should receive a score of 5, (4) a user profile containing the user's preferences and background, and (5) a score rubric describing the evaluation criteria.
Your task is to:
1. Write detailed feedback that assesses how well the response is personalized to this specific user, strictly following the given score rubric. Do **not** comment on general quality unrelated to personalization. 2. Carefully consider how the response aligns with the user's preferences, interests, and background information in the user profile. 3. After writing the feedback, output a single integer score between 1 and 5 that best matches the rubric. You **must** choose an integer. 4. The output format must be: "(write a feedback for criteria) [RESULT] (an integer number between 1 and 5)" 5. Do not add any extra opening, closing, or explanations.
The instruction to evaluate: instruction
Response to evaluate: response
Reference Answer : reference answer
User Profile: user profile
Score Rubric: rubric
Feedback:

## A.2 Score Rubric

**Score Rubric**

criteria: "Evaluate how well the response is personalized to the specific user."
score1 description: "Generic or impersonal. Ignores the provided user profile and personality cues. Style does not match the user; may feel robotic, off-topic, or contradict stated preferences. No meaningful use of user details; largely boilerplate."
score2 description: "Minimal personalization. Mentions a profile detail superficially but remains mostly generic. Weak style match; limited relevance to the user's interests or situation. Includes filler or distracting disclaimers. Significant deviation from the reference's intent or emphasis."
score3 description: "Basic personalization. References a few relevant details and partially adapts to them. Generally on topic but misses important user nuances (interests, constraints, or personality cues). Moderate similarity to the reference; may be verbose or somewhat generic."
score4 description: "Good personalization. Integrates multiple user details accurately; content is relevant and helpful. Tone mostly matches the user's personality and preferred style. Clear and engaging, with only minor misses compared to the reference."
score5 description: "Excellent personalization. Seamlessly weaves in key profile details; highly relevant and tailored guidance or conversation. Tone closely matches the user's personality—empathetic, engaging, and concise. Avoids boilerplate and unnecessary disclaimers. Very closely aligned with the user's likely preference as indicated by the reference."

## A.3 Hyperparameter Settings

| Component | Setting |
|---|---|
| Backbone LLM | Qwen3 (frozen) |
| Sentence Encoder | BGE-base-en-v1.5 (frozen) |
| PAG Profile Generation | vLLM, greedy decoding |
| Decoding Temperature | 0.0 |
| History Sampling ($k$) | 10 |
| Encoder Max Length | 4096 |
| Prefix Injection | Input embedding layer (`inputs_embeds`) |
| Prefix Length | 3 tokens (system, user, query) |
| Embedding Normalization | $\ell_2$ normalization |
| Training Epochs | 5 (early stopping) |
| Learning Rate | $1 \times 10^{-4}$ |
| Batch Size | Task-dependent |
| Precision | bfloat16 |
| Inference Strategy | Greedy decoding |
| Hardware | 8 × NVIDIA H100 GPUs |

Table 8: Hyperparameter settings used in all experiments.

# B Detailed Baseline Implementations

In this section, we provide the exact implementation protocols, hyperparameter configurations, and prompt templates used for all baseline methods. All experiments were conducted on 8 NVIDIA H100 GPUs using PyTorch with `bfloat16` precision to ensure numerical stability.

## B.1 Non-Personalized Zero-Shot (Zero-Shot)

The Zero-Shot baseline evaluates the backbone model's intrinsic ability to follow task and style instructions without user-specific context. We employ Qwen3-8B as the backbone model. To maximize inference throughput, we utilize the `vLLM` library for serving.

**Prompt Construction.** The input to the model is constructed by wrapping the style instruction and the task input within the model's standard chat template. Based on our experimental code, the prompt structure is defined as follows:

- **Style Instruction ($I_{style}$):** We prepend a natural language instruction to control the generation style. For example, for the *Concise* style, the instruction is:

  ```
  "Please write in a concise and formal way, using precise language and avoiding unnecessary elaboration:"
  ```

- **Task Template:** The task-specific instruction is concatenated immediately after the style prompt. For the LaMP-7 (Tweet Paraphrasing) task, the template is:

  ```
  {I_{style}} Paraphrase the following text into tweet without any explanation before or after it: {article}
  ```

This combined string is then processed by the tokenizer's `apply_chat_template` function with `add_generation_prompt=True`.

**Decoding Configuration.** To ensure reproducibility and minimize variance in the baseline, we use **Greedy Decoding**. The specific sampling parameters derived from our implementation are:

- **Temperature:** $0.0$ (Deterministic generation).
- **Top-p:** $1.0$.

## B.2 Retrieval-Augmented Generation (RAG)

Our RAG baseline employs a sparse retrieval mechanism to inject relevant historical examples into the context window. We prioritize a budget-aware context construction strategy to handle variable lengths of user history.

**Retrieval Mechanism (BM25).** We implement a custom **Okapi BM25** retriever.

- **Tokenization:** We use a regex-based tokenizer `r"\w+"` combined with lowercasing. This lightweight tokenization ensures robust matching for English text without the overhead of heavy NLP pipelines.
- **Hyperparameters:** We use standard BM25 parameters: $k_1 = 1.5$ and $b = 0.75$.
- **Query Formulation:** The query is derived strictly from the current task input (e.g., the article to be summarized). We intentionally exclude the style instruction from the retrieval query to avoid retrieving documents based on generic style keywords (like "formal" or "concise") rather than content relevance.
- **Scoring:** For a user profile $\mathcal{H}_u$, we compute the BM25 score for every historical document. If the maximum score is $\leq 0$ (indicating no lexical overlap), the system falls back to the Zero-Shot behavior to avoid injecting noise. We retrieve the top-$K = 4$ documents.

**Budget-Aware Prompt Construction.** A critical challenge in RAG is fitting multiple historical examples within a fixed context window. We implement a dynamic truncation strategy:

1. **Fixed Overhead Calculation:** We first calculate the token usage of fixed components, including the system instructions (e.g., `"Following the given patterns"`), the style prefix, and special control tokens (e.g., `/no_think` to suppress reasoning traces in QWEN reasoning models).

2. **Dynamic Allocation:** The remaining token budget is distributed evenly among the $K$ retrieved documents.

3. **Truncation:** For each retrieved example (consisting of a title and abstract), we truncate the abstract to fit the allocated slot while preserving the full title.

**In-Context Learning Template.** The retrieved examples $\{d_1, \ldots, d_K\}$ are formatted as few-shot demonstrations. The prompt provided to the model follows this schema:

```
"{title_1}" is the title for "{abstract_1}" , and ... "title_K" is the title for "abstract_K". Following the given patterns {Style_Instruction} {Input}
```

This format explicitly instructs the model to observe the mapping pattern in the history before processing the current input.

### B.3 Profile-Augmented Generation (PAG)

PAG addresses the context limitation of RAG by compressing the user history into a natural language profile. Following the protocols in LaMP (Salemi et al., 2024b), we design task-specific prompts to extract the most relevant stylistic or content preferences for each domain.

**Offline Profile Generation.** We employ a "summarize-then-generate" pipeline using an instruction-tuned LLM (Qwen3-8B-Instruct). Since user preferences manifest differently across tasks (e.g., formatting style for citations vs. tonal style for tweets), we utilize distinct prompts for each dataset. The process involves two steps:

1. **History Formatting:** We retrieve a set of historical examples from the user's corpus and format them into a structured string (see "History Item Format" in Table 9).

2. **Profile Extraction:** We feed these formatted examples into the summarizer using a task-specific instruction to generate the profile $p_u$.

The exact prompts and formatting templates for all tasks are detailed in Table 9.

**Inference Integration.**

### B.4 Personalized Plug-in (PPlug)

The PPlug baseline represents the standard plugin-based personalization approach. It shares the continuous embedding architecture with our PsPLUG but differs fundamentally in training objectives.

**Architecture.** We implement PPlug using a **History Encoder** and a **Projector**.

- **Encoder:** We use a frozen BERT-base model (Devlin et al., 2019) to encode raw historical texts into dense vectors $\{h_i\}$.

- **Attention Aggregator:** An input-aware attention mechanism computes a weighted sum of history vectors: $e_u = \sum \alpha_i h_i$, where $\alpha_i \propto \exp(h_i^\top W q)$ and $q$ is the query vector of the current input.

- **Projector:** A two-layer MLP maps the aggregated dimension (768) to the LLM's embedding dimension (4096 for Qwen3-8B).

**Training Details.**

- **Objective:** PPLUG is trained using standard Causal Language Modeling (CLM) loss: $\mathcal{L}_{CLM} = -\log P(y^u|x, z_u)$. Crucially, it does not employ the preference optimization (DPO) or the style-balancing residual loss used in PsPLUG.

- **Optimization:** We train for 3 epochs using the AdamW optimizer with a learning rate of $1e-4$ for the projector and encoder adapter. The backbone LLM remains entirely frozen.

- **Batch Size:** We use a global batch size of 128.

### B.5 One-PEFT-Per-User (OPPU)

OPPU serves as the theoretical upper bound for personalization fidelity, where we train a separate adapter for every single user.

| Task | History Item Format | Profile Generation Prompt |
|---|---|---|
| **Tweet Paraphrase** (LaMP-7) | `tweet: {text}` | Given this person's previous tweets, try to describe a template for their tweets. I want to take a generic sentence and rephrase it to sound like one of their tweets, with the same style/punctuation/capitalization/wording/tone/etc. as them. Only give me the template description, nothing else. User History: {} Answer: |
| **News Headline** (LaMP-4) | `article: {text}`<br>`headline: {title}` | Given this author's previous articles, try to describe a template for their headlines. I want to be able to accurately predict the headline gives one of their articles. Be specific about their style and wording, don't tell me anything generic. User History: {} Answer: |
| **Scholarly Title** (LaMP-5) | `abstract: {abstract}`<br>`title: {title}` | Given this author's previous publications, try to describe a template for their titles. I want to be able to accurately predict the title of one of the papers from the abstract. Only generate the template description, nothing else. User History: {} Answer: |
| **Citation** (LaMP-1) | `paper title: {title}`<br>`reference: {citation}` | Write a summary, in English, of the research interests and topics of a researcher who has published the following papers. Only generate the summary, no other text. User History: {} Answer: |
| **Movie Tagging** (LaMP-2 M) | `description: {description}`<br>`tag: {tag}` | Look at the following past movies this user has watched and determine the most popular tag they labeled. Answer in the following form: most popular tag: <tag>. User History: {} Answer: |
| **Product Rating** (LaMP-3) | `review: {text}`<br>`score: {score}` | Based on this user's past reviews, what are the most common scores they give for positive and negative reviews? Answer in the following form: most common positive score: <pos>, most common negative score: <neg>. User History: {} Answer: |

Table 9: Task-specific prompts and history formatting templates used for offline profile generation in the PAG baseline. The `{}` slot in the *Profile Generation Prompt* is populated with multiple history items formatted according to the *History Item Format* column.

**Implementation (LoRA).** We utilize Low-Rank Adaptation (LoRA) to efficiently fine-tune per-user parameters.

- **Rank Configuration:** We set the LoRA rank $r = 8$ and alpha $\alpha = 16$.
- **Target Modules:** Adapters are attached to the query ($W_q$) and value ($W_v$) projection matrices of the attention layers.
- **Base Model Adaptation:** Before user-specific training, we first perform instruction tuning on the generic training set (all users pooled) for 1 epoch. This yields a task-adapted base model $\pi_{\text{base}}$.

**User-Specific Optimization.** For each user $u$, we initialize a fresh set of LoRA weights $\Delta\theta_u$.

- **Training Data:** We use the user's historical input-output pairs $\{(x_i, y_i)\}_{i=1}^{N_u}$.
- **Hyperparameters:** Each user-specific adapter is fine-tuned for 5 epochs with a learning rate of $5e-4$ and a batch size of 4 (with gradient accumulation to effective batch size 16).
- **Storage:** We save the LoRA weights for each user and dynamically load them during inference based on the user ID.

As discussed in the main text, OPPU is evaluated only in the no-style setting ($s = \emptyset$) because the user-specific parameters tightly overfit the historical style, making the model unresponsive to conflicting style instructions.

## C System Prompts

To automate the evaluation of style alignment, we utilize specific system prompts designed for the judge LLM. Each prompt consists of a task description, the input instruction, the generated response, a reference answer (gold standard), and a detailed scoring rubric tailored to the specific target style (Hu and Collier, 2024; Zheng et al., 2023; Liu et al., 2023b; Kim et al., 2024). The prompts explicitly instruct the judge to focus strictly on style matching rather than factual correctness.

Below, we detail the specific prompts and rubrics used for the four evaluated styles: *Warm*, *Concise* , *Critical*, and *Elaborative*.

### C.1 Warm and Humorous Prompt

This prompt evaluates the model's ability to adopt a friendly persona. It emphasizes the use of gentle jokes and a soft, uplifting tone, penalizing robotic or overly serious responses.

**Evaluation Prompt: Warm and Humorous**

**Task Description:** An instruction and a response to evaluate, a reference answer that can get a score of 5, and a score rubric representing an evaluation criteria are given. You are evaluating how well the response matches the following target style: *"Please write in a warm, humorous style that uses gentle jokes and a soft, uplifting comedy way."*
1. Write a detailed feedback that assesses the quality of the response strictly based on the given score rubric, focusing only on style match, not on factual correctness or task success in general. 2. When evaluating, compare the response's tone, phrasing, and structure against the target style description and the reference answer. 3. After writing a feedback, write a score that is an integer between 1 and 5. You should refer to the score rubric. 4. The output format should look as follows: "(write a feedback for criteria) [RESULT] (an integer number between 1 and 5)" 5. Please do not generate any other opening, closing, and explanations.
**The instruction to evaluate:** {{ instruction }}
**Response to evaluate:** {{ response }}
**Reference Answer:** {{ reference_answer }}
**Score Rubrics:**

- **Criteria:** Evaluate how well the response matches a warm, humorous style that uses gentle jokes and soft, uplifting comedy.
- **Score 1 (Severe mismatch):** Tone is cold, flat, or overly serious. No meaningful humor or warmth. May sound robotic, harsh, or purely informational. Does not resemble gentle, uplifting comedy.
- **Score 2 (Minimal alignment):** One or two mild attempts at warmth or humor, but overall tone remains neutral, dry, or formal. Jokes (if any) feel forced or out of place. The response does not consistently feel warm or uplifting.
- **Score 3 (Basic alignment):** Recognizably warm and somewhat humorous, but uneven. Some parts feel friendly and lightly funny, while others are neutral or generic. Humor may be sparse or occasionally awkward, yet the overall tone is not cold.
- **Score 4 (Good alignment):** Tone is consistently warm and friendly, with several gentle jokes or light comedic touches. Humor is generally appropriate and uplifting, with only minor moments that feel flat, generic, or slightly off-style.
- **Score 5 (Excellent alignment):** Strongly and consistently warm, empathetic, and genuinely funny in a gentle way. Jokes are soft, uplifting, and well-integrated into the explanation. No harshness, sarcasm, or coldness. The style closely matches an ideal warm, humorous response.

**Feedback:**

### C.2 Concise and Formal Prompt

This prompt assesses the model's capability to produce professional, objective text. The scoring criteria prioritize precision, technical accuracy, and the elimination of unnecessary elaboration or colloquialisms.

**Evaluation Prompt: Concise and Formal**

**Task Description:** An instruction and a response to evaluate, a reference answer that can get a score of 5, and a score rubric representing an evaluation criteria are given. You are evaluating how well the response matches the following target style: *"Please write in a concise and formal way, using precise language and avoiding unnecessary elaboration."*
1. Write a detailed feedback that assesses the quality of the response strictly based on the given score rubric, focusing only on style match, not on factual correctness or task success in general. [... Steps 2-5 same as above ...]
**The instruction to evaluate:** {{ instruction }}
**Response to evaluate:** {{ response }} **Reference Answer :** {{ reference_answer }}

**Score Rubrics:**

- **Criteria:** Evaluate how well the response matches a concise and formal style that uses precise language and avoids unnecessary elaboration.
- **Score 1 (Severe mismatch):** Tone is casual, chatty, or overly emotional. Language is imprecise or slangy. Response is rambling, disorganized, or much longer than needed. Little to no sense of formality.
- **Score 2 (Minimal alignment):** Some formal wording appears, but overall tone is still somewhat casual or conversational. The response contains noticeable redundancy or digressions. Explanations are longer than necessary and only partially precise.
- **Score 3 (Basic alignment):** Generally formal in tone, but may include occasional casual phrases or mild redundancy. Explanations are understandable yet could be more succinct. Some sentences are precise, but others are wordy or loosely structured.
- **Score 4 (Good alignment):** Tone is clearly formal and professional. Language is mostly precise, with only minor verbosity or repetition. Structure is clear and focused, and the response is reasonably brief while still covering the key points.
- **Score 5 (Excellent alignment):** Tone is consistently formal and objective. Language is precise, technically accurate, and free of slang. The response is concise, well-structured, and free of unnecessary elaboration while still being complete and easy to follow.

**Feedback:**

### C.3 Sharply Critical Prompt

The prompt below measures the response's alignment with a critical and direct persona. It specifically rewards the direct identification of flaws and discourages "hedging" or overly balanced, diplomatic phrasing.

**Evaluation Prompt: Sharply Critical**

**Task Description:** An instruction and a response to evaluate, a reference answer that can get a score of 5, and a score rubric representing an evaluation criteria are given. You are evaluating how well the response matches the following target style: *"Please write in a sharply critical way, directly point out flaws or problems and avoid overly balanced phrasing."*
1. Write a detailed feedback that assesses the quality of the response strictly based on the given score rubric, focusing only on style match, not on factual correctness or task success in general. [... Steps 2-5 same as above ...]
**The instruction to evaluate:** {{ instruction }}
**Response to evaluate:** {{ response }} **Reference Answer :** {{ reference_answer }}
**Score Rubrics:**

- **Criteria:** Evaluate how well the response matches a sharply critical style that directly points out flaws or problems and avoids overly balanced phrasing.
- **Score 1 (Severe mismatch):** Tone is neutral, vague, or even overly positive. Little to no criticism is expressed. Problems are ignored or only hinted at. The response feels like a generic summary or polite feedback, not a critical evaluation.
- **Score 2 (Minimal alignment):** Some mild criticism is present, but it is heavily softened, hedged, or mixed with excessive praise. The response often uses balanced or diplomatic phrasing that dilutes the critical tone.
- **Score 3 (Basic alignment):** The response identifies key flaws and weaknesses, but still uses moderate or cautious language. Some sentences are direct, while others remain neutral or balanced. Overall tone is somewhat critical, but not sharply so.
- **Score 4 (Good alignment):** Tone is clearly critical and focuses on problems and shortcomings. The response directly points out issues with limited hedging or

softening. There may still be a few neutral or mildly positive phrases, but the main emphasis is on critique.

- **Score 5 (Excellent alignment):** Tone is consistently and explicitly critical, directly highlighting flaws, inconsistencies, and limitations. Language is clear, firm, and unambiguous without being abusive or disrespectful. The response avoids unnecessary balance or praise and maintains a sharp, problem-focused style throughout.

**Feedback:**

### C.4 Reflective and Elaborative Prompt

Finally, this prompt evaluates the depth and thoughtfulness of the generated text. It rewards detailed reasoning, the use of concrete examples, and the consideration of multiple perspectives, distinguishing deep reflection from simple surface-level explanations.

**Evaluation Prompt: Reflective and Elaborative**

**Task Description:** An instruction and a response to evaluate, a reference answer that can get a score of 5, and a score rubric representing an evaluation criteria are given. You are evaluating how well the response matches the following target style: *"Please write in a reflective and elaborative way, carefully explaining reasoning with detailed examples and considering multiple perspectives."*
1. Write a detailed feedback that assesses the quality of the response strictly based on the given score rubric, focusing only on style match, not on factual correctness or task success in general. [... Steps 2-5 same as above ...]
**The instruction to evaluate:** {{ instruction }}
**Response to evaluate:** {{ response }} **Reference Answer :** {{ reference_answer }}
**Score Rubrics:**

- **Criteria:** Evaluate how well the response matches a reflective and elaborative style that carefully explains reasoning with detailed examples and considers multiple perspectives.
- **Score 1 (Severe mismatch):** Response is brief, surface-level, or purely declarative. Little to no explanation of reasoning. No meaningful examples or alternative perspectives. Tone is more like a short answer than a reflection.
- **Score 2 (Minimal alignment):** Some reasoning is mentioned, but it is shallow or underdeveloped. Few or no concrete examples. Alternative perspectives are barely acknowledged or only named without real discussion. Overall, the response feels more explanatory than reflective.
- **Score 3 (Basic alignment):** The response includes a clear line of reasoning and at least one example or analogy, but depth is uneven. It may mention another perspective but not explore it deeply. The style is somewhat reflective and elaborative, yet could benefit from more detail or perspective-taking.
- **Score 4 (Good alignment):** The response thoughtfully explains its reasoning, uses several concrete details or examples, and meaningfully acknowledges at least one alternative viewpoint. It engages in reflection rather than just stating conclusions, with only minor gaps in depth or coverage.
- **Score 5 (Excellent alignment):** Highly reflective and thoroughly elaborative. Carefully walks through the reasoning process, supports points with detailed and relevant examples, and considers multiple perspectives in a balanced and insightful way. The response feels like a deep, thoughtful reflection rather than a simple explanation.

**Feedback:**

## D Alignment Between Human Evaluation and LLM Judgment

To quantitatively validate the reliability and robustness of our automated evaluation pipeline (Figure 4), we rigorously analyzed the correlation between human judgments and LLM-based scores. We aggregated the mean scores across all 5 evaluated

models and 4 target styles on the LaMP-7 dataset, for both the Style and Persona (Personalization) dimensions.

To capture a comprehensive view of the alignment—evaluating both the linear agreement of absolute scores and the consistency of relative model rankings—we computed three standard statistical metrics: Spearman's rank correlation coefficient ($\rho$), Kendall's rank correlation coefficient ($\tau$), and the Pearson correlation coefficient ($r$). The system-level correlation results are summarized in Table 10.

Table 10: System-level correlation between aggregated human scores and LLM judgments on LaMP-7.

| **Metric** | **Spearman's $\rho$** | **Kendall's $\tau$** | **Pearson's $r$** |
|---|---|---|---|
| Style Score | **0.864** | 0.725 | 0.858 |
| Persona Score | **0.712** | 0.589 | 0.704 |

**High Alignment in Style Evaluation.** As shown in Table 10, the LLM judge exhibits an exceptionally strong correlation with human evaluators in assessing style match (Pearson $r = 0.858$). More importantly, the high rank correlation scores (Spearman's $\rho = 0.864$, Kendall's $\tau = 0.725$) indicate that the LLM is highly reliable at correctly ranking the models' stylistic capabilities. Both sets of evaluations consistently demonstrate how models adapt to explicit instructions (e.g., favoring the *Concise* setting while struggling with the *Critical* setting), proving that the LLM robustly captures surface-level stylistic traits in a manner nearly indistinguishable from human annotators.

**Strong Consistency in Persona Assessment.** For the personalization dimension (Persona Score), we also observe a strong positive correlation, with a Pearson's $r$ of $0.704$ and a Spearman's $\rho$ of $0.712$. This indicates that the LLM judge successfully operationalizes the complex, profile-grounded rubrics to evaluate true persona adherence.

Crucially, the strong rank correlation ($\rho = 0.712$) demonstrates robust system-level agreement: both human and LLM evaluators consistently identify **PsPLUG** as the state-of-the-art approach across various stylistic contexts, while appropriately penalizing non-personalized or weakly-grounded baselines. The minor remaining variance between human and LLM scores ($r \approx 0.70$) primarily stems from the inherent subjectivity of personalization. Human annotators occasionally weight surface-level fluency and verbosity slightly higher than strict profile usage, whereas the LLM adheres rigidly to the predefined persona-grounding rubric. Nonetheless, the overall strong alignment across both dimensions thoroughly justifies the employment of our LLM-based judge as a rigorous, scalable, and highly accurate proxy for human evaluation.