BAYESIAN USER-STATE INFERENCE FOR ONLINE LLM PERSONALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalizing large language models (LLMs) online requires inferring user preferences from sparse histories and refining these estimates as new interactions arrive. Existing approaches typically select or summarize histories without explicitly modeling uncertainty about preferences. Yet observed user behavior reflects both context and personal preferences, making limited evidence ambiguous and complicating how much to revise estimates. To address this ambiguity and refine preference estimates as evidence accumulates, we propose Bayesian User-State Inference (BayUSI), a framework that treats deviations from input-conditioned population predictions as personal evidence and combines them with shared preference patterns. A learned mixture prior captures recurring preferences across users without annotated groups, while Bayesian updates incorporate each user's history into a posterior over their preferences. As interactions accumulate, posterior uncertainty governs how strongly new evidence updates preference estimates, providing query-specific conditioning for a frozen LLM. Training through the LLM's prediction loss aligns preference inference with personalized generation without per-user fine-tuning. Experiments on four LaMP tasks show consistent improvements under sparse-history and evolving-history protocols, demonstrating the value of uncertainty-aware preference inference for online LLM personalization.

1 INTRODUCTION

Personalizing large language models (LLMs) for users is becoming a central capability for interactive systems (Wu et al., 2025; Zhang et al., 2025b). Deployed LLMs should adapt to each user's writing style (Salemi et al., 2024b; Mysore et al., 2024), preferences (Kemper et al., 2024; He et al., 2023), and notion of relevance in search (Baek et al., 2024). In long-running deployments, effective personalization requires learning from incrementally available user histories (Jiang et al., 2025).

However, personalization in such settings requires predicting with incomplete knowledge of the user. Patterns learned across users provide a starting point when individual histories are sparse, but this understanding must be refined as personal evidence accumulates. This raises a central question: *how should a model revise its estimate of a user's preferences with each new observation?*

Existing methods incorporate personal information through retrieved histories (Salemi et al., 2024b;a), compact user representations (Liu et al., 2025b; Tang et al., 2024), and user-specific parameter adaptation (Tan et al., 2025; Zhang et al., 2025a; Tan et al., 2024). However, these approaches leave preference uncertainty implicit: retrieval scores and attention weights determine which observations to emphasize, but not how strongly they support a preference. This distinction matters because even a highly relevant interaction may provide only limited evidence about a user's preferences. For example, a high rating alone does not reveal whether a user generally rates generously or simply liked that particular item. Preference inference therefore requires interpreting each response in its input context. As new observations arrive, the model must determine how much to revise its understanding of those preferences, balancing patterns learned across users with the user-specific evidence.

Therefore, we view online LLM personalization as sequential inference over user preferences, with two central challenges: **(1) What to share?** Without annotated user groups, the model must identify transferable preference patterns and determine which apply to each individual. **(2) How much to update?** As interactions accumulate, the model must determine how strongly new evidence should revise existing preference estimates.

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

To address both challenges, we propose Bayesian User-State Inference (BayUSI), a framework for LLM personalization that shares information across users through *learned Bayesian adaptation* while calibrating individual adaptation to available evidence. To address **what to share**, BayUSI learns a mixture prior over preference patterns and uses each user's evidence to infer posterior weights over these patterns, enabling selective sharing without annotated groups. To address **how much to update**, Bayesian conditioning scales preference updates by posterior uncertainty and the disagreement between new evidence and current predictions, allowing stronger corrections under high uncertainty and more conservative adjustments when existing evidence is already strong. Evidence is derived from deviations of the user's observed responses from an input-conditioned population prediction, separating personal preferences from input effects. The resulting posterior provides a query-specific preference adjustment that conditions a frozen LLM through lightweight prefixes. Shared representations, the mixture prior, and the conditioning interface are jointly trained to predict users' observed responses, enabling adaptation through posterior updates without per-user fine-tuning.

Our work establishes a Bayesian formulation of online LLM personalization that unifies statistical sharing across users with uncertainty-aware adaptation to individuals. We characterize how shared priors and posterior uncertainty govern the influence of personal evidence on query-specific predictions, providing a theoretical account of the adaptation process. Experiments on the LaMP benchmark show consistent improvements in online personalization, with analyses tracing how posterior updates shape the user states used to condition the LLM.

2 Problem Definition

We study online personalization, where the model must generate for a user using only evidence observed before the current prediction. For each user $u \in \mathcal{U}$, consider a stream of T_u interactions, where $h_{u,i} = (x_{u,i}, y_{u,i})$ denotes the i-th input–response pair. At step t, the model receives a query $x_{u,t} \in \mathcal{X}$ and predicts $y_{u,t} \in \mathcal{Y}$ using only previously observed interactions, which form the prefix history $\mathcal{H}_{u,<t} = (h_{u,1}, \ldots, h_{u,t-1})$. The objective is to estimate

$$p_{\Theta,\theta}(y_{u,t} \mid x_{u,t}, \mathcal{H}_{u,<t}), \tag{1}$$

where Θ denotes the frozen LLM parameters and θ denotes shared personalization parameters learned from training users. Evaluation users are disjoint from training users. At deployment, Θ and θ remain fixed; personalization changes only through the evidence available for each user.

We study personalization from partial user histories that become available incrementally in non-overlapping segments of input–response pairs. Before each prediction, the newly observed segment is added to $\mathcal{H}_{u,<t}$. The model then predicts $y_{u,t}$ for the query $x_{u,t}$ using only this accumulated history.

When the available history is sparse, individual observations may provide insufficient evidence to infer user preferences reliably. Our goal is to support this inference with patterns learned across users and refine each user's preference estimate as interactions accumulate, without per-user fine-tuning.

3 Bayesian User-State Inference (BayUSI)

We cast online personalization as sequential Bayesian inference over user preferences. As shown in Figure 1, BayUSI interprets responses in their input contexts as personal evidence and combines it with a shared mixture prior over recurring preference patterns without annotated user groups. Bayesian conditioning yields a posterior over which shared patterns apply to the user and how the user's preferences deviate from those patterns, with posterior uncertainty governing how strongly new evidence updates the preference estimate. This posterior provides query-specific conditioning for a frozen LLM. The shared components are trained through the LLM's response prediction loss, while deployment adaptation requires only posterior updates without per-user fine-tuning.

3.1 Latent User Preferences and Shared Prior

We approach personalization by inferring user preferences from observed interactions. These preferences are not directly observable, since each user's response also depends on its input. We thus characterize their effect through systematic deviations from population predictions across inputs.

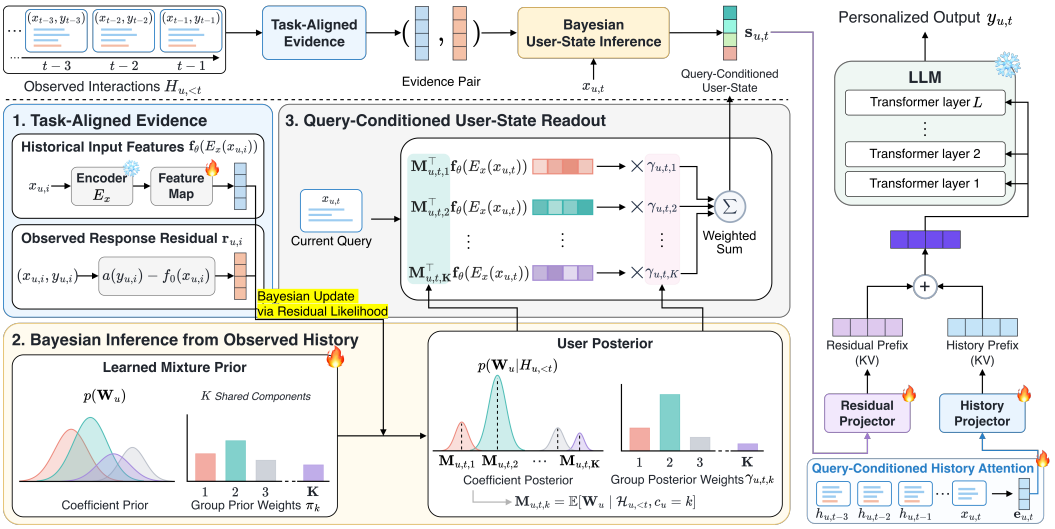


Figure 1: Overview of **Bayesian User-State Inference (BayUSI)**. **(1)** Historical interactions yield task-aligned evidence: input features and response residuals against an input-conditioned population prediction. **(2)** This evidence updates a learned mixture prior over shared preference patterns, yielding posterior user coefficients and pattern weights. **(3)** Posterior-weighted component predictions form a query-specific user state $\mathbf{s}_{u,t}$. The user state and a query-conditioned history representation produce separate key–value prefixes that jointly condition the frozen LLM. The shared prior is learned **without group annotations**, and unseen users adapt through posterior updates **without per-user fine-tuning**.

To capture these deviations, we represent user u's preferences with a latent coefficient matrix $\mathbf{W}_u \in \mathbb{R}^{d\times p}$. These latent coefficients describe how the user's expected response deviates from a population prediction as a function of input features, capturing both input-independent response tendency and input-dependent variation. We specify the features and observation model in Section 3.2.

We introduce a shared mixture prior, addressing unreliable estimation of $\mathbf{W}_u$ under sparse histories:

$$\begin{aligned} c_u &\sim \text{Categorical}(\boldsymbol{\pi}), \\ \mathbf{W}_u \mid c_u = k &\sim \mathcal{MN}(\mathbf{M}_{0,k}, \boldsymbol{\Lambda}_{0,k}^{-1}, \mathbf{I}_p), \qquad k = 1, \ldots, K. \end{aligned} \tag{2}$$

Here c_u indexes a shared preference pattern, and $\mathcal{MN}$ denotes a matrix-normal distribution. For pattern k, $\mathbf{M}_{0,k}$ gives the typical preference coefficients, while $\boldsymbol{\Lambda}_{0,k}^{-1}$ captures across-user variation around this mean. The prior π_k represents the probability of pattern k before observing the user's history. We learn these pattern means, their variation, and prior probabilities from training users without group annotations. The number of patterns K is fixed; we use $K = 4$ in the main experiments and examine sensitivity on LaMP-2 and LaMP-4 in Appendix D.1.

Personalization then requires inferring which shared patterns are supported by the user's history and how the user's coefficients differ from each pattern's mean. We therefore maintain a joint posterior over c_u and $\mathbf{W}_u$. We use one latent coefficient matrix $\mathbf{W}_u$ to explain the user's interactions within a stream and update this joint posterior as new observations become available.

3.2 Constructing User Evidence

To infer these preferences, we interpret each response relative to its input context. Let $\mathbf{a}(y) \in \mathbb{R}^p$ be a fixed task-specific response representation, encoding categorical labels, rating values, or projected frozen response embeddings. Let $f_0(x) \in \mathbb{R}^p$ predict this representation from the input alone. For each previously observed interaction $(x_{u,i}, y_{u,i})$, $i < t$, we compute the observed response residual:

$$\mathbf{r}_{u,i} = \mathbf{a}(y_{u,i}) - f_0(x_{u,i}). \tag{3}$$

We fit f_0 via user-balanced ridge regression on PCA features of frozen input embeddings from training users, then freeze it. Appendix A details the response representations and population predictor.

Each observed residual is paired with features of the input that elicited it to infer the user's latent preference coefficients $\mathbf{W}_u$. A frozen encoder E_x produces an input embedding, which a learnable feature map transforms into $\mathbf{f}_{u,i} = \mathbf{f}_\theta(E_x(x_{u,i})) \in \mathbb{R}^d$, including a constant intercept feature. For brevity, we write $\mathbf{f}_\theta(x)$ for $\mathbf{f}_\theta(E_x(x))$ below.

We distinguish the observed residual $\mathbf{r}_{u,i}$ computed in Equation (3) from its conditional mean under the model, $\bar{\mathbf{r}}_{u,i}(\mathbf{W}_u) := \mathbf{W}_u^\top \mathbf{f}_{u,i}$. We specify the observation likelihood as

$$p(\mathbf{r}_{u,i} \mid x_{u,i}, \mathbf{W}_u) = \mathcal{N}\big(\mathbf{r}_{u,i}; \bar{\mathbf{r}}_{u,i}(\mathbf{W}_u), \sigma^2 \mathbf{I}_p\big)\,. \tag{4}$$

This likelihood treats the observed residual as a noisy realization around the conditional mean, allowing the historical pairs $(\mathbf{f}_{u,i}, \mathbf{r}_{u,i})$ to provide evidence for inferring $\mathbf{W}_u$.

The coefficients associated with the intercept capture input-independent response tendencies, while the remaining coefficients describe how deviations vary with the input. Thus, the same response residual can provide different evidence about preferences in different contexts. The observation variance $\sigma^2 > 0$ is set using training data and held fixed.

3.3 Sequential Bayesian Updates

Given the observed pairs $\{(\mathbf{f}_{u,i}, \mathbf{r}_{u,i})\}_{i<t}$, we infer a joint posterior over the user's coefficients $\mathbf{W}_u$ and component membership c_u. Combining the shared prior with the residual likelihood yields closed-form Gaussian updates within each component (Bishop, 2006):

$$\begin{aligned} \mathbf{\Lambda}_{u,t,k} &= \mathbf{\Lambda}_{0,k} + \sigma^{-2} \sum_{i<t} \mathbf{f}_{u,i} \mathbf{f}_{u,i}^\top, \\ \mathbf{M}_{u,t,k} &= \mathbf{\Lambda}_{u,t,k}^{-1} \left(\mathbf{\Lambda}_{0,k} \mathbf{M}_{0,k} + \sigma^{-2} \sum_{i<t} \mathbf{f}_{u,i} \mathbf{r}_{u,i}^\top \right). \end{aligned} \tag{5}$$

Within component k, the conditional posterior is $\mathcal{MN}(\mathbf{M}_{u,t,k}, \mathbf{\Lambda}_{u,t,k}^{-1}, \mathbf{I}_p)$. Its mean $\mathbf{M}_{u,t,k}$ adapts the shared preference pattern $\mathbf{M}_{0,k}$ to the user's observed responses and their input contexts, yielding a component-specific estimate of the user's coefficients. The prior precision $\mathbf{\Lambda}_{0,k}$ controls how strongly this estimate relies on the shared pattern. The posterior covariance $\mathbf{\Lambda}_{u,t,k}^{-1}$ captures the remaining uncertainty in these coefficients and helps determine how strongly subsequent observations revise the estimate.

To determine which patterns the history supports, we integrate out $\mathbf{W}_u$ and compute the marginal likelihood,

$$\mathcal{E}_{u,t,k} = p(\{\mathbf{r}_{u,i}\}_{i<t} \mid \{x_{u,i}\}_{i<t}, c_u = k)\,. \tag{6}$$

The posterior component probabilities are

$$\gamma_{u,t,k} = p(c_u = k \mid \mathcal{H}_{u,<t}) = \frac{\pi_k \mathcal{E}_{u,t,k}}{\sum_{j=1}^K \pi_j \mathcal{E}_{u,t,j}}. \tag{7}$$

This marginal likelihood accounts for coefficient uncertainty. Together, $\gamma_{u,t,k}$, $\mathbf{M}_{u,t,k}$, and $\mathbf{\Lambda}_{u,t,k}^{-1}$ describe which shared patterns apply to the user and the user's preferences within each pattern. Appendix B derives the posterior and evidence.

Uncertainty-aware adaptation. With shared parameters fixed, the posterior can be updated recursively as new interactions arrive. Within one component, let $\mathbf{M}$ and $\mathbf{S}$ denote the coefficient mean and covariance before observing a new pair $(\mathbf{f}, \mathbf{r})$. Then

$$\mathbf{g} = \frac{\mathbf{S}\mathbf{f}}{\sigma^2 + \mathbf{f}^\top \mathbf{S}\mathbf{f}}, \qquad \mathbf{M}^+ = \mathbf{M} + \mathbf{g}(\mathbf{r} - \mathbf{M}^\top \mathbf{f})^\top, \qquad \mathbf{S}^+ = \mathbf{S} - \mathbf{g}\mathbf{f}^\top \mathbf{S}. \tag{8}$$

The innovation $\mathbf{r} - \mathbf{M}^\top \mathbf{f}$ measures the difference between the newly observed residual and its current predicted mean, and the gain $\mathbf{g}$ determines how strongly this difference updates the coefficient estimate. Component probabilities are updated by reweighting their previous values with the predictive likelihood $\mathcal{N}(\mathbf{r}; \mathbf{M}^\top \mathbf{f}, (\sigma^2 + \mathbf{f}^\top \mathbf{S}\mathbf{f})\mathbf{I}_p)$ and renormalizing. The following proposition makes the effect on a query explicit.

Proposition 1 (Query-specific influence of new evidence)**.** *For fixed shared parameters and a fixed mixture component, an observation* $(\mathbf{f}, \mathbf{r})$ *changes the predicted residual at a query with feature vector* $\mathbf{f}_*$ *by*

$$(\mathbf{M}^+ - \mathbf{M})^\top \mathbf{f}_* = \frac{\mathbf{f}_*^\top \mathbf{S}\mathbf{f}}{\sigma^2 + \mathbf{f}^\top \mathbf{S}\mathbf{f}}(\mathbf{r} - \mathbf{M}^\top \mathbf{f}). \tag{9}$$

At the observed input, the multiplier is $v/(\sigma^2 + v) \in [0, 1)$*, where* $v = \mathbf{f}^\top \mathbf{S}\mathbf{f}$*, and* $\mathbf{S}^+ \preceq \mathbf{S}$*.*

Within each component, new evidence changes the predicted residual at a query according to the observation's prediction error and the covariance-weighted relationship between the observed input and the query. At the observed input, greater uncertainty relative to noise gives new evidence more weight. The full mixture prediction reflects updated posterior weights over components. These coefficient updates follow standard Gaussian conditioning, with learned features and priors shaping their effect on personalization. Appendix C proves the proposition and batch–sequential equivalence.

3.4 Conditioning a Frozen LLM

For the current query $x_{u,t}$, the response $y_{u,t}$ is unobserved, so its residual cannot yet be computed. Instead, we average its conditional mean over the coefficient posterior inferred from past interactions:

$$\mathbf{s}_{u,t} = \mathbb{E}_{\mathbf{W}_u | \mathcal{H}_{u,<t}}[\bar{\mathbf{r}}_{u,t}(\mathbf{W}_u)] = \sum_{k=1}^{K} \gamma_{u,t,k} \mathbf{M}_{u,t,k}^\top \mathbf{f}_\theta(x_{u,t}). \tag{10}$$

Each component's predicted residual is weighted by its posterior probability. Thus, $\mathbf{s}_{u,t}$ is the posterior predictive mean for the current query, whereas $\mathbf{r}_{u,i}$ is an observed residual from the history.

To retain interaction details that complement this estimate, we also compute a query-dependent history representation $\mathbf{e}_{u,t}$ using a PPlug-style attention aggregator (Liu et al., 2025b) over $\mathcal{H}_{u,<t}$; see Appendix D. We combine the two signals through an additive deep prefix. At transformer layer ℓ, the key and value prefix tensors are

$$\mathbf{P}_{u,t}^{(\ell)} = G_\theta^{(\ell)}(\mathbf{e}_{u,t}) + \Delta G_\theta^{(\ell)}(\mathbf{s}_{u,t}), \tag{11}$$

The two mappings produce tensors of the same shape, with addition applied separately to key and value prefixes. The history branch preserves query-relevant interaction details, while the Bayesian branch incorporates preferences inferred from the shared prior and personal evidence. Both branches are jointly trained from scratch, with the Bayesian branch's output projection initialized to zero. The resulting prefixes condition the frozen LLM:

$$p_{\Theta,\theta}(y_{u,t} \mid x_{u,t}, \mathcal{H}_{u,<t}) = p_\Theta(y_{u,t} \mid x_{u,t}, \{\mathbf{P}_{u,t}^{(\ell)}\}_\ell). \tag{12}$$

Posterior uncertainty influences the inference of $\mathbf{s}_{u,t}$. Bayesian marginalization thus occurs in the residual model; LLM generation uses a deterministic posterior readout.

3.5 Task-Aligned Training and Online Inference

We jointly optimize the input feature map, shared mixture prior, and LLM conditioning interface through the response prediction loss, allowing task supervision to shape both preference inference and its use in personalized generation:

$$\mathcal{L}(\theta) = -\sum_{(u,t)\in\mathcal{D}_{\text{train}}} \log p_\Theta\Big(y_{u,t} \mid x_{u,t}, \{\mathbf{P}_{u,t}^{(\ell)}(\theta)\}_\ell\Big), \tag{13}$$

where θ collects the trainable personalization parameters and Θ denotes the frozen LLM parameters. Each prefix is constructed from the current query and the observed history $\mathcal{H}_{u,<t}$; the target response $y_{u,t}$ provides supervision but is excluded from the posterior used to predict it. Gradients propagate through the posterior means and component probabilities, shaping the feature space and shared prior to support task-relevant Bayesian adaptation. The LLM, base encoders, response representation, population predictor, and observation variance remain fixed.

This procedure learns a shared adaptation mechanism for unseen users without per-user fine-tuning. Starting from the learned mixture prior, the model conditions on available history to construct a query-specific prefix, incorporating newly revealed responses into future predictions. Adaptation requires neither optimizer steps nor shared-parameter updates.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate on LaMP-{2,3,4,5}, covering movie tagging, product rating, news headline generation, and scholarly title generation (Salemi et al., 2024b). We use the user-based split to study adaptation to unseen users and construct online prediction episodes from validation profiles. Our protocol requires observed input–response pairs for the original task. **LaMP-1 and LaMP-7 do not satisfy this requirement:** they lack the task-aligned input–response pairs required by our protocol; including them would require synthetic inputs. LaMP-6 is not publicly available.

Online evaluation. We treat each user profile as an ordered sequence of interactions, each comprising an input and its observed response. We progressively reveal these interactions in the supplied order;*online* denotes sequential evidence availability. *Evolving-history* reveals 10 interactions per step and predicts one deterministically selected target from the next unrevealed block of up to 10 interactions. *Sparse-history* reveals one interaction per step and predicts the next response, yielding histories of 1–10 observed interactions and at most 10 predictions per profile. This protocol differs from LaMP's official fixed-target evaluation (Salemi et al., 2024b; Liu et al., 2025b).

Training and model selection. We preserve the official training/validation boundary and partition training data into separate subsets for fitting response representations and the population predictor, calibrating observation noise, training personalization modules, and selecting checkpoints. Training uses all unique targets in the sparse and evolving settings, retaining their natural proportions. Checkpoints are selected on an internal training-data holdout.

Metrics and aggregation. Following Salemi et al. (2024b); Liu et al. (2025b), we report Accuracy and macro-F1 for LaMP-2, MAE and RMSE for LaMP-3, and ROUGE-1/L for LaMP-{4,5}. Category-level aggregates normalize each metric against BM25, using $100x/x_{\mathrm{BM25}}$ for higher-is-better metrics and $100x_{\mathrm{BM25}}/x$ otherwise, then average within tasks and across tasks.

Baselines. We compare against four personalization baselines, grouped into two categories. **Retrieval-based methods** include **BM25** (Robertson & Zaragoza, 2009) and **Contriever** (Izacard et al., 2022), which retrieve relevant interactions from the user's observed history through lexical and dense semantic matching, respectively. **Plug-in methods** include **PPlug** (Liu et al., 2025b), which conditions a frozen LLM on a learned, query-dependent user representation, and **PPlug+kNN**, which augments this representation with the mean representation of neighboring training users retrieved by history similarity. We implement both PPlug variants with a trainable query encoder and a deep-prefix interface, whereas BayUSI learns query projections over frozen base embeddings. All methods including our method use the same frozen backbone LLM, `Llama-3.1-8B-Instruct`.

Implementation. All methods use the frozen `Llama-3.1-8B-Instruct` backbone. BayUSI uses frozen `bge-base-en-v1.5` embeddings and jointly trains its feature map, finite mixture prior, history aggregator, and prefix mappings with AdamW. LLM input limits are 256 tokens for LaMP-{2,4} and 512 for LaMP-{3,5}; encoder inputs are limited to 512 tokens. We use learning rate of 1×10^{-4}, weight decay 0.05, batch size 64, and warmup ratio 0.05. We train for 20 epochs on LaMP-2 and for 5 epochs on LaMP-4 and LaMP-5 and 1 epoch on LaMP-3.

4.2 MAIN RESULTS

Table 1 shows that BayUSI consistently outperforms retrieval-based and plug-in baselines across all reported metrics on LaMP-{2,3,4} under both evaluation settings. The comparison with PPlug is particularly informative: BayUSI improves over a model that already aggregates history in a query-dependent manner. Its advantage also persists against PPlug+kNN, which explicitly incorporates information from neighboring training users. These comparisons support the effectiveness of our approach to combining individual evidence with shared preference structure.

The clearest gains appear in the evolving-history setting, particularly on movie tagging and product rating. Relative to the strongest baseline for each metric, BayUSI improves LaMP-2 macro-F1 by 18.2% and reduces LaMP-3 MAE by 15.8%. Improvements in both accuracy and macro-F1 indicate that the tagging gains are not confined to overall prediction accuracy, while lower MAE and RMSE

Table 1: Online personalization on LaMP-2–5 under evolving- and sparse-history settings. LaMP-1/7 are excluded because their profile items do not provide input-target pairs required for segment-level next-target evaluation. Results are sample-weighted, with RMSE pooled from squared errors. Arrows indicate the direction of improvement. Bold/underline denote the best/second-best scores, respectively, including ties. Shaded rows report task-balanced normalized averages relative to BM25 (100), using $100x/x_{\mathrm{BM25}}$ for ↑ metrics and $100x_{\mathrm{BM25}}/x$ for ↓ metrics.

		RAG		Plug-in		
Dataset	**Metric**	**BM25**	**Contriever**	**PPlug**	**PPlug+kNN**	**BayUSI**
Evolving-history						
LaMP-2: Movie tagging	Acc. (↑)	0.489	0.486	<u>0.632</u>	0.630	**0.732**
	F1 (↑)	0.412	0.406	<u>0.603</u>	<u>0.611</u>	**0.722**
LaMP-3: Product rating	MAE (↓)	0.349	0.340	<u>0.259</u>	0.261	**0.218**
	RMSE (↓)	0.694	0.676	<u>0.564</u>	0.567	**0.509**
Classification mean (norm. avg.)		100.0	100.8	<u>133.4</u>	133.3	**155.3**
LaMP-4: News headline	ROUGE-1 (↑)	0.145	0.153	0.206	<u>0.211</u>	**0.217**
	ROUGE-L (↑)	0.131	0.138	0.184	<u>0.189</u>	**0.197**
LaMP-5: Scholarly title	ROUGE-1 (↑)	0.224	0.218	0.470	**0.472**	**0.472**
	ROUGE-L (↑)	0.196	0.190	0.429	**0.430**	**0.430**
Generation mean (norm. avg.)		100.0	101.3	177.8	<u>180.0</u>	**182.5**
Sparse-history						
LaMP-2: Movie tagging	Acc. (↑)	0.458	0.446	<u>0.673</u>	0.652	**0.696**
	F1 (↑)	0.384	0.377	<u>0.612</u>	0.602	**0.656**
LaMP-3: Product rating	MAE (↓)	0.364	0.362	<u>0.255</u>	0.256	**0.238**
	RMSE (↓)	0.715	0.711	<u>0.565</u>	<u>0.565</u>	**0.540**
Classification mean (norm. avg.)		100.0	99.2	<u>143.9</u>	142.0	**152.0**
LaMP-4: News headline	ROUGE-1 (↑)	0.142	0.144	<u>0.205</u>	0.204	**0.206**
	ROUGE-L (↑)	0.125	0.127	<u>0.186</u>	<u>0.186</u>	**0.190**
LaMP-5: Scholarly title	ROUGE-1 (↑)	0.198	0.194	0.470	0.472	**0.473**
	ROUGE-L (↑)	0.171	0.168	0.429	**0.431**	**0.431**
Generation mean (norm. avg.)		100.0	99.8	195.4	<u>195.7</u>	**197.0**

Table 2: Effect of coefficient updating on target-response prediction. We report NLL reductions ($\mathrm{NLL}_{\text{without update}} - \mathrm{NLL}_{\text{full}}$); positive values indicate improvement. The control replaces posterior coefficient means with learned prior means at the same checkpoint, retaining posterior group weights. Brackets denote 95% paired dependency-block bootstrap confidence intervals (2,000 resamples).

Dataset	Setting	NLL reduction ↑	95% CI
LaMP-2	Sparse	+0.0488	[0.0168, 0.0847]
	Evolving	+0.0609	[0.0308, 0.1103]
LaMP-4	Sparse	+0.0150	[0.0103, 0.0197]
	Evolving	+0.0489	[0.0351, 0.0621]

show a consistent reduction in rating errors. On generation tasks, our method also achieves the best aggregate performance, with a normalized average of 182.5. This pattern suggests that BayUSI provides useful personalization signals as evidence accumulates, beyond those captured by history aggregation or neighbor-based sharing alone.

BayUSI remains effective under sparse-history evaluation, where each prediction relies on only 1–10 observed interactions. It achieves the strongest aggregate performance for both classification and generation, and obtains the best results on all metrics. These improvements are consistent with the motivation for combining a learned population prior with limited individual evidence. At the same time, the larger gains under evolving-history show that its empirical advantage is not confined to sparse histories. The two settings therefore provide complementary evidence of effectiveness across different levels of available user information.

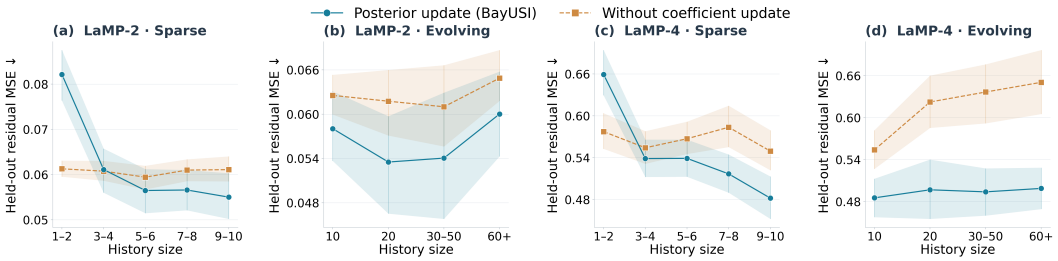


Figure 2: Effect of posterior coefficient updates on held-out residual prediction on LaMP-2 and LaMP-4 across history sizes. Compared with a control that retains learned prior coefficient means, BayUSI achieves lower average residual MSE in the evolving-history setting, showing the benefit of refining preference estimates with user history. Lower MSE indicates more accurate prediction.

Table 3: Ablation study of BayUSI on LaMP-2 and LaMP-4 under evolving- and sparse-history settings. "w/o Bayesian state" removes state conditioning; "Prior-mean readout" replaces posterior coefficient means with prior means; "w/o residualization" uses response encodings directly. F1 denotes macro-F1; R-1/R-L denote ROUGE-1/ROUGE-L. Higher is better; bold and underline mark the best and second-best scores, including ties.

	LaMP-2				LaMP-4			
	Evolving		Sparse		Evolving		Sparse	
Method	Acc.	F1	Acc.	F1	R-1	R-L	R-1	R-L
BayUSI (Ours)	**0.732**	**0.722**	<u>0.696</u>	**0.656**	**0.217**	**0.197**	**0.206**	**0.190**
w/o Bayesian state	0.677	<u>0.663</u>	0.668	0.620	<u>0.214</u>	<u>0.191</u>	<u>0.205</u>	<u>0.187</u>
Prior-mean readout	0.646	0.612	0.675	0.618	0.212	0.190	<u>0.205</u>	<u>0.187</u>
w/o residualization	<u>0.709</u>	0.644	**0.700**	<u>0.647</u>	0.209	0.187	**0.206**	0.186

4.3 Effect of Posterior Coefficient Updates

We examine whether adapting user-specific coefficients improves personalization beyond reweighting shared preference patterns. At the same checkpoint, we replace posterior coefficient means with learned prior means while retaining posterior group weights. Figure 2 measures held-out residual MSE on LaMP-2 and LaMP-4, comparing the predicted residual $\mathbf{s}_{u,t}$ with $\mathbf{a}(y_{u,t}) - f_0(x_{u,t})$. The target response is used only to compute evaluation error.

The benefit of coefficient adaptation becomes clearer as evidence accumulates. With only 1–2 interactions, BayUSI has higher residual MSE than the control on both tasks, highlighting the difficulty of individual adaptation from minimal evidence. In the 1–2 interaction bin, coefficient updates do not improve residual MSE; gains emerge with additional evidence, starting from the 5–6 bin on LaMP-2 and the 3–4 bin on LaMP-4. Under evolving histories, BayUSI outperforms the control across all bins on both tasks. The contrast is strongest on LaMP-4: BayUSI maintains stable error while the control's error rises with history size. These results support adapting the shared coefficients to capture individual response tendencies as histories grow.

Beyond residual prediction, Table 2 assesses whether coefficient updates improve the conditioned LLM's target response predictions. Posterior coefficient updates reduce average response NLL in all four task–setting combinations, with all 95% confidence intervals for the reduction above zero. Together, these findings support coefficient adaptation as a useful mechanism for translating accumulated user evidence into better personalized predictions.

4.4 Ablation Study

We examine three variants in Table 3. *Without Bayesian state* removes the additive Bayesian prefix branch while retaining query-dependent history aggregation. *Prior-mean readout* replaces posterior coefficient means with learned group prior means while retaining posterior group weights. *Without residualization* models response features directly, without subtracting the population prediction.

The full model outperforms both the history-only and prior-mean variants on every metric in both settings. The larger gaps on LaMP-2, particularly under evolving histories, suggest that adapting user-specific coefficients provides information beyond history aggregation and posterior group weighting alone. Improvements on LaMP-4 are smaller but consistent across both ROUGE metrics.

Residualization improves all metrics under evolving histories, supporting the value of modeling deviations from the population prediction. Its benefits are more selective under sparse histories: the full model achieves higher macro-F1 on LaMP-2 and ROUGE-L on LaMP-4, but slightly lower accuracy and tied ROUGE-1, respectively. These results support residualization as a useful component, with clearer benefits when more individual evidence is available.

5 Related Work

LLM personalization. Recent approaches personalize LLMs by incorporating user histories through parameter adaptation, retrieval, or user representations (Liu et al., 2025a; Xu et al., 2025). Parameter-efficient and collaborative fine-tuning provide user-specific capacity at additional training and deployment costs (Tan et al., 2025; 2024). Other methods condition a frozen backbone on retrieved interactions (Salemi et al., 2024b; Mysore et al., 2024), textual profiles (Tang et al., 2024), or learned plug-in user representations (Liu et al., 2025b). These mechanisms incorporate personal information but do not inherently represent uncertainty in preferences inferred from limited evidence. Our method complements history aggregation with Bayesian preference inference, refining uncertain preference estimates as evidence accumulates.

Online and shared user-state modeling. Sparse feedback and evolving preferences have long motivated temporal user modeling in recommender systems (Hu et al., 2008; Koren, 2010). LLM personalization inherits these challenges while requiring efficient adaptation without repeatedly retraining the generator. For LLM personalization, GPO (Zhao et al., 2024) enables few-shot adaptation to group preferences, PGraphRAG (Au et al., 2025) retrieves evidence through user-centric knowledge graphs, and PROPER (Zhang et al., 2025a) combines automatically learned group experts with user-specific LoRA adaptation. However, GPO requires group-specific preference feedback, PGraphRAG relies on an available user-centric graph, and PROPER incurs per-user LoRA training costs. In contrast, our method learns shared preference patterns without annotated user groups and adapts to new evidence through posterior updates, avoiding per-user fine-tuning.

Bayesian adaptation with learned priors. Conjugate Bayesian regression combines priors with observations through closed-form posterior updates (Bishop, 2006). Beyond fixed priors, hierarchical Bayesian methods learn shared structure across tasks for adaptation from limited data: Grant et al. (2018) interpret gradient-based meta-learning as approximate hierarchical Bayesian inference, while probabilistic MAML captures few-shot ambiguity by learning a distribution over models and approximating task-specific posteriors through stochastic adaptation (Finn et al., 2019). Complementing these gradient-based approaches, ALPaCA jointly learns neural features and a Gaussian weight prior for closed-form online updates (Harrison et al., 2018). Building on these ideas, our method learns a mixture prior over user-specific response tendencies and optimizes Bayesian conditioning through a frozen LLM's response prediction loss.

6 Conclusion

We presented BayUSI, an online LLM personalization framework that infers user preferences from sparse and incrementally observed evidence. BayUSI treats deviations from population-level response predictions as evidence of user-specific tendencies and combines this evidence with a shared mixture prior through Bayesian updates. The resulting posterior predicts a query-specific preference adjustment that conditions a frozen LLM. Jointly learning the input feature map, mixture prior, and conditioning interface through the response prediction loss aligns Bayesian inference with personalized generation, without per-user fine-tuning. Experiments on LaMP demonstrate the effectiveness of this approach under both evolving-history and sparse-history settings. Our results support Bayesian preference inference as a practical mechanism for combining shared patterns across users with individual evidence for online personalization. Future work will explore integration with retrieval for tasks requiring exact historical evidence and extensions to richer, long-horizon interaction streams.

AI USE STATEMENT

We used ChatGPT to check grammar, improve sentence clarity, and assist with LaTeX formatting of tables and figures. The authors reviewed and revised all AI-assisted content to ensure its accuracy and consistency with the work, and take full responsibility for the final manuscript.

REFERENCES

Steven Au, Cameron J. Dimacali, Ojasmitha Pedirappagari, Namyong Park, Franck Dernoncourt, Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan A. Rossi, and Nesreen K. Ahmed. Personalized graph-based retrieval for large language models, 2025. URL https://arxiv.org/abs/2501.02157.

Jinheon Baek, Nirupama Chandrasekaran, Silviu Cucerzan, Allen herring, and Sujay Kumar Jauhar. Knowledge-augmented large language models for personalized contextual query suggestion, 2024. URL https://arxiv.org/abs/2311.06318.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning, 2019. URL https://arxiv.org/abs/1806.02817.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes, 2018. URL https://arxiv.org/abs/1801.08930.

James Harrison, Apoorva Sharma, and Marco Pavone. Meta-learning priors for efficient online bayesian regression, 2018. URL https://arxiv.org/abs/1807.08912.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian Mcauley. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 720–730, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614949. URL https://doi.org/10.1145/3583780.3614949.

Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pp. 263–272, 2008. doi: 10.1109/ICDM.2008.22.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2022. URL https://arxiv.org/abs/2112.09118.

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

Sara Kemper, Justin Cui, Kai Dicarlantonio, Kathy Lin, Danjie Tang, Anton Korikov, and Scott Sanner. Retrieval-augmented conversational recommendation with prompt-based semi-structured natural language state tracking. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2786–2790. ACM, 2024. doi: 10.1145/3626772.3657670. URL http://dx.doi.org/10.1145/3626772.3657670.

Yehuda Koren. Collaborative filtering with temporal dynamics. *Commun. ACM*, 53(4):89–97, April 2010. ISSN 0001-0782. doi: 10.1145/1721654.1721677. URL https://doi.org/10.1145/1721654.1721677.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. A survey of personalized large language models: Progress and future directions, 2025a. URL https://arxiv.org/abs/2502.11528.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. LLMs + persona-plug = personalized LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9373–9385, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.461. URL https://aclanthology.org/2025.acl-long.461/.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In Sachin Kumar, Vidhisha Balachandran, Chan Young Park, Weijia Shi, Shirley Anugrah Hayati, Yulia Tsvetkov, Noah Smith, Hannaneh Hajishirzi, Dongyeop Kang, and David Jurgens (eds.), *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pp. 198–219, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.customnlp4u-1.16. URL https://aclanthology.org/2024.customnlp4u-1.16/.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, April 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Alireza Salemi, Surya Kallumadi, and Hamed Zamani. Optimization methods for personalizing large language models through retrieval augmentation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 752–762, New York, NY, USA, 2024a. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657783. URL https://doi.org/10.1145/3626772.3657783.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization, 2024b. URL https://arxiv.org/abs/2304.11406.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. Personalized pieces: Efficient personalized large language models through collaborative efforts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6459–6475, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.371. URL https://aclanthology.org/2024.emnlp-main.371/.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning, 2025. URL https://arxiv.org/abs/2402.04401.

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

Shujin Wu, Yi R. Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. Aligning LLMs with individual preferences via interaction. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7648–7662, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.511/.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. Personalized generation in large model era: A survey. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24607–24649, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1201. URL https://aclanthology.org/2025.acl-long.1201/.

Linhai Zhang, Jialong Wu, Deyu Zhou, and Yulan He. Proper: A progressive learning framework for personalized large language models with group-level adaptation, 2025a. URL https://arxiv.org/abs/2503.01303.

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, Nesreen Ahmed, and Yu Wang. Personalization of large language models: A survey, 2025b. URL https://arxiv.org/abs/2411.00027.

Siyan Zhao, John Dang, and Aditya Grover. Group preference optimization: Few-shot alignment of large language models, 2024. URL https://arxiv.org/abs/2310.11523.

A Representations and Population Predictor

A.1 Frozen embeddings and separate feature maps

We use the frozen `BAAI/bge-base-en-v1.5` encoder with CLS pooling and embedding dimension $D_x = 768$. Encoder inputs are truncated to 512 tokens. Let $\mathbf{h}(x) = E_x(x) \in \mathbb{R}^{768}$ denote its raw, unnormalized CLS embedding. For the input-only encoding, we extract the task input (movie description, review, article, or paper abstract) from the benchmark prompt and prepend the fixed string `Represent this sentence for searching relevant passages:`. The implementation concatenates this prefix directly with the extracted text. Response embeddings use the response text alone, without a query or retrieval prefix. Under the default encoder-instruction setting, history interaction strings likewise receive no additional task instruction.

The population predictor uses centered, whitened PCA coordinates of the raw input embeddings:

$$\begin{aligned} \mathbf{q}_0(x) &= \mathbf{D}_0^{-1/2}\mathbf{U}_0^\top(\mathbf{h}(x) - \boldsymbol{\mu}_0) \in \mathbb{R}^{q_0}, \\ \tilde{\mathbf{q}}_0(x) &= [1; \mathbf{q}_0(x)], \qquad q_0 = 7. \end{aligned} \tag{14}$$

Here $\mathbf{U}_0$ contains the leading eigenvectors of the sample covariance, computed with denominator $N_F - 1$, and $\mathbf{D}_0$ contains their eigenvalues. Both are fitted on the retained fitting interactions in split F. PCA fitting pools interactions without user weights; the subsequent ridge regression is user-balanced. There is no unit-length normalization before PCA, and no additional clipping or scaling after whitening. A requested rank with insufficient numerical support causes preparation to fail.

The task-trained Bayesian feature map instead normalizes the raw embedding before its learned projection:

$$\begin{aligned} \overline{\mathbf{h}}(x) &= \frac{\mathbf{h}(x)}{\max(\|\mathbf{h}(x)\|_2, 10^{-12})}, \\ \mathbf{f}_\theta(x) &= \begin{bmatrix} 1 \\ \tanh(\mathbf{A}_\theta \overline{\mathbf{h}}(x) + \mathbf{b}_\theta) \end{bmatrix} \in \mathbb{R}^d, \qquad d = 8. \end{aligned} \tag{15}$$

The projection has seven output coordinates and a bias; the intercept is exactly one. No further feature scaling or clipping is applied. In particular, the implementation argument `TASK_BAYES_DIM=8` specifies the total dimension *including* the intercept. The PCA coordinates of the population predictor and the learned Bayesian features are separate maps, even though their total dimensions are both eight. Neither reduces the dimension of the history embeddings supplied to the history-aggregation branch.

A.2 Task-specific response representations

The fixed map $\mathbf{a}(y) \in \mathbb{R}^p$ represents an observed response in the auxiliary space used for Bayesian inference. Its construction depends on the task.

Movie tagging (LaMP-2). We encode the $C = 15$ movie tags using orthonormal Helmert contrasts. Let $\mathbf{H} \in \mathbb{R}^{C\times(C-1)}$ satisfy $\mathbf{H}^\top\mathbf{H} = \mathbf{I}_{C-1}$ and $\mathbf{H}^\top\mathbf{1}_C = \mathbf{0}$. For a response with category index $c(y)$, we use

$$\mathbf{a}(y) = \mathbf{H}^\top \mathbf{e}_{c(y)} \in \mathbb{R}^{14}, \tag{16}$$

where $\mathbf{e}_{c(y)}$ is the corresponding one-hot vector. This representation preserves categorical distinctions without introducing an ordinal relationship between tags.

Product rating (LaMP-3). For an integer rating $y \in \{1, 2, 3, 4, 5\}$, we use

$$\mathbf{a}(y) = [(y - 3)/2] \in \mathbb{R}, \tag{17}$$

mapping the rating scale to $[-1, 1]$.

Text generation (LaMP-4 and LaMP-5). We encode the response text alone with a frozen encoder E_y and apply centered, whitened PCA:

$$\mathbf{a}(y) = \mathbf{D}_y^{-1/2}\mathbf{U}_y^\top \big(E_y(y) - \boldsymbol{\mu}_y\big) \in \mathbb{R}^p. \tag{18}$$

Here $\boldsymbol{\mu}_y$ is the mean response embedding, $\mathbf{U}_y$ contains the leading p covariance eigenvectors, and $\mathbf{D}_y$ contains their eigenvalues. These quantities are fitted using only response embeddings from the fitting split F of the training data. The response PCA is separate from the input PCA used by the population predictor. All response representations remain fixed during personalization training and evaluation.

A.3 Fitting the population predictor

The population predictor estimates the response representation from the input alone, providing a shared reference for constructing observed residuals. Using the augmented input PCA features $\widetilde{\mathbf{q}}_0(x)$ defined above, we write

$$f_0(x) = \mathbf{B}_0^\top \widetilde{\mathbf{q}}_0(x), \qquad \mathbf{B}_0 \in \mathbb{R}^{(q_0+1)\times p}. \tag{19}$$

Let $\mathcal{U}_F$ denote the users in fitting split F, and let $\mathcal{I}_u^F$ contain the retained fitting interactions of user u, with $n_u = |\mathcal{I}_u^F|$. We estimate $\mathbf{B}_0$ by user-balanced ridge regression:

$$\mathbf{B}_0 = \arg\min_{\mathbf{B}} \frac{1}{|\mathcal{U}_F|} \sum_{u\in\mathcal{U}_F} \frac{1}{n_u} \sum_{i\in\mathcal{I}_u^F} \big\|\mathbf{a}(y_{u,i}) - \mathbf{B}^\top \widetilde{\mathbf{q}}_0(x_{u,i})\big\|_2^2 + \lambda\|\mathbf{P}\mathbf{B}\|_F^2, \tag{20}$$

where $\lambda = 0.01$ and $\mathbf{P} = \mathrm{diag}(0, 1, \ldots, 1)$. The zero entry leaves the intercept unpenalized. Each fitting user receives the same total weight, regardless of the number of retained interactions.

After fitting, we freeze $\mathbf{B}_0$ and construct each observed response residual as

$$\mathbf{r}_{u,i} = \mathbf{a}(y_{u,i}) - \mathbf{B}_0^\top \widetilde{\mathbf{q}}_0(x_{u,i}). \tag{21}$$

The population coefficients $\mathbf{B}_0$ are shared across users and fitted before personalization training. They are distinct from the latent user coefficients $\mathbf{W}_u$, whose posterior is inferred from historical input features and these observed residuals. Neither the response representation nor the population predictor is fitted using evaluation users.

Hyperparameter settings. For the representations defined above, we use input feature dimension $d = 8$ (including the intercept) and response dimensions $p = 14, 1, 8, 8$ for LaMP-2–5, respectively. We use $K = 4$ mixture components in the main experiments; the sensitivity to K is examined separately in the appendix. The observation variance σ^2 is calibrated on the training-data calibration split and fixed at 0.057872, 0.136357, 0.569896, and 0.325113 for LaMP-2–5, respectively. These values reflect the scale of each task's response space and are not directly comparable across tasks.

B Posterior and Marginal-Likelihood Derivations

B.1 Completing the square

Fix a user, an observed prefix of n interactions, a component k, and all shared parameters. Suppress these indices and stack the input features and the observed residuals computed in Equation 3 as $\mathbf{F} \in \mathbb{R}^{n\times d}$ and $\mathbf{R} \in \mathbb{R}^{n\times p}$, whose rows are $\mathbf{f}_i^\top$ and $\mathbf{r}_i^\top$, respectively. Set $\mathbf{S}_0 = \boldsymbol{\Lambda}_0^{-1}$. The prior and likelihood are

$$\begin{aligned} p(\mathbf{W} \mid k) &= (2\pi)^{-dp/2}|\boldsymbol{\Lambda}_0|^{p/2} \exp\!\big[-\tfrac{1}{2}\,\mathrm{tr}\{(\mathbf{W}-\mathbf{M}_0)^\top \boldsymbol{\Lambda}_0(\mathbf{W}-\mathbf{M}_0)\}\big]\,, \\ p(\mathbf{R} \mid \mathbf{F}, \mathbf{W}) &= (2\pi\sigma^2)^{-np/2} \exp\!\big[-\tfrac{1}{2\sigma^2}\|\mathbf{R}-\mathbf{F}\mathbf{W}\|_F^2\big]\,. \end{aligned} \tag{22}$$

Define the posterior precision and information matrix by

$$\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_0 + \sigma^{-2}\mathbf{F}^\top\mathbf{F}, \qquad \mathbf{H} = \boldsymbol{\Lambda}_0\mathbf{M}_0 + \sigma^{-2}\mathbf{F}^\top\mathbf{R}, \qquad \mathbf{M} = \boldsymbol{\Lambda}^{-1}\mathbf{H}. \tag{23}$$

Expanding the quadratic terms gives

$$\begin{aligned}\sigma^{-2}\|\mathbf{R}-\mathbf{F}\mathbf{W}\|_F^2+\operatorname{tr}\{(\mathbf{W}-\mathbf{M}_0)^\top\mathbf{\Lambda}_0(\mathbf{W}-\mathbf{M}_0)\}\\ =\operatorname{tr}\{(\mathbf{W}-\mathbf{M})^\top\mathbf{\Lambda}(\mathbf{W}-\mathbf{M})\}+Q,\\ Q=\sigma^{-2}\operatorname{tr}(\mathbf{R}^\top\mathbf{R})+\operatorname{tr}(\mathbf{M}_0^\top\mathbf{\Lambda}_0\mathbf{M}_0)-\operatorname{tr}(\mathbf{M}^\top\mathbf{\Lambda}\mathbf{M}).\end{aligned} \tag{24}$$

The terms depending on $\mathbf{W}$ are therefore the kernel of $\mathcal{MN}(\mathbf{M},\mathbf{\Lambda}^{-1},\mathbf{I}_p)$, which proves Equation (5). Although $\mathbf{\Lambda}_0$ is diagonal, $\mathbf{F}^\top\mathbf{F}$ generally is not, so no diagonal posterior approximation follows from the specified model.

B.2 Integrating out the coefficients

The component evidence integrates over all dp coefficient entries:

$$\mathcal{E}_k=\int_{\mathbb{R}^{d\times p}}p(\mathbf{R}\mid\mathbf{F},\mathbf{W})p(\mathbf{W}\mid k)\,d\mathbf{W}. \tag{25}$$

The integral of the completed-square kernel is $(2\pi)^{dp/2}|\mathbf{\Lambda}|^{-p/2}$. Consequently,

$$\log\mathcal{E}_k=-\frac{np}{2}\log(2\pi\sigma^2)+\frac{p}{2}\left(\log|\mathbf{\Lambda}_0|-\log|\mathbf{\Lambda}|\right)-\frac{1}{2}Q. \tag{26}$$

This expression uses the sufficient statistics $\mathbf{F}^\top\mathbf{F}$, $\mathbf{F}^\top\mathbf{R}$, and $\operatorname{tr}(\mathbf{R}^\top\mathbf{R})$; the determinant exponent is $p/2$ because all p coefficient columns are integrated out.

An equivalent expression follows by marginalizing the linear Gaussian model directly:

$$\mathbf{R}\mid\mathbf{F},k\sim\mathcal{MN}(\mathbf{F}\mathbf{M}_0,\mathbf{C}_k,\mathbf{I}_p),\qquad\mathbf{C}_k=\sigma^2\mathbf{I}_n+\mathbf{F}\mathbf{S}_0\mathbf{F}^\top. \tag{27}$$

Thus, with $\mathbf{Z}_k=\mathbf{R}-\mathbf{F}\mathbf{M}_0$,

$$\log\mathcal{E}_k=-\frac{np}{2}\log(2\pi)-\frac{p}{2}\log|\mathbf{C}_k|-\frac{1}{2}\operatorname{tr}(\mathbf{Z}_k^\top\mathbf{C}_k^{-1}\mathbf{Z}_k). \tag{28}$$

The determinant identity $|\mathbf{C}_k|=(\sigma^2)^n|\mathbf{\Lambda}|/|\mathbf{\Lambda}_0|$ and the Woodbury identity establish equivalence with Equation (26). Posterior probabilities can be evaluated stably as

$$\gamma_{u,t,k}=\operatorname{softmax}_k(\log\pi_k+\log\mathcal{E}_{u,t,k}). \tag{29}$$

For an empty history, $\mathcal{E}_{u,t,k}=1$, $\gamma_{u,t,k}=\pi_k$, and each coefficient posterior equals its prior.

B.3 Predictive distribution and mean readout

For query feature $\mathbf{f}_*$, define $\boldsymbol{\mu}_k=\mathbf{M}_k^\top\mathbf{f}_*$ and $v_k=\mathbf{f}_*^\top\mathbf{S}_k\mathbf{f}_*$, where $\mathbf{S}_k=\mathbf{\Lambda}_k^{-1}$. Integrating out the coefficients gives

$$p(\mathbf{r}_*\mid\mathbf{f}_*,\mathcal{H})=\sum_{k=1}^{K}\gamma_k\mathcal{N}\big(\mathbf{r}_*;\boldsymbol{\mu}_k,(\sigma^2+v_k)\mathbf{I}_p\big)\,. \tag{30}$$

Its mean is $\mathbf{s}=\sum_k\gamma_k\boldsymbol{\mu}_k$, and its covariance is

$$\operatorname{Cov}(\mathbf{r}_*\mid\mathbf{f}_*,\mathcal{H})=\sum_k\gamma_k\left[(\sigma^2+v_k)\mathbf{I}_p+(\boldsymbol{\mu}_k-\mathbf{s})(\boldsymbol{\mu}_k-\mathbf{s})^\top\right]. \tag{31}$$

For the latent expected residual $\mathbf{W}^\top\mathbf{f}_*$, omit $\sigma^2\mathbf{I}_p$ from this covariance; its mean is the same. The prefix uses only that mean. The conditional mean of the modeled response representation is $f_0(x_*)+\mathbf{s}$, but the implemented LLM interface receives $\mathbf{s}$ through a learned prefix rather than adding it to token logits.

C Sequential Updates and Proof of Query Influence

All results in this section condition on fixed shared parameters, including the feature map, response representation, population predictor, observation variance, and mixture prior. Fix a component and let $(\mathbf{M},\mathbf{S})$ denote its current posterior parameters, with $\mathbf{S}\succ 0$. For a newly revealed interaction $(\mathbf{f},\mathbf{r})$, the information-form update is

$$(\mathbf{S}^+)^{-1}=\mathbf{S}^{-1}+\sigma^{-2}\mathbf{f}\mathbf{f}^\top,\qquad(\mathbf{S}^+)^{-1}\mathbf{M}^+=\mathbf{S}^{-1}\mathbf{M}+\sigma^{-2}\mathbf{f}\mathbf{r}^\top. \tag{32}$$

C.1 Covariance and mean updates

Applying the Sherman–Morrison identity gives

$$\mathbf{S}^+ = \mathbf{S} - \frac{\mathbf{S}\mathbf{f}\mathbf{f}^\top\mathbf{S}}{\sigma^2 + \mathbf{f}^\top\mathbf{S}\mathbf{f}} = \mathbf{S} - \mathbf{g}\mathbf{f}^\top\mathbf{S}, \qquad \mathbf{g} = \frac{\mathbf{S}\mathbf{f}}{\sigma^2 + \mathbf{f}^\top\mathbf{S}\mathbf{f}}. \tag{33}$$

Using $\mathbf{S}^+\mathbf{S}^{-1} = \mathbf{I}_d - \mathbf{g}\mathbf{f}^\top$ and $\sigma^{-2}\mathbf{S}^+\mathbf{f} = \mathbf{g}$ in Equation (32) yields

$$\mathbf{M}^+ = (\mathbf{I}_d - \mathbf{g}\mathbf{f}^\top)\mathbf{M} + \mathbf{g}\mathbf{r}^\top = \mathbf{M} + \mathbf{g}(\mathbf{r} - \mathbf{M}^\top\mathbf{f})^\top. \tag{34}$$

Moreover, for every $\mathbf{z} \in \mathbb{R}^d$,

$$\mathbf{z}^\top(\mathbf{S} - \mathbf{S}^+)\mathbf{z} = \frac{(\mathbf{z}^\top\mathbf{S}\mathbf{f})^2}{\sigma^2 + \mathbf{f}^\top\mathbf{S}\mathbf{f}} \geq 0, \tag{35}$$

which proves $\mathbf{S}^+ \preceq \mathbf{S}$.

C.2 Proof of Proposition 1

Proof. Transpose Equation (34) after subtracting $\mathbf{M}$, and multiply by $\mathbf{f}_*$:

$$(\mathbf{M}^+ - \mathbf{M})^\top\mathbf{f}_* = (\mathbf{r} - \mathbf{M}^\top\mathbf{f})\mathbf{g}^\top\mathbf{f}_* = \frac{\mathbf{f}_*^\top\mathbf{S}\mathbf{f}}{\sigma^2 + \mathbf{f}^\top\mathbf{S}\mathbf{f}}(\mathbf{r} - \mathbf{M}^\top\mathbf{f}). \tag{36}$$

At $\mathbf{f}_* = \mathbf{f}$, positive semidefiniteness gives $v \geq 0$, and $\sigma^2 > 0$ implies $0 \leq v/(\sigma^2 + v) < 1$. Covariance contraction follows from Equation (35). □

The interval bound applies at the observed input. At another query, $\mathbf{f}_*^\top\mathbf{S}\mathbf{f}$ may be negative and the multiplier need not lie in $[0, 1)$. Likewise, component-wise covariance contraction does not imply monotonic contraction of the entire mixture covariance: changing component probabilities also changes between-component uncertainty.

C.3 Component probabilities and batch–sequential equivalence

Before incorporating the new observation, component k assigns it predictive density

$$\ell_k = \mathcal{N}\big(\mathbf{r}; \mathbf{M}_k^\top\mathbf{f}, (\sigma^2 + \mathbf{f}^\top\mathbf{S}_k\mathbf{f})\mathbf{I}_p\big)\,. \tag{37}$$

Bayes' rule gives $\gamma_k^+ = \gamma_k\ell_k / \sum_j \gamma_j\ell_j$. Repeated application of Equation (32) gives

$$\begin{aligned} \mathbf{\Lambda}_{n,k} &= \mathbf{\Lambda}_{0,k} + \sigma^{-2}\sum_{i=1}^{n}\mathbf{f}_i\mathbf{f}_i^\top, \\ \mathbf{\Lambda}_{n,k}\mathbf{M}_{n,k} &= \mathbf{\Lambda}_{0,k}\mathbf{M}_{0,k} + \sigma^{-2}\sum_{i=1}^{n}\mathbf{f}_i\mathbf{r}_i^\top. \end{aligned} \tag{38}$$

These are exactly the batch posterior parameters. By the chain rule, the product of the sequential predictive densities is

$$\prod_{i=1}^{n} p(\mathbf{r}_i \mid \mathbf{f}_{1:i}, \mathbf{r}_{1:i-1}, k) = p(\mathbf{R} \mid \mathbf{F}, k) = \mathcal{E}_{n,k}. \tag{39}$$

Thus repeated component reweighting also equals the batch update $\gamma_{n,k} \propto \pi_k\mathcal{E}_{n,k}$. Both results hold in exact arithmetic; finite-precision implementations can differ numerically. Segment boundaries do not affect the posterior provided each observed interaction is included exactly once. Replacing a segment with its average generally changes the sufficient statistics and is not equivalent.

For completeness, the change in the mixture readout decomposes as

$$\mathbf{s}^+ - \mathbf{s} = \sum_k \gamma_k^+(\mathbf{M}_k^+ - \mathbf{M}_k)^\top\mathbf{f}_* + \sum_k(\gamma_k^+ - \gamma_k)\mathbf{M}_k^\top\mathbf{f}_*. \tag{40}$$

The first term reflects within-component coefficient adaptation; the second reflects changed component probabilities. At deployment, the fixed-parameter assumption holds. During training, shared-parameter updates can change all historical features and prior parameters, so posterior statistics must be recomputed under the current parameters to retain the stated batch interpretation.

Table 4: Sensitivity of BayUSI to the number of latent preference groups K. All metrics are higher-is-better. Bold indicates the best result within each metric and history setting.

		Evolving-history			Sparse-history		
Dataset	Metric	$K=1$	$K=4$	$K=8$	$K=1$	$K=4$	$K=8$
LaMP-2	Accuracy	**0.744**	0.732	0.735	**0.713**	0.696	0.712
	F1	0.709	**0.722**	0.709	**0.661**	0.656	0.660
LaMP-4	ROUGE-1	0.211	**0.217**	0.216	**0.208**	0.206	0.205
	ROUGE-L	0.191	**0.197**	0.195	0.189	**0.190**	0.187

Table 5: User-unit counts allocated to each preparation partition. Feature fitting and calibration support the construction of personal evidence, while Training is used for learning shared personalization parameters and Validation for checkpoint selection. Counts describe partition membership, not training examples or the number of units contributing optimization targets.

Dataset	Feature fitting	Calibration	Validation	Training	Total
LaMP-2	97	24	61	426	608
LaMP-3	3,200	800	2,000	14,000	20,000
LaMP-4	258	65	161	1,130	1,614
LaMP-5	1,549	387	968	6,776	9,680

D PPLUG-STYLE USER REPRESENTATION

We compute the user representation using a PPlug-style query-dependent attention aggregator over the available historical interactions. Let $f_{\text{hist}}(\cdot)$ denote a history encoder and $f_{\text{qry}}(\cdot)$ denote a query encoder. For each historical interaction $h_{u,i} \in \mathcal{H}_{u,<t}$, we compute

$$\mathbf{v}_{u,i} = f_{\text{hist}}(h_{u,i}), \qquad \mathbf{q}_{u,t} = f_{\text{qry}}(x_{u,t}). \tag{41}$$

The user representation is then obtained by query-dependent attention:

$$\mathbf{e}_{u,t} = \sum_{i<t} \alpha_{u,t,i} \phi_{\text{user}}(\mathbf{v}_{u,i}), \qquad \alpha_{u,t,i} = \frac{\exp(\mathbf{q}_{u,t}^\top \mathbf{v}_{u,i})}{\sum_{j<t} \exp(\mathbf{q}_{u,t}^\top \mathbf{v}_{u,j})}. \tag{42}$$

Here ϕ_{user} is a trainable user projection head, implemented as a lightweight MLP, that maps history-encoder representations into the personalization space. The history embeddings $\mathbf{v}_{u,i}$ preserve interaction details for query-dependent aggregation. They are distinct from the observed response residuals $\mathbf{r}_{u,i} = \mathbf{a}(y_{u,i}) - f_0(x_{u,i})$, which are paired with input features for Bayesian coefficient inference. The resulting history representation complements the posterior predictive residual $\mathbf{s}_{u,t}$ in conditioning the frozen LLM.

D.1 SENSITIVITY TO THE NUMBER OF PREFERENCE GROUPS

We examine the sensitivity of BayUSI to the number of latent preference groups, varying $K \in \{1, 4, 8\}$ on LaMP-2 and LaMP-4 under both history settings. With $K = 1$, BayUSI uses a single shared prior while retaining user-specific posterior coefficient updates.

Table 4 shows that the effect of K depends on the task and history setting. Under evolving history, $K = 4$ achieves the highest F1 on LaMP-2 and the highest ROUGE scores on LaMP-4, whereas $K = 1$ yields the highest LaMP-2 accuracy. Under sparse history, $K = 1$ remains competitive, achieving the highest LaMP-2 accuracy and F1 and LaMP-4 ROUGE-1, while $K = 4$ performs best on ROUGE-L. Increasing K to 8 provides no consistent improvement. These results suggest that multiple prior components can benefit some tasks as evidence accumulates, but increasing the number of groups alone does not guarantee better personalization.

D.2 Data Preparation and Split Statistics

Table 5 reports the realized partition sizes within the training data. We use separate partitions to fit the population predictor and frozen feature transformations, calibrate the observation noise, select model checkpoints, and optimize the shared personalization parameters. The two internal training subsets are combined in the Training column. Checkpoint validation is held out from parameter optimization; these counts do not include external evaluation users.

A user unit is the identity or dependency unit used by the preparation pipeline. Where authoritative user identifiers are unavailable, profile-overlap information is used to construct these units; they should therefore not be interpreted as verified counts of distinct individuals.