EXACT: EXPLICIT ATTRIBUTE-GUIDED DECODING-TIME PERSONALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized alignment requires adapting large language models to preferences that can vary with the current prompt. We introduce EXACT, a decoding-time personalization method that learns compact, explicit preference representations from limited pairwise feedback. Offline, EXACT selects a prompt-specific subset from a predefined attribute vocabulary by maximizing a baseline-anchored log-likelihood gap between preferred and dispreferred responses. Online, it retrieves the subset associated with the most semantically similar historical prompt and injects those attributes into the generation context. The method uses a single base generation model and a lightweight retrieval encoder, while exposing an inspectable personalization signal. We relate greedy attribute selection to classical approximation guarantees under normalized monotonicity and a positive submodularity ratio, and analyze pooling and retrieval in an idealized topic model. Experiments on two human-preference datasets and three model families show competitive preference modeling and LLM-judged personalized generation performance, with ablations examining attribute vocabulary and retrieval design.

1 INTRODUCTION

Large language models (LLMs) have become foundational to a wide range of real-world applications, driven in part by advances in reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019). Conventional RLHF pipelines aim to align models with population-level preferences by aggregating large-scale feedback from diverse users. While effective for general alignment, this paradigm overlooks substantial heterogeneity in individual user preferences, motivating growing interest in personalized alignment methods (Xu et al., 2025; Liu et al., 2025). Personalized LLMs seek to produce user-specific outputs while preserving contextual coherence, enabling applications such as personalized recommendation systems (Lyu et al., 2024; Zhang et al., 2024a; Yu et al., 2025), adaptive dialogue agents (Wu et al., 2025), and customized content generation platforms (Mysore et al., 2024).

Decoding-time personalization avoids parameter updates and is attractive when user feedback is sparse (Chen et al., 2025; Kim et al., 2025a). Prompt-based methods include cue-guided reasoning (Wang et al., 2023), task-aware history summarization and retrieval (Richardson et al., 2023), and CIPHER's retrieval of preference descriptions inferred from user edits (Gao et al., 2024a). These approaches can expose readable personalization signals. Other methods steer generation probabilities: PAD uses reward-guided decoding (Chen et al., 2025), while Drift composes interpretable attributes using user-specific weights (Kim et al., 2025a). EXACT instead selects discrete attribute subsets from pairwise feedback and retrieves a subset for each incoming prompt, avoiding attribute-wise logit composition at generation time.

Constructive preference theory emphasizes that preferences depend on the decision context (Slovic, 1995). In LLM personalization, PAD discusses scenario-dependent preferences (Chen et al., 2025), CIPHER retrieves preferences from similar contexts (Gao et al., 2024a), and CUPID evaluates contextual preference inference from interactions (Kim et al., 2025b). We refer to such variation as *contextual preference shifts*; the draft illustrations in Table 1 await source verification. Building on prior observations, we study how to select compact attribute subsets from pairwise comparisons and retrieve them according to prompt similarity.

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.

Dataset (User)	Prompt (Title)	Response A	Response B	Style
PRISM `RgH765...`	**P-A:** *Me [26M] with my fiancee... wedding invitations*	✓ *"Her parents printed the wedding invitation card and want her to hide it from me."* (*neutral restatement*)	× *"...hide the wedding card... because of my in-laws' hate towards me."*	**Neutral factual**
PRISM `RgH765...`	**P-B:** *Insurance company's lawyer seeking reimbursement?*	✓ *"Neighbor wants money... previous owner caused damage... neighbor's insurance paid... now wants grandma to pay back. Our insurance is basically MIA."* (*context-rich narrative*)	× *"Neighbor's insurer asked for reimbursement... wants $1600..."* (*minimal facts*)	**Context rich narrative**
Summarize from Human Feedback `user5`	**T1:** *I want to cook lasagne...*	× *"There are many great recipes... it depends on personal preference."*	✓ *"Ingredients: ... Instructions: 1. ... 2. ..."*	**Structured step-based**
Summarize from Human Feedback `user5`	**T2:** *Last-minute cramming tips?*	✓ *"Prioritize key topics, practice active recall, use mnemonics, take breaks, and sleep."* (*flowing narrative*)	× *"1. Focus on core concepts... 2. Use mnemonics..."*	**Flowing narrative**

Table 1: **Draft illustrations of contextual preference shifts (provenance pending).** The examples contrast neutral restatement with context-rich narrative (P-A/P-B), and structured formatting with flowing advice (T1/T2). Dataset assignments and preference labels are retained from the earlier draft and require source-record verification before these cases can serve as empirical evidence.

We propose **EXACT** (**EX**plicit **A**ttribute-guided de**C**oding-**T**ime personalization), an interpretable decoding-time method motivated by KL-regularized RLHF. Offline, EXACT selects prompt-specific preference attributes from pairwise feedback using the likelihood of preferred and dispreferred responses. Online, it retrieves the attribute subset associated with the most semantically similar historical prompt and injects it into the generation context. This yields a compact, inspectable preference representation and requires only a single base generation model, together with a lightweight retrieval encoder. Our analysis relates greedy selection to established approximation guarantees and studies retrieval under an idealized topic model.

Our contributions are summarized as follows:

- We study *contextual preference shifts* through a prompt-conditioned attribute-selection formulation, building on prior observations of context-dependent preferences.
- We propose EXACT, a decoding-time personalization method that learns explicit preference attributes from pairwise feedback and retrieves a compact, inspectable subset for each incoming prompt.
- We evaluate EXACT on two datasets and three backbones using pairwise ranking accuracy and LLM-judged generation comparisons, with ablations examining attribute vocabulary and retrieval design.

2 METHODOLOGY

Our method follows a two-stage pipeline: Offline Indexing in Section 2.1, and Online Inference in Section 2.2. We defer the RLHF preliminaries and derivation used to motivate our objective to Appendix B.1 and Appendix B.2.

2.1 LEARNING USER PREFERENCES VIA PREFERENCE ATTRIBUTES

We study *decoding-time* personalization under a fixed base large language model (LLM) π_{base}. Let x denote a user prompt and $\mathcal{A} = \{a_1, \ldots, a_K\}$ be a predefined set of semantic attributes, following

the attribute design in Drift (Kim et al., 2025a). Each attribute captures a distinct aspect of user preference or stylistic intent (e.g., conciseness, vividness, formality) (see Table 4 for the full list). For convenience, we group these attributes into four coarse categories: *style* (e.g., *Formal*, *Concise*), *tone* (e.g., *Direct*, *Empathetic*), *expertise* (e.g., *Analytic*, *Code*), and *values* (e.g., *Principled*, *Utilitarian*). Attributes are not mutually exclusive and can be correlated (e.g., *Direct* often co-occurs with *Concise*). We model each user by an *unknown* subset of attributes that governs their preference over generated responses.

Given a prompt x, a user (or an annotator acting on the user's behalf (Lee et al., 2024)) provides a preferred response y_w and a dispreferred response y_l. Following standard RLHF, we assume these preferences arise from a latent reward function $r^*(x, y)$, which induces a human preference distribution $p^*(y_w \succ y_l \mid x)$ (see Appendix B.1).

Our goal is to infer user-specific preference attributes *at decoding-time* and use them to steer personalized generation *without* updating model parameters.

Attribute-guided prompt modification. Let $\pi(\cdot)$ denote the base language model. Given a set of selected attributes $A \subseteq \mathcal{A}$, we construct an attribute-augmented prompt by concatenating the original prompt with natural-language descriptions of the attributes,

$$x^A = (x, A),$$

where we modify the user prompt by appending an explicit attribute block. Specifically, after the original user query, we add a new line that starts with the keyword "Attributes:", followed by a comma-separated list of the selected attribute subset (e.g., Direct, Concise, Analytic). Appendix E.3.2 contains draft prompt-format illustrations whose source records and vocabulary mapping remain to be verified. This yields a modified conditional policy

$$\pi^A(y \mid x) \;:=\; \pi(y \mid x^A) = \pi(y \mid x, A). \tag{1}$$

Importantly, this modification is performed purely at inference time and does not alter the model parameters.

Optimization objective from human preferences. Applying Eq. (13) to the attribute-guided policy π^A, we obtain the induced reward

$$r^A(x, y) = \beta \log \frac{\pi^A(y \mid x)}{\pi(y \mid x)} + \beta \log Z^A(x). \tag{2}$$

Substituting into the Bradley-Terry model yields

$$p^*(y_w \succ y_l \mid x) = \frac{1}{1 + \exp\Big(\beta \log \frac{\pi^A(y_l|x)}{\pi(y_l|x)} - \beta \log \frac{\pi^A(y_w|x)}{\pi(y_w|x)}\Big)} \propto \log \pi^A(y_w \mid x) - \log \pi^A(y_l \mid x), \tag{3}$$

where the last line follows from the monotonicity of the logistic function, and we only retain the terms that depend on the induced policy π^A. Constants and reference-policy terms that do not affect the ordering are omitted. Therefore, learning user-specific personalization reduces to selecting an attribute set A such that the induced policy π^A maximizes the likelihood of the observed human preference data. We can find the most important ones for user preference via

$$\max_{A \subseteq \mathcal{A}} \quad \log \pi(y_w \mid x, A) - \log \pi(y_l \mid x, A). \tag{4}$$

For each preference pair, we use greedy search to select a size-k attribute subset from K candidates (Algorithm 1), reducing the number of objective evaluations from $\mathcal{O}(2^K)$, or $\mathcal{O}(\binom{K}{k})$ for size-constrained search, to $\mathcal{O}(Kk)$; the method also admits a classical approximation guarantee under normalized monotonicity and a positive submodularity ratio (Section 3.1). To construct the prompt–attribute index, we perform prompt-level deduplication and retain for each repeated prompt only the subset with the largest objective value in Eq. (4), yielding a unique representative attribute set for each distinct prompt in $\mathcal{D}$.

Incremental vocabulary expansion. When new attributes become available after the offline indexing stage, EXACT can extend the attribute library without retraining the base model: for a prompt x, first retrieve the current attribute subset A from history, then treat each new candidate attribute a as a plug-in and accept it only if adding a improves the preference objective in Eq. (4) on the available preference pairs; otherwise, discard it. Appendix D.3 gives a hypothetical example based on formatting preferences.

Algorithm 1 Attribute Selection for Each Prompt (Per User)

input Attribute library $\mathcal{A} = \{a_1, \ldots, a_K\}$; budget k; preference pairs $\mathcal{D} = \{(x^{(n)}, y_w^{(n)}, y_l^{(n)})\}_{n=1}^N$; base model π.
output Per-prompt subsets $\{A^\star(x^{(n)})\}_{n=1}^N$ with $|A^\star(x^{(n)})| \le k$.

for $n \in [N]$ **do**
  $A \leftarrow \emptyset$
  for $t \in [k]$ **do**
    $a^\star \leftarrow \arg\max_{a \in \mathcal{A} \setminus A} \left[\log \pi\big(y_w^{(n)} \mid x^{(n)}, A \cup \{a\}\big) - \log \pi\big(y_l^{(n)} \mid x^{(n)}, A \cup \{a\}\big)\right]$
    $A \leftarrow A \cup \{a^\star\}$
  end for
  $A^\star(x^{(n)}) \leftarrow A$
end for

2.2 Attribute retrieval at inference time

Motivated by context-dependent preferences, EXACT adapts its attribute representation to each new prompt via retrieval rather than using a single static attribute profile.

For a fixed user, suppose we have observed s historical prompts $\{x_i\}_{i=1}^s$, each paired with an inferred attribute subset $A_i \subseteq \mathcal{A}$. Given a new prompt x, we first encode prompts into normalized embeddings using the retrieval encoder and compute cosine similarity in the embedding space. We retrieve the most relevant historical prompt by

$$i^* = \arg\max_{i \in \{1,\ldots,s\}} \cos(x, x_i),$$

and use the corresponding attributes A_{i^*} to guide generation for x. We summarize the overall inference-time retrieval and attribute-injection procedure in Algorithm 2. This makes the injected subset prompt-dependent; whether semantic neighbors transfer useful preferences is an empirical question.

Optional constraint-aware extension. Semantically similar prompts can impose conflicting constraints (e.g., "two sentences" or "three bullet points"). A possible extension would parse explicit constraints and filter incompatible history before similarity search. The reported experiments do not isolate this extension, so they do not establish a reduction in constraint violations.

When new preference feedback becomes available, we infer an attribute set A_{s+1} for the new prompt x_{s+1} and append (x_{s+1}, A_{s+1}) to the user memory. This update adds new feedback to the retrieval memory without changing model parameters; its cost includes offline attribute selection for the new pair.

Algorithm 2 Inference-Time Attribute Retrieval and Personalized Generation

input User memory $\mathcal{M} = \{(x_i, A_i)\}_{i=1}^s$; new prompt x; embedding function $\phi(\cdot)$; base model π.
output Personalized response $\hat{y}$.

$i^\star \leftarrow \arg\max_{i \in \{1,\ldots,s\}} \cos(\phi(x), \phi(x_i))$
$\hat{y} \leftarrow \pi(\cdot \mid x, A_{i^\star})$
return $\hat{y}$ (and $\mathcal{M}$)

Computation and latency. Our method separates personalization into an *offline* indexing stage (Algorithm 1) and an *online* inference stage (Algorithm 2). In the offline stage, we infer a prompt-specific attribute subset for each historical preference pair and construct a retrieval index. This

one-shot cost can be amortized across future queries and updated incrementally as new feedback arrives. At inference time, personalization adds only two lightweight steps: (i) computing an embedding for the incoming prompt and (ii) performing prompt-similarity-based retrieval, after which generation proceeds with the base model. The reported embedding and retrieval time accounts for 5.27% of total online inference time in the main experimental setting. This fraction excludes offline attribute selection and does not establish an end-to-end cost advantage over RAG.

3 Theoretical Analysis

We provide theory for two design choices in EXACT: greedy subset selection and prompt-level retrieval. Full assumptions and proofs are deferred to Appendix D and Appendix D.2.

3.1 Greedy attribute selection

We formulate attribute selection as a cardinality-constrained set maximization. Let $\mathcal{A} = \{a_1, \ldots, a_K\}$ be the attribute library and define

$$F(A) := \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log \frac{\pi(y_w \mid x, A)}{\pi(y_w \mid x)} - \log \frac{\pi(y_l \mid x, A)}{\pi(y_l \mid x)}\Big]. \tag{5}$$

Our goal is to select at most k attributes:

$$\max_{A\subseteq\mathcal{A},\ |A|\le k} F(A). \tag{6}$$

Greedy selection requires only $\mathcal{O}(Kk)$ objective evaluations, compared with exhaustive search over all subsets. Under normalized monotonicity and weak submodularity with submodularity ratio $\gamma \in (0, 1]$ (Appendix D), it inherits the classical constant-factor guarantee of Das & Kempe (2018, Theorem 6), restated below for our setting.

Theorem 3.1 (Greedy guarantee under weak submodularity)**.** *Let S_k be the output of greedy selection, and let $S^* \in \arg\max_{|A|\le k} F(A)$ be an optimal solution to* (6)*. If F is normalized and monotone and has a submodularity ratio γ, then*

$$F(S_k) \ \ge\ \big(1 - e^{-\gamma}\big)\, F(S^*). \tag{7}$$

A full statement of the assumptions, the definition of γ, and the complete proof of Theorem 3.1 are provided in Appendix D. To link Theorem 3.1 to practice, Table 7 in Appendix E.5.1 compares the output of greedy selection S_k with the exhaustive optimum S^* on a real dataset at k=3. Empirically, greedy achieves a high ratio $F(S_k)/F(S^*)$ with a small gap.

3.2 Contextual preference shifts across topics

We model contextual preference shifts using an idealized multi-topic mean-estimation problem. Intuitively, pooling estimates a single global center for all prompts, whereas retrieval selects a topic-relevant center; the full setup is given in Appendix D.2.

Theorem 3.2 (Retrieval mitigates contextual preference shifts)**.** *Let $\phi(x) = \mu_{t(x)} + \varepsilon$ with $\mathbb{E}[\varepsilon] = 0$, and assume topic centers satisfy $\|\mu_j - \mu_\ell\|_2 \ge r$ for all $j \ne \ell$. Let $\widehat{\mu}_{\mathrm{pool}} = \frac{1}{m_1}\sum_{i=1}^{m_1}\phi(x_i)$ denote global pooling, and let retrieval form topic-wise centers $\widehat{\mu}_c = \frac{1}{n_c}\sum_{i:t(x_i)=c}\phi(x_i)$ and return $\widehat{\mu}_{\mathrm{ret}}(x) = \widehat{\mu}_{\widehat{j}(x)}$. Let $\Sigma_j = \mathrm{Cov}(\varepsilon \mid t(x) = j)$ and assume $\mathrm{tr}(\Sigma_j) \le d\sigma^2$. For a test prompt x with $t(x) = j$, define the mixture mean $\bar{\mu} = \sum_{c=1}^{k} p_c\mu_c$. Then pooling satisfies*

$$\mathbb{E}\big[\|\widehat{\mu}_{\mathrm{pool}} - \mu_j\|_2^2\big] = \|\bar{\mu} - \mu_j\|_2^2 + \frac{1}{m_1}\sum_{c=1}^{k} p_c\,\mathrm{tr}(\Sigma_c), \tag{8}$$

hence it can incur an $\mathcal{O}(r^2)$ drift bias for some topics.

Moreover, if ε is σ^2-subGaussian and

$$n_{\min} \triangleq \min_{c\in[k]} n_c \;\geq\; C \cdot \frac{\sigma^2}{r^2}\Big(d + \log\frac{k}{\delta}\Big), \tag{9}$$

then with probability at least $1-\delta$, retrieval identifies the correct topic and its difference is variance-dominated:

$$\mathbb{E}\big[\|\widehat{\mu}_{\mathrm{ret}}(x) - \mu_j\|_2^2\big] \;\leq\; \frac{\mathrm{tr}(\Sigma_j)}{n_j} + 4\delta r^2 \;\leq\; \frac{d\sigma^2}{n_j} + 4\delta r^2. \tag{10}$$

Draft theory verification. This idealized model retrieves topic centers, whereas Algorithm 2 retrieves individual historical prompts. The variance decomposition and query-noise argument require correction before this theorem can be used as a guarantee for EXACT. Appendix E.5.3 separately describes the retrieval ablation and its metric verification.

4 Experiments

Following previous work (Kim et al., 2025a), we evaluate our method on two tasks: 1) efficient few-shot preference modeling and 2) personalized generation. In Section 4.1, we discuss the datasets and experimental setup. In Section 4.2, we evaluate different approaches by how closely the approximation matches the user's preference choice. In Section 4.3, to further evaluate personalization alignment for each user, LLM-as-a-judge is applied to report the win rate of each method over generation from the pure base LLM. Finally, in Section 4.5, we provide detailed case studies for EXACT.

4.1 Data and experiment setup

Datasets. We evaluate on two human-preference datasets. ***PRISM*** (Kirk et al., 2024) contains user feedback on LLM conversations; we use its pairwise preference setting, following Drift (Kim et al., 2025a), which reports about 19.5 preference examples per user. ***Summarize from Human Feedback*** (Stiennon et al., 2020) is a summarization benchmark built from TL;DR Reddit posts (Völske et al., 2017), where annotators compare two model-generated summaries for the same post. Together, these datasets cover open-ended generation and summarization under pairwise supervision.

Data processing. To account for randomness in user sampling, we repeat the entire sampling procedure with *three* different random seeds, each producing an independent set of selected users. For the accuracy results in Table 2, we report the mean and sample standard deviation across these three runs; generation judgments use the aggregation stated separately below. For ***PRISM***, we randomly sample 10 users, and each selected user has about 20 preference pairs on average. For ***Summarize From Human Feedback***, the raw data contains a large number of users with highly imbalanced numbers of pairs. We sample 10 users whose per-user preference pairs fall in the range [20, 100] to ensure comparable supervision across users.

Hyperparameters. For each user, we randomly shuffle all pairs and split them into training and test sets using an 8:2 ratio. Across both datasets and all three reported random seeds, we verified that no prompt/post appears in both the training and test sets for the same user (0% prompt overlap). We use an attribute library (See Table 4 in Appendix E.3.1) of size $K = 42$ and select exactly k attributes per training instance via greedy selection, with the default $k = 3$. For generations, we use temperature 0.7 and top-p 0.95 sampling, with `max_new_tokens`$= 200$ for PRISM and `max_new_tokens`$= 300$ for Summarize from Human Feedback; full decoding details are provided in Appendix E.2.

Base models. Prior work on decoding-time personalization (e.g., Drift; Kim et al., 2025a) typically evaluates on open, instruction-tuned backbones such as *Llama-3.1-8B* (Grattafiori et al., 2024) and *Gemma-2-9B-it* (Team et al., 2024), and may use smaller variants (e.g., *Llama-3.2-1B* or *Gemma-2-2B-it*) as the *small* models required by the Drift pipeline. We adopt comparable open-weight base LLMs for fair comparison, and additionally include *Qwen2.5-7B-Instruct* (Qwen et al., 2025) as an extra base model to broaden coverage across model families (with *Qwen2.5-0.5B-Instruct* as its corresponding *small* model when needed).

Dataset	Explicit attributes	Method	Llama-3.1 8B	Gemma-2 9B-it	Qwen2.5 7B-Instruct
PRISM	No	Base	57.22 (0.21)	59.27 (0.16)	54.83 (0.11)
		RAG few-shot	<u>65.21</u> (0.21)	**66.31** (0.23)	<u>65.34</u> (0.34)
		PAD	59.32 (0.12)	61.24 (0.21)	58.83 (0.15)
		Reward	54.78 (0.11)	58.91 (0.23)	49.58 (1.02)
	Yes	Drift	58.96 (0.47)	61.31 (0.54)	56.16 (0.28)
		EXACT	**66.62** (0.98)	<u>65.93</u> (0.15)	**66.41** (0.64)
Summarize from Human Feedback	No	Base	57.54 (0.18)	63.58 (0.18)	54.04 (0.12)
		RAG few-shot	**66.21** (0.35)	64.12 (0.23)	59.21 (0.33)
		PAD	60.12 (0.18)	<u>64.34</u> (0.12)	56.73 (0.13)
		Reward	54.21 (0.15)	54.33 (0.13)	**60.75** (0.21)
	Yes	Drift	58.68 (0.13)	58.21 (0.12)	55.63 (0.16)
		EXACT	<u>65.12</u> (0.14)	**66.01** (0.13)	<u>60.28</u> (0.15)

Table 2: Personalized modeling results: mean user-macro accuracy (%) and sample standard deviation over three random seeds, reported separately for each dataset and backbone. "Explicit attributes" indicates whether a method provides an explicit preference representation. Within each dataset and backbone column, the best result is shown in **bold** and the second-best in <u>underline</u>.

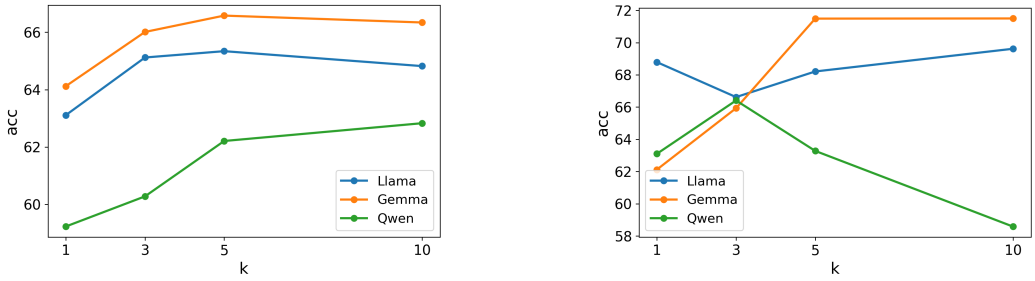
Baselines and our methods. We compare the following approaches, using the same generation backbone and data splits within each comparison, with shared sampling parameters and method-specific personalization components. Further experimental details (compute environment, decoding configuration, and baseline implementations) are described in Appendix E.2. **Base** is a naive history-injection baseline that prepends the user's available training history (prompts and preference choices) as context before generation. **RAG few-shot** retrieves a small set of the most similar historical preference pairs and injects them as few-shot context. **Drift** (Kim et al., 2025a) is a decoding-time personalization baseline that represents each user with an explicit attribute-weight vector and composes attributes via a *linear combination* to form a personalized control signal at inference time. **PAD** (Chen et al., 2025) steers generation via reward-guided decoding. **Reward** (RM) is a personalized reward-model baseline that learns a personalized reward model from each user's pairwise preferences (via a per-user LoRA adapter) and uses it at test time to score candidate responses. **EXACT** retrieves the most relevant attributes from the user's history and injects the selected subset into the prompt for personalized generation. Because EXACT is retrieval-based and prompt-conditioned, it can also be combined with external decoding-time steering modules such as Context Steering (CoS) (He et al., 2025); Appendix E.5.1 reports an additional PRISM ablation for EXACT+CoS and RAG few-shot+CoS.

4.2 Personalization modeling accuracy

For method m, let $s_m(x, y; H_u)$ be the score assigned to response y using user u's training history H_u. A test pair is correct only if its preferred response y_w scores strictly higher than y_l:

$$\mathrm{Acc}_u(m) = \frac{1}{|T_u|} \sum_{(x,y_w,y_l)\in T_u} \mathbf{1}[s_m(x, y_w; H_u) > s_m(x, y_l; H_u)]\,, \tag{11}$$

where T_u is the user's test set and ties count as incorrect. For EXACT, $s_m(x, y; H_u) = \log \pi(y \mid x, A_u(x))$, using the same retrieved attributes for both candidates. Scores sum response-token log probabilities without length normalization; reward-model baselines use their scalar reward scores. We macro-average over users within each run and report the mean and sample standard deviation over three seeds. An improvement relative to the base-model gap does not count as a correct ranking. Appendix E.2 specifies the scoring protocol. For similarity-based retrieval, we compared multiple



(a) Summarize from Human Feedback.

(b) PRISM.

Figure 1: **EXACT with different k.** Pairwise accuracy (%) across three backbones as the attribute budget varies. Trends depend on the dataset and backbone: Qwen declines on PRISM beyond k=3, while increasing on Summarize from Human Feedback.

Dataset	Method	Gemini-3-Flash	GPT-4o
PRISM	Drift	58.65	57.75
	Reward	59.23	57.81
	PAD	57.99	57.20
	RAG few-shot	<u>61.66</u>	**63.43**
	EXACT	**62.33**	<u>63.18</u>
Summarize from Human Feedback	Drift	67.84	68.40
	Reward	67.64	66.79
	PAD	65.02	63.79
	RAG few-shot	<u>73.13</u>	<u>70.80</u>
	EXACT	**74.72**	**70.99**

Table 3: **Win-rate (%) of personalized generation with two LLM judges.** Win-rates against the *pure base LLM* generations on PRISM and Summarize from Human Feedback, averaged over backbones. Within each dataset and judge column, the best result is shown in **bold** and the second-best in <u>underline</u>. Response order is randomly shuffled before judgment.

embedding models (Appendix E.2.6) and used BGE-small from the BGE family (Xiao et al., 2024) in our main experiments.

Table 2 reports the main personalization-modeling results across two datasets and three backbones. The reported values place EXACT first in three of six dataset–backbone combinations. RAG few-shot leads on PRISM with Gemma and on Summarize from Human Feedback with Llama, while Reward leads on the latter dataset with Qwen. Thus the table supports competitive, rather than uniform, superiority. EXACT exposes a compact attribute representation; Appendix E.2.7 discusses its tradeoff with raw-history RAG in representation, prompt length, and offline computation.

4.3 Personalized Generation Quality

We assess personalized generation with two LLM judges, Gemini-3-Flash (Doshi, 2025) and GPT-4o (Hurst et al., 2024). Each judge receives the user's training preference pairs (historical prompt, preferred response, and dispreferred response), the test prompt, and two candidate generations in randomly shuffled order. It selects the response that better matches the user's inferred preferences. Table 3 reports win-rates against pure base-LLM generations, averaged over the three backbones.

EXACT has the highest reported win-rate on PRISM under Gemini-3-Flash, while RAG few-shot leads slightly under GPT-4o. On Summarize from Human Feedback, EXACT leads under both judges, with RAG close behind. These proxy judgments suggest competitive personalization but do not establish human preference or a direct EXACT–RAG advantage: each method is compared separately with the base LLM.

Human-evaluation protocol and status. Appendix F specifies a blinded EXACT–RAG study, originally targeting 100 prompts per dataset and three annotators per prompt with Llama-3.1-8B. It separates history-conditioned preference fit from task quality. Aggregate comparison counts have since been received, but study metadata and item-level records remain to be reconciled before reporting a completed human-study result.

4.4 Ablation study

We report additional ablations on algorithmic stability, retrieval and selection design, and vocabulary robustness; detailed results are provided in Figure 1 and Appendix E.5.

Algorithmic stability and modularity. Figure 1 shows that budget sensitivity depends on the backbone and dataset, including a substantial decline for Qwen on PRISM beyond $k = 3$. Appendix E.5.1 compares EXACT+CoS with RAG+CoS; compatibility with steering does not imply improvement over standalone EXACT on every backbone.

Retrieval and selection design. We analyze greedy vs. exhaustive selection (Appendix E.5.1) and multi-neighbor retrieval (Appendix E.5.3). The reported comparisons favor greedy selection as a practical approximation and top-1 over naive neighbor aggregation. The retrieval vs. non-retrieval comparison requires metric verification (Appendix E.5.3).

Vocabulary robustness and extensibility. We examine robustness to perturbations of the predefined attribute library and extensions beyond the fixed vocabulary. Retaining 80% of the vocabulary lowers reported accuracy by 0.07–0.89 percentage points across the six settings (Appendix E.5.2). Incremental expansion improves the reported PRISM results, but also permits larger subsets; isolating vocabulary coverage requires a matched-budget control (Appendix E.5.2).

4.5 Case Study: Topic-Dependent Preference Modes

The draft case record for user `thott7Xe` contains 8 indexed prompts and 10 test prompts. Retrieval selects two history items $6/10$ and $4/10$ times, with distinct selected subsets: {Formal, Concise, Principled} and {Concise, Emotion, Base}. This illustrates variation in the model's retrieved representation, not independent evidence of a true preference shift. Dataset provenance remains unresolved: this case was labeled Summarize from Human Feedback, whereas Appendix E.5.1 assigns the matching user-ID prefix to PRISM. Further draft cases appear in Appendices E.4.1 and E.6.

5 Conclusion

EXACT learns explicit preference attributes from pairwise feedback and retrieves context-relevant subsets for decoding-time personalization. Reported experiments on two datasets and three backbones indicate competitive preference ranking and LLM-judged generation performance. The representation is inspectable, while human preference fit and attribute faithfulness require separate evidence.

AI USE STATEMENT

We use large language models as the objects of personalization and as components of the experimental evaluation. Llama-3.1-8B, Gemma-2-9B-it, and Qwen2.5-7B-Instruct generate responses under the methods described in Section 4.1; Gemini-3-Flash and GPT-4o serve as automated judges of personalized generation in Section 4.3. These judgments are evaluation proxies, with limitations discussed in Appendix A.1. OpenAI Codex assisted with adapting the manuscript to the ICLR 2027 LaTeX template, checking submission requirements and literature sources, revising citation-related wording, clarifying evaluation definitions and theoretical assumptions, designing and drafting the planned human-evaluation protocol, checking consistency across the manuscript and reports, summarizing author-supplied aggregate counts, and drafting the submission statements. The authors are responsible for the final text, claims, and artifacts.

ETHICS STATEMENT

The reported experiments use existing human-preference datasets. Appendix F retains the original human-evaluation protocol and notes received aggregate comparisons. The actual study procedures, rater identities, recruitment, informed consent, compensation, and applicable institutional review determination remain to be documented; these facts cannot be inferred from the counts. Prompt histories and inferred preference attributes can contain sensitive information; deployment should minimize retained data, restrict access, and allow users to inspect, correct, or remove their personalization information. Personalized generation can amplify bias or reinforce harmful preferences, so attribute-based steering should remain subject to appropriate safety constraints. Automated judgments and third-party assessments conditioned on history do not establish the original users' satisfaction or safety in deployment.

REPRODUCIBILITY STATEMENT

Section 2 and Algorithms 1–2 specify offline attribute selection and online retrieval. Section 4.1 describes the datasets, data splits, models, and principal hyperparameters, while Appendix E.2 provides compute, decoding, and method-specific settings. The attribute vocabulary and prompt examples appear in Appendices E.3.1 and E.3.2. Assumptions and proofs for the theoretical results are presented in Appendices D and D.2; additional ablations appear in Appendix E.5. Appendix F specifies the planned human-evaluation protocol and records to retain. This manuscript package includes aggregate comparison counts but no executable experiment code or item-level human ratings. Draft tables and cases with unresolved metric or provenance conflicts are marked for reconciliation.

REFERENCES

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Andrew An Bian, Joachim M Buhmann, Andreas Krause, and Sebastian Tschiatschek. Guarantees for greedy maximization of non-submodular functions with applications. In *International conference on machine learning*, pp. 498–507. PMLR, 2017.

Avinandan Bose, Zhihan Xiong, Yuejie Chi, Simon Shaolei Du, Lin Xiao, and Maryam Fazel. LoRe: Personalizing LLMs via low-rank reward modeling. *arXiv preprint arXiv:2504.14439*, 2025.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. PAD: Personalized alignment of LLMs at decoding-time. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=e7AUJpP8bV.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Abhimanyu Das and David Kempe. Submodular meets spectral: greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning*, pp. 1057–1064, 2011.

Abhimanyu Das and David Kempe. Approximate submodularity and its applications: Subset selection, sparse approximation and dictionary selection. *Journal of Machine Learning Research*, 19(3): 1–34, 2018. URL https://jmlr.org/papers/v19/16-534.html.

Tulsee Doshi. Gemini 3 Flash: Frontier intelligence built for speed. Google, on behalf of the Gemini team, December 2025. URL https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/.

Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. Aligning LLM agents by learning latent preference from user edits. In *Advances in Neural Information Processing Systems*, volume 37, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f75744612447126da06767daecce1a84-Abstract-Conference.html.

Songyang Gao, Qiming Ge, Wei Shen, Shihan Dou, Junjie Ye, Xiao Wang, Rui Zheng, Yicheng Zou, Zhi Chen, Hang Yan, et al. Linear alignment: A closed-form solution for aligning human preferences without tuning and feedback. In *International Conference on Machine Learning*, pp. 14702–14722. PMLR, 2024b.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Seungwook Han, Idan Shenfeld, Akash Srivastava, Yoon Kim, and Pulkit Agrawal. Value augmented sampling for language model alignment and personalization. *arXiv preprint arXiv:2405.06639*, 2024.

Jerry Zhi-Yang He, Sashrika Pandey, Mariah L. Schrum, and Anca Dragan. Context steering: Controllable personalization at inference time. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/db178cd03313e23cffb8937e93f0d464-Abstract-Conference.html.

James Y Huang, Sailik Sengupta, Daniele Bonadiman, Yi-an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. DeAL: Decoding-time alignment for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 26280–26300, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. GPT-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. ARGS: Alignment as reward-guided search. In *The Twelfth International Conference on Learning Representations*, 2024.

Minbeom Kim, Kang-il Lee, Seongho Joo, Hwaran Lee, Thibaut Thonet, and Kyomin Jung. Drift: Decoding-time personalized alignments with implicit user preferences. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 6107–6126, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.324. URL https://aclanthology.org/2025.findings-emnlp.324/.

Tae Soo Kim, Yoonjoo Lee, Yoonah Park, Jiho Kim, Young-Ho Kim, and Juho Kim. CUPID: Evaluating personalized and contextualized alignment of LLMs from interactions. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=JMxRn7orEk.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. *Advances in Neural Information Processing Systems*, 37:105236–105344, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/be2e1b68b44f2419e19f6c35a1b8cf35-Abstract-Datasets_and_Benchmarks_Track.html.

Xiaoyu Kong, Jiancan Wu, An Zhang, Leheng Sheng, Hui Lin, Xiang Wang, and Xiangnan He. Customizing language models with instance-wise LoRA for sequential recommendation. *Advances in Neural Information Processing Systems*, 37:113072–113095, 2024.

Seongyun Lee, Sue Hyun Park, Seungone Kim, and Minjoon Seo. Aligning to thousands of preferences via system message generalization. *Advances in Neural Information Processing Systems*, 37: 73783–73829, 2024.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023.

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

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. Decoding-time realignment of language models. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 31015–31031, 2024.

Hang Lv, Sheng Liang, Hao Wang, Hongchao Gu, Yaxiong Wu, Wei Guo, Defu Lian, Yong Liu, and Enhong Chen. CoSteer: Collaborative decoding-time personalization via local delta steering. *arXiv preprint arXiv:2507.04756*, 2025.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-Rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 583–612, 2024.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. Controlled decoding from language models. In *Forty-first International Conference on Machine Learning*, 2024.

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 *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, 2024.

Dan Peng, Zhihui Fu, and Jun Wang. PocketLLM: Enabling on-device fine-tuning for personalized LLMs. In *Proceedings of the Fifth Workshop on Privacy in Natural Language Processing*, pp. 91–96, 2024.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025. URL https://arxiv.org/abs/2412.15115v2. Version 2, January 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

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

Stephanie Schoch, Diyi Yang, and Yangfeng Ji. "this is a problem, don't you agree?" framing and bias in human evaluation for natural language generation. In *Proceedings of the 1st Workshop on Evaluating NLG Evaluation*, pp. 10–16. Association for Computational Linguistics, 2020. URL https://aclanthology.org/2020.evalnlgeval-1.2/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Paul Slovic. The construction of preference. *American psychologist*, 50(5):364, 1995.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. Democratizing large language models via personalized parameter-efficient fine-tuning. *arXiv preprint arXiv:2402.04401*, 2024.

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

Chris van der Lee, Albert Gatt, Emiel van Miltenburg, Sander Wubben, and Emiel Krahmer. Best practices for the human evaluation of automatically generated text. In *Proceedings of the 12th International Conference on Natural Language Generation*, pp. 355–368. Association for Computational Linguistics, 2019. doi: 10.18653/v1/W19-8643. URL https://aclanthology.org/W19-8643/.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63, 2017.

Hongru Wang, Rui Wang, Fei Mi, Yang Deng, Zezhong Wang, Bin Liang, Ruifeng Xu, and Kam Fai Wong. Cue-CoT: Chain-of-thought prompting for responding to in-depth dialogue questions with LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 12047–12064. Association for Computational Linguistics (ACL), 2023.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022. URL https://arxiv.org/abs/2212.03533v1.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *arXiv preprint arXiv:2002.10957*, 2020. URL https://arxiv.org/abs/2002.10957.

Stanisław Woźniak, Bartłomiej Koptyra, Arkadiusz Janz, Przemysław Kazienko, and Jan Kocoń. Personalized large language models. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 511–520. IEEE, 2024.

Bowen Wu, Wenqing Wang, Haoran Li, Yunhan Deng, Ying Li, Jingsong Yu, and Baoxun Wang. Interpersonal memory matters: A new task for proactive dialogue utilizing conversational history. In *Proceedings of the 29th Conference on Computational Natural Language Learning*, pp. 47–67, 2025. URL https://aclanthology.org/2025.conll-1.4/.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033, 2023.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general chinese embeddings. *arXiv preprint arXiv:2309.07597*, 2024. URL https://arxiv.org/abs/2309.07597v5. Version 5, September 2024.

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. *arXiv preprint arXiv:2503.02614*, 2025.

Haocheng Yu, Yaxiong Wu, Hao Wang, Wei Guo, Yong Liu, Yawen Li, Yuyang Ye, Junping Du, and Enhong Chen. Thought-augmented planning for LLM-powered interactive recommender agent. *arXiv preprint arXiv:2506.23485*, 2025.

Jizhi Zhang, Keqin Bao, Wenjie Wang, Yang Zhang, Wentao Shi, Wanhong Xu, Fuli Feng, and Tat-Seng Chua. Prospect personalized recommendation on large language model-based agent platform. *arXiv preprint arXiv:2402.18240*, 2024a.

Kai Zhang, Yejin Kim, and Xiaozhong Liu. Personalized LLM response generation with parameterized memory injection. *arXiv preprint arXiv:2404.03565*, 2024b.

Zhaowei Zhang, Fengshuo Bai, Qizhi Chen, Chengdong Ma, Mingzhi Wang, Haoran Sun, Zilong Zheng, and Yaodong Yang. Amulet: ReAlignment during test time for personalized preference adaptation of LLMs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. UserAdapter: Few-shot user learning in sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 1484–1488, 2021.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 10586–10613, 2024.

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

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Appendix Overview

The appendix is organized as follows:

- **Appendix A: Limitations and Scope** (Section A.1). This subsection discusses the limitations of automated preference judgments and the scope of the reported evidence.
- **Appendix B: RLHF Preliminaries and KL-regularized Derivation** (Section B). This section collects the RLHF setup used only to motivate our decoding-time objective in the main text and then gives the full closed-form derivation of the KL-regularized optimal policy. It ends by connecting prompt modification to an induced reward representation and proving the exponentiated reweighting form of the optimizer.
- **Appendix C: Related Work** (Section C). This section expands the background discussion from the main paper and situates EXACT relative to three lines of prior work: LLM alignment, training-time personalization, and decoding-time personalization. It ends by clarifying that EXACT is positioned as a decoding-time personalization method that prioritizes efficiency, interpretability, and deployment practicality.
- **Appendix D: Details for Theoretical Analysis** (Section D). This section provides the technical setup, assumptions, and proofs underlying the theory in the main text.
  - **D.1 Proof for Section 3.1** (Section D). This subsection formalizes the set objective and weak-submodularity machinery used in the greedy analysis. It ends by proving Theorem D.2, namely that greedy selection achieves a $(1 - e^{-\gamma})$ approximation under normalized monotone weak submodularity.
  - **D.2 Proofs for Section 3.2** (Section D.2). This subsection proves the pooling and retrieval results used to analyze topic drift. The retrieval result is retained as a draft requiring corrections to its variance decomposition, query-noise conditions, and error bound; it does not yet establish a guarantee for the implemented algorithm.
  - **D.3 Example of Incremental Vocabulary Expansion** (Section D.3). This subsection gives a concrete worked example of how new formatting-related attributes can be added after offline indexing. It ends by emphasizing that vocabulary expansion is modular and does not require retraining the base model.
- **Appendix E: Experiments** (Section E.1). This section collects the empirical details and supplementary analyses that support the main experimental claims.
  - **Appendix E.2: Experimental Setup**. This subsection specifies the compute environment, decoding hyperparameters, and baseline/EXACT implementations, including Base (Section E.2.1), RAG few-shot (Section E.2.2), Drift (Section E.2.3), PAD (Section E.2.4), Reward/RM (Section E.2.5), and EXACT (Section E.2.6). It ends by fixing the implementation choices used for all empirical comparisons.
  - **Appendix E.3: Supporting Materials for EXACT**. This subsection documents the attribute vocabulary (Section E.3.1) and representative user-query examples (Section E.3.2) used throughout the experiments. It ends by clarifying what information EXACT stores and injects when transferring preferences across prompts.
  - **Appendix E.4: Qualitative Analysis and Case Studies**. This subsection presents additional examples of contextual preference shifts, especially the same user preferring different response styles across prompts (Section E.4.1). Dataset assignments and source records for these cases remain to be reconciled.
  - **Appendix E.5: Additional Ablations**. This subsection groups the ablations into algorithmic stability and modularity, vocabulary robustness and extensibility, and retrieval and selection design. It reports the supplementary comparisons and identifies the retrieval ablation whose metric remains to be verified.
  - **Appendix E.6: Further Case Study**. This subsection adds user-level analyses beyond the main examples to show how retrieval repeatedly switches among distinct attribute subsets for the same user. These illustrate variation in retrieved model representations, not an independent measure of users' true preference shifts.
- **Appendix F: Blinded Human Evaluation Protocol**. This appendix retains the original study protocol and distinguishes received aggregate counts from a verified completed human study.

A.1 Limitations and Scope

LLM-as-a-judge is only a proxy for personalized human evaluation. Table 3 reports automated judgments from two LLMs. The aggregate EXACT–RAG counts noted in Appendix F have not yet been reconciled with the human-study records. This choice should not be interpreted as showing that the judged winner is always the response a user would truly prefer under real deployment conditions. In particular, an LLM judge may capture whether a response exhibits recognizable personalization signals, yet still fail to measure whether the degree of personalization is correct or whether the response is genuinely preferred by the target user.

At the same time, this limitation is difficult to avoid in the current benchmark setting. To the best of our knowledge, existing public personalized preference datasets provide offline user-level preference pairs, but do not give direct access to the same users' judgments when comparing two newly generated responses at test time. Obtaining such labels would require collecting fresh user-specific annotations for each compared method and prompt, or running a live user study in which the same users repeatedly evaluate personalized outputs online. Both are substantially more expensive than offline benchmarking and are not part of the current public evaluation protocol. For this reason, we follow recent decoding-time personalization work in using LLM-as-a-judge to maintain a controlled and comparable evaluation pipeline. Our generation-quality results should therefore be interpreted as supportive but not definitive evidence of personalized quality. A promising direction for future evaluation is to develop debiasing or calibration procedures that better align LLM judges with human personalized preferences, and to complement offline benchmarks with prospective human studies or online personalization experiments.

B RLHF Preliminaries and KL-regularized Derivation

B.1 RLHF Preliminaries

For a prompt x, let (y_w, y_l) denote a preferred/dispreferred response pair under a latent Bradley–Terry preference model, where y_w is preferred to y_l. We use the standard KL-regularized RLHF view only to motivate the decoding-time objective behind EXACT; the method itself does not require policy fine-tuning. Under this view, for a base policy π_{base}, one can consider the regularized objective

$$\max_{\pi_\theta} \ \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}\Big[r(x,y) - \beta\,\mathrm{KL}\big(\pi_\theta \,\|\, \pi_{\text{base}}\big)\Big]. \tag{12}$$

Here $\beta > 0$ controls the tradeoff between reward maximization and staying close to the base model. This formulation is useful because it provides a principled way to interpret prompt-conditioned policy changes as implicit reward shaping.

If a prompt modification m induces a new conditional policy $\pi^m(y \mid x) := \pi_{\text{base}}(y \mid x, m)$, then its implicit reward relative to the base model can be written as

$$r^m(x,y) = \beta \log \frac{\pi^m(y \mid x)}{\pi_{\text{base}}(y \mid x)} + \beta \log Z^m(x), \tag{13}$$

where $Z^m(x)$ is the induced partition function.

B.2 Derivation of the KL-regularized optimal policy

We provide a self-contained derivation of the closed-form optimizer of the KL-regularized objective in Eq. (12). For a fixed prompt x, consider optimizing over distributions $\pi(\cdot \mid x)$:

$$\max_{\pi(\cdot|x)} \ \sum_y \pi(y \mid x)\, r(x,y) - \beta \sum_y \pi(y \mid x) \log \frac{\pi(y \mid x)}{\pi_{\text{base}}(y \mid x)} \quad \text{s.t. } \sum_y \pi(y \mid x) = 1,\ \pi(y \mid x) \geq 0. \tag{14}$$

Proposition B.1 (Closed-form optimizer of KL-regularized RLHF (Rafailov et al., 2023))**.** *For any $\beta > 0$ and fixed x, the unique maximizer of Eq.* (14) *is*

$$\pi_r(y \mid x) = \frac{1}{Z(x)}\, \pi_{\text{base}}(y \mid x) \exp\!\left(\frac{1}{\beta}\, r(x,y)\right), \tag{15}$$

where the partition function is

$$Z(x) = \sum_y \pi_{\text{base}}(y \mid x) \exp\Big(\frac{1}{\beta} r(x, y)\Big). \tag{16}$$

Proof. We form the Lagrangian for Eq. (14) with multiplier λ enforcing $\sum_y \pi(y \mid x) = 1$:

$$\mathcal{L}(\pi, \lambda) = \sum_y \pi(y \mid x)\, r(x, y) - \beta \sum_y \pi(y \mid x) \log \frac{\pi(y \mid x)}{\pi_{\text{base}}(y \mid x)} + \lambda\Big(\sum_y \pi(y \mid x) - 1\Big). \tag{17}$$

For any y with $\pi(y \mid x) > 0$, the stationarity condition gives

$$0 = \frac{\partial \mathcal{L}}{\partial \pi(y \mid x)} = r(x, y) - \beta\Big(\log \frac{\pi(y \mid x)}{\pi_{\text{base}}(y \mid x)} + 1\Big) + \lambda. \tag{18}$$

Rearranging yields

$$\log \frac{\pi(y \mid x)}{\pi_{\text{base}}(y \mid x)} = \frac{1}{\beta}\big(r(x, y) + \lambda - \beta\big), \tag{19}$$

so

$$\pi(y \mid x) = \pi_{\text{base}}(y \mid x) \exp\Big(\frac{1}{\beta} r(x, y)\Big) \cdot \exp\Big(\frac{\lambda - \beta}{\beta}\Big). \tag{20}$$

The last factor is a constant w.r.t. y and is determined by normalization:

$$1 = \sum_y \pi(y \mid x) = \exp\Big(\frac{\lambda - \beta}{\beta}\Big) \sum_y \pi_{\text{base}}(y \mid x) \exp\Big(\frac{1}{\beta} r(x, y)\Big). \tag{21}$$

Thus $\exp\Big(\frac{\lambda-\beta}{\beta}\Big) = 1/Z(x)$ with $Z(x)$ defined in Eq. (16), which gives Eq. (15). Uniqueness follows since the objective in Eq. (14) is strictly concave in $\pi(\cdot \mid x)$ for $\beta > 0$ (negative KL is strictly concave). □

C Related Work

Large language model (LLM) alignment. LLM alignment aims to make the LLM behavior consistent with human preferences, often using reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Bai et al., 2022). Broadly, existing approaches fall into two categories. (1) RLHF with an explicit reward model: a reward model is trained from human feedback, and then policy optimization, typically Proximal Policy Optimization (PPO) (Schulman et al., 2017), is applied to obtain an aligned policy. In contrast, *decoding-time alignment* avoids expensive RL training by steering generation at inference time (Mudgal et al., 2024; Khanov et al., 2024; Han et al., 2024; Liu et al., 2024; Huang et al., 2025). (2) Direct preference optimization: methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) train the policy directly from preference data, eliminating the need to fit a separate reward model.

Training-time personalization. Training-time methods adapt language models using preference or history data, including parameter-efficient fine-tuning (PEFT) (Zhang et al., 2024b; Zhu et al., 2024; Woźniak et al., 2024; Kong et al., 2024) and user-specific adaptation (Zhong et al., 2021; Tan et al., 2024). PocketLLM studies on-device fine-tuning (Peng et al., 2024); these architectural choices alone do not imply a general privacy guarantee. LoRe learns a shared low-rank reward basis together with user-specific combination weights from multi-user feedback (Bose et al., 2025), rather than maintaining fully isolated reward models for each user. Multi-objective alignment also spans distinct optimization paradigms: Fine-Grained RLHF combines fine-grained reward signals for reinforcement learning (Wu et al., 2023), whereas MODPO extends direct preference optimization to multiple objectives without reinforcement learning (Zhou et al., 2024).

Decoding-time personalization. Inference-time methods avoid updating the generation model's parameters (Chen et al., 2025; Kim et al., 2025a). Prompt-based approaches include Cue-CoT's reasoning about dialogue cues (Wang et al., 2023) and task-aware summarization with retrieval (Richardson et al., 2023). CIPHER learns readable preference descriptions from user edits and aggregates

descriptions retrieved from similar contexts (Gao et al., 2024a). CUPID evaluates whether models infer and apply context-relevant preferences from interaction histories (Kim et al., 2025b). Thus neither interpretable preference memory nor contextual retrieval is unique to EXACT. Our method instead optimizes discrete attribute subsets using pairwise likelihood gaps and injects a retrieved subset at inference. Other approaches control token probabilities through decoding objectives or steering (Gao et al., 2024b; Li et al., 2023; He et al., 2025; Lv et al., 2025; Zhang et al., 2025). PAD uses reward-guided decoding (Chen et al., 2025); Drift learns weights over interpretable attributes and composes their decoding signals (Kim et al., 2025a). EXACT shares the use of explicit attributes with Drift, but uses prompt-specific subset retrieval and context injection in place of user-level weighted logit composition.

D Details for Theoretical Analysis

D.1 Proof for Section 3.1

D.1.1 Setup and Definitions

We consider the set maximization problem (6) with objective (5). For any $S \subseteq \mathcal{A}$ and $a \in \mathcal{A} \setminus S$, define the marginal gain

$$\Delta(a \mid S) \;:=\; F(S \cup \{a\}) - F(S), \qquad \Delta(L \mid S) \;:=\; F(S \cup L) - F(S) \tag{22}$$

for $L \subseteq \mathcal{A} \setminus S$.

Definition D.1 (Global submodularity ratio)**.** For monotone F, we use a global variant of Das & Kempe (2018, Definition 2), taking the infimum over all disjoint sets:

$$\gamma \;:=\; \inf_{S \subseteq \mathcal{A}} \; \inf_{L \subseteq \mathcal{A} \setminus S} \frac{\sum_{a \in L} \Delta(a \mid S)}{\Delta(L \mid S)}. \tag{23}$$

We take $0/0 = 1$ and a positive numerator over zero as $+\infty$, so $\gamma \in [0, 1]$. The approximation result assumes $\gamma > 0$.

D.1.2 Assumptions and Discussion

Normalized. It is straightforward to verify that $F(\varnothing) = 0$.

Monotonicity assumption. The guarantee assumes $F(S \cup \{a\}) \geq F(S)$ for every S and $a \notin S$. Attribute interactions may violate this assumption. A non-negativity filter can ensure improvement along the selected greedy sequence, but does not establish monotonicity over all subsets. The theorem is therefore conditional; such a filter alone does not establish its applicability to the empirical objective.

Weak submodularity. We assume F has a strictly positive submodularity ratio γ. Intuitively, γ is close to 1 when higher-order interactions among attributes are mild, and decreases as interactions strengthen. This notion of *weak submodularity* is commonly quantified by the *submodularity ratio* γ, originally introduced by Das & Kempe (2011; 2018) to measure how close a monotone set function is to being submodular. Under this definition, $\gamma = 1$ recovers exact submodularity, while $\gamma < 1$ captures departures due to interactions/overlap among elements. Moreover, the greedy approximation guarantee we use (scaling with $1 - e^{-\gamma}$) is not new and follows prior analyses of greedy maximization under submodularity ratio (e.g., Das & Kempe, 2018; Bian et al., 2017).

D.1.3 Proof for Theorem 3.1

Greedy algorithm. Initialize $S_0 = \varnothing$; for $t = 0, 1, \ldots, k-1$,

$$a_{t+1} \in \arg\max_{a \in \mathcal{A} \setminus S_t} \Delta(a \mid S_t), \qquad S_{t+1} = S_t \cup \{a_{t+1}\}. \tag{24}$$

Theorem D.2 (Greedy guarantee under weak submodularity (restated))**.** *Let S_k be the output of greedy selection, and let $S^* \in \arg\max_{|A| \leq k} F(A)$ be an optimal solution. If F is normalized and monotone and has a submodularity ratio γ, then*

$$F(S_k) \;\geq\; \left(1 - e^{-\gamma}\right) F(S^*). \tag{25}$$

Proof. The bound is established by a one-step progress inequality and then unrolling the recursion.

Step 1: a key inequality from the submodularity ratio. Fix any iteration $t \in \{0, 1, \ldots, k-1\}$ and consider the current set S_t. Let $L := S^* \setminus S_t$ denote the elements in the optimal set not yet selected. Note that $L \subseteq \mathcal{A} \setminus S_t$ and $|L| \le |S^*| \le k$.

By Definition D.1 (submodularity ratio), applied to (S, L) with $S = S_t$, we have

$$\sum_{a \in L} \Delta(a \mid S_t) \;\ge\; \gamma\, \Delta(L \mid S_t) \;=\; \gamma\big(F(S_t \cup L) - F(S_t)\big). \tag{26}$$

Since $S_t \cup L = S_t \cup (S^* \setminus S_t) = S_t \cup S^*$, monotonicity implies $F(S_t \cup S^*) \ge F(S^*)$. Therefore,

$$F(S_t \cup L) - F(S_t) \;=\; F(S_t \cup S^*) - F(S_t) \;\ge\; F(S^*) - F(S_t). \tag{27}$$

Combining (26) and (27) yields

$$\sum_{a \in L} \Delta(a \mid S_t) \;\ge\; \gamma\big(F(S^*) - F(S_t)\big). \tag{28}$$

Step 2: existence of a good element in the remaining optimal set. Since $|L| \le k$, the average marginal gain over elements in L satisfies

$$\max_{a \in L} \Delta(a \mid S_t) \;\ge\; \frac{1}{|L|} \sum_{a \in L} \Delta(a \mid S_t) \;\ge\; \frac{1}{k} \sum_{a \in L} \Delta(a \mid S_t). \tag{29}$$

Plugging (28) into (29) gives

$$\max_{a \in L} \Delta(a \mid S_t) \;\ge\; \frac{\gamma}{k}\big(F(S^*) - F(S_t)\big). \tag{30}$$

Step 3: greedy makes at least this progress. Greedy chooses a_{t+1} maximizing $\Delta(a \mid S_t)$ over all $a \in \mathcal{A} \setminus S_t$, hence in particular

$$\Delta(a_{t+1} \mid S_t) \;=\; \max_{a \in \mathcal{A} \setminus S_t} \Delta(a \mid S_t) \;\ge\; \max_{a \in L} \Delta(a \mid S_t). \tag{31}$$

Combining (31) and (30) yields the one-step progress inequality

$$F(S_{t+1}) - F(S_t) \;=\; \Delta(a_{t+1} \mid S_t) \;\ge\; \frac{\gamma}{k}\big(F(S^*) - F(S_t)\big). \tag{32}$$

Step 4: unrolling the recursion. Rearranging (32) gives

$$F(S^*) - F(S_{t+1}) \;\le\; \Big(1 - \frac{\gamma}{k}\Big)\big(F(S^*) - F(S_t)\big). \tag{33}$$

Applying (33) repeatedly from $t = 0$ to $t = k - 1$ yields

$$F(S^*) - F(S_k) \;\le\; \Big(1 - \frac{\gamma}{k}\Big)^k F(S^*), \tag{34}$$

where we used $F(S_0) = F(\varnothing) = 0$.

Finally, using the standard inequality $\big(1 - \frac{\gamma}{k}\big)^k \le e^{-\gamma}$, we obtain

$$F(S_k) \;\ge\; \Big(1 - e^{-\gamma}\Big) F(S^*), \tag{35}$$

which completes the proof. □

The submodularity ratio γ quantifies how far F deviates from additive behavior: when interactions are mild, γ approaches 1 and greedy recovers the classical $(1 - 1/e)$ guarantee; as interactions strengthen, γ decreases, and the bound degrades gracefully.

D.2 Proofs for Section 3.2

D.2.1 Proof of Pooling's Results

Recall $\widehat{\mu}_{\text{pool}} = \frac{1}{m_1}\sum_{i=1}^{m_1}\phi(x_i)$ and $\bar{\mu} = \sum_{c=1}^{k} p_c\mu_c$. Under the mean model in Theorem 3.2, $\mathbb{E}[\phi(x_i) \mid t(x_i) = c] = \mu_c$ and thus $\mathbb{E}[\phi(x_i)] = \sum_{c=1}^{k} p_c\mu_c = \bar{\mu}$, implying $\mathbb{E}[\widehat{\mu}_{\text{pool}}] = \bar{\mu}$. Hence, by bias–variance decomposition,

$$\mathbb{E}\|\widehat{\mu}_{\text{pool}} - \mu_j\|_2^2 = \|\bar{\mu} - \mu_j\|_2^2 + \mathbb{E}\|\widehat{\mu}_{\text{pool}} - \bar{\mu}\|_2^2.$$

Since the samples are independent,

$$\mathbb{E}\|\widehat{\mu}_{\text{pool}} - \bar{\mu}\|_2^2 = \frac{1}{m_1}\text{tr}(\text{Cov}(\phi(x))) = \frac{1}{m_1}\sum_{c=1}^{k} p_c\,\text{tr}(\Sigma_c),$$

which proves (8). The drift bias term $\|\bar{\mu} - \mu_j\|_2^2$ does not vanish as $m_1 \to \infty$.

D.2.2 Proof of Retrieval's Results (Eq. 10)

We prove two claims: (i) topic identification succeeds under (9); (ii) the MSE is variance-dominated.

Step 1: uniform concentration of topic centers. For each topic c, $\widehat{\mu}_c - \mu_c$ is the mean of n_c i.i.d. σ^2-subGaussian vectors. A standard vector concentration inequality implies that for $\eta > 0$,

$$\Pr(\|\widehat{\mu}_c - \mu_c\|_2 \geq \eta) \leq 2\exp\left(-\tilde{c}\,n_c\frac{\eta^2}{\sigma^2} + \tilde{c}'d\right),$$

for universal constants $\tilde{c}, \tilde{c}' > 0$. Choosing $\eta = r/4$ and applying a union bound over $c \in [k]$, condition (9) ensures the event

$$\mathcal{E} \triangleq \left\{\max_{c\in[k]}\|\widehat{\mu}_c - \mu_c\|_2 \leq r/4\right\}$$

holds with probability at least $1 - \delta/2$ (absorbing constants into C).

Step 2: correct nearest-center classification. Assume $\mathcal{E}$ holds and the test prompt satisfies $\phi(x) = \mu_j + \varepsilon$ with $t(x) = j$. For any $c \neq j$, by the topic-separation assumption in Theorem 3.2 and the triangle inequality,

$$\|\widehat{\mu}_c - \widehat{\mu}_j\|_2 \geq \|\mu_c - \mu_j\|_2 - \|\widehat{\mu}_c - \mu_c\|_2 - \|\widehat{\mu}_j - \mu_j\|_2 \geq r - \frac{r}{4} - \frac{r}{4} = \frac{r}{2}.$$

Also,

$$\|\phi(x) - \widehat{\mu}_j\|_2 \leq \|\varepsilon\|_2 + \frac{r}{4}.$$

If additionally $\|\varepsilon\|_2 \leq r/8$ (which holds with probability at least $1 - \delta/2$ by subGaussian tails after adjusting constants in (9)), then for any $c \neq j$,

$$\|\phi(x) - \widehat{\mu}_c\|_2 \geq \|\widehat{\mu}_c - \widehat{\mu}_j\|_2 - \|\phi(x) - \widehat{\mu}_j\|_2 \geq \frac{r}{2} - \left(\frac{r}{8} + \frac{r}{4}\right) = \frac{r}{8},$$

while $\|\phi(x) - \widehat{\mu}_j\|_2 \leq \frac{r}{8} + \frac{r}{4} = \frac{3r}{8}$. Thus $\widehat{j}(x) = j$ on this event. Combining with Step 1 yields $\Pr(\widehat{j}(x) = j) \geq 1 - \delta$.

Step 3: MSE decomposition. Decompose by correct vs. incorrect retrieval:

$$\mathbb{E}\|\widehat{\mu}_{\text{ret}}(x) - \mu_j\|_2^2 \leq \Pr(\widehat{j}(x) = j)\cdot\mathbb{E}\|\widehat{\mu}_j - \mu_j\|_2^2 + \Pr(\widehat{j}(x) \neq j)\cdot\sup_{c\neq j}\|\widehat{\mu}_c - \mu_j\|_2^2.$$

The first term is $\text{tr}(\Sigma_j)/n_j$. For the second term, using separation and $\mathcal{E}$, we can bound $\|\widehat{\mu}_c - \mu_j\|_2 \leq 2r$ (up to constants), hence the squared penalty is at most $4r^2$. Since $\Pr(\widehat{j}(x) \neq j) \leq \delta$, we obtain

$$\mathbb{E}\|\widehat{\mu}_{\text{ret}}(x) - \mu_j\|_2^2 \leq \frac{\text{tr}(\Sigma_j)}{n_j} + 4\delta r^2,$$

which proves the result in (10).

D.3 Example of Incremental Vocabulary Expansion

We provide a hypothetical example to illustrate how incremental vocabulary expansion can capture formatting preferences that are not covered by the initial attribute library. Consider a PRISM user whose historical preferences on advice-style prompts are already indexed with a retrieved subset

$$A = \{\textsc{Concise}, \textsc{Direct}, \textsc{Analytic}\}.$$

Suppose we later decide that the current vocabulary does not adequately describe response-format preferences and introduce a small external candidate set

$$\mathcal{A}_{\text{new}} = \{\textsc{Bullet-point}, \textsc{Step-by-step}, \textsc{Narrative}\}.$$

For a training prompt whose preferred response is organized as short bullet points while the dispreferred response is a single paragraph, we reapply Algorithm 1 only over $\mathcal{A}_{\text{new}}$, starting from the existing subset A. Concretely, for each $a \in \mathcal{A}_{\text{new}}$, we evaluate whether

$$\log \pi(y_w \mid x, A \cup \{a\}) - \log \pi(y_l \mid x, A \cup \{a\}) > \log \pi(y_w \mid x, A) - \log \pi(y_l \mid x, A),$$

which is exactly the improvement test induced by Eq. (4). If Bullet-point yields the largest positive gain, then the stored subset for that prompt is updated to

$$A' = \{\textsc{Concise}, \textsc{Direct}, \textsc{Analytic}, \textsc{Bullet-point}\},$$

whereas Step-by-step and Narrative are discarded if they do not improve the objective. At online inference time, when a new prompt is retrieved to this historical item, EXACT injects the expanded subset A' so that the model receives both the original style cues and the newly discovered formatting preference. This example shows that the expansion is modular: it augments the stored prompt-level attribute set without retraining the base model or redefining the original library.

E Experiment

E.1 Overview

This appendix section mirrors the empirical structure of the main paper. We first provide detailed experimental settings, including compute environment, decoding hyperparameters, and implementation details for all baselines and EXACT (Appendix E.2). We then summarize supporting materials for EXACT, including the attribute vocabulary and prompt-format examples used for attribute injection (Appendix E.3). Next, we group the additional ablations into the same three categories used in the main text (Appendix E.5): *algorithmic stability and modularity*, covering greedy selection, budget-sensitivity statistics, and compatibility with Context Steering; *vocabulary robustness and extensibility*, covering sensitivity to vocabulary perturbations and fixed-vocabulary extensions; and *retrieval and selection design*, covering the semantic-similarity assumption, multi-neighbor retrieval, and retrieval vs. non-retrieval comparisons. Finally, we provide qualitative analyses and case studies that complement the main-paper discussion of contextual preference shifts (Appendix E.4).

E.2 Experimental Setup

Compute environment. All experiments are run on a single NVIDIA H100 GPU (80GB). Unless otherwise stated, we use the same base instruction-tuned LLM backbone for all generation-based methods within a comparison group (e.g., Llama-3.1-8B-Instruct, Gemma-2-9B-it, or Qwen2.5-7B-Instruct), and keep decoding hyperparameters fixed across methods for fairness.

Decoding hyperparameters. For all generation-based methods, we use top-p sampling with `do_sample=True`, temperature 0.7, and $top_p = 0.95$. To ensure a fair comparison across methods, we fix the decoding configuration within each dataset and only vary the method-specific components (e.g., retrieved attributes or scoring signals). We cap the number of *newly generated tokens* to control output length: max_new_tokens$= 200$ for ***PRISM*** and max_new_tokens$= 300$ for ***Summarize from Human Feedback***. We do not use beam search (num_beams$= 1$) and stop generation at the model EOS token (or equivalent chat end token) when it appears.

Pairwise scoring protocol. All preference-modeling accuracy measurements use Eq. (11). For a likelihood-based method with conditional distribution π_m, the response score is

$$s_m(x, y; H_u) = \sum_{t=1}^{|y|} \log \pi_m(y_t \mid x, H_u, y_{<t}).$$

Only response tokens are scored; prompt, history, padding, and special template tokens are excluded. Both candidates use identical personalization context, fixed from the test prompt and training history before scoring; test responses and preference labels do not enter attribute selection or retrieval. EXACT, Base, and RAG use their respective conditioned language-model distributions. Decoding-based baselines use their method-specific conditional distributions, and a reward-model baseline uses its direct scalar reward. Exact ties count as incorrect; invalid or missing scores must be resolved before aggregation rather than silently excluded. The sequence score is a sum, not a length-normalized average or a change relative to the base policy.

For each seed, we first average correctness indicators within each user and then average equally over users. Across the three seeds, we report the mean and sample standard deviation of these user-macro accuracies, multiplied by 100. The fraction of examples whose likelihood gap improves over the base model is a separate diagnostic and is not reported as accuracy. LLM-judge generation win-rate remains a separate metric (Section 4.3).

E.2.1 BASE: HISTORY-INJECTION PROMPTING

Setup. **Base** is a prompt-engineering baseline that conditions the model on the user's historical preference pairs by serializing training comparisons into a system prompt (history-injection). Concretely, we include up to `max_history_pairs`=200 training pairs and truncate the resulting history to `max_history_chars`=20000 characters to control context length. We then query the same backbone LLM used by other methods and keep decoding hyperparameters fixed. For chat models whose templates do not support the `system` role (e.g., Gemma), we apply a robust fallback by merging system text into the user message to ensure consistent formatting.

E.2.2 RAG FEW-SHOT: RETRIEVAL-AUGMENTED PROMPTING

Setup. **RAG few-shot** is a retrieval-augmented prompting baseline that directly injects *raw* user history, without attribute abstraction. Given a test prompt, we embed it and retrieve a small set of the most similar training prompts from the user's history using cosine similarity (using the same embedding model as EXACT for a fair comparison). We then serialize the corresponding historical preference pairs (prompt, preferred response, dispreferred response) as an in-context few-shot history block and prepend it to the test prompt before generation. To keep context length controlled, we apply the same `max_history_pairs` and `max_history_chars` truncation rules as Base, and use the same backbone and decoding hyperparameters as other methods.

E.2.3 DRIFT: DIFFERENTIAL SCORING WITH A SMALL MODEL

Setup. **Drift** uses a dual-model configuration: a large LLM for generation (`llm_model`) and a smaller instruction-tuned model for differential scoring (`slm_model`). The key hyperparameter is β (default `beta`=1.0), which controls the strength of the preference signal from the small model. When the token vocabularies of the two models differ, we enable an approximate token-string mapping to align scores across vocabularies. Drift does *not* rely on an explicit reward model or additional alignment fine-tuning beyond the pretrained instruction-tuned checkpoints.

E.2.4 PAD: REWARD-GUIDED DECODING

Setup. **PAD** (Chen et al., 2025) personalizes decoding by steering generation with a reward signal at inference time. We adapt its reward-guided decoding mechanism to our setting: given a prompt, PAD proposes candidates under the same backbone and decoding hyperparameters, and uses the PAD guidance rule to select/steer toward higher-reward continuations. In our experiments, the reward signal is derived from each user's preference data, and we keep the evaluation protocol and data splits identical to other methods for a fair comparison.

E.2.5 REWARD (RM): PERSONALIZED REWARD MODELING (LORA)

Setup. **RM** trains a per-user reward model using pairwise preference data. We instantiate a sequence-classification reward model (`rm_model`) and fine-tune it with LoRA (`lora_r`=16, `lora_alpha`=32, `lora_dropout`=0.05) for `epochs`=3 using AdamW with learning rate `lr`=1e-4, batch size `batch_size`=8, and max sequence length `max_length`=1024.

E.2.6 EXACT

Setup. We tested BGE-small and BGE-base from the BGE family (Xiao et al., 2024), a MiniLM-based encoder (Wang et al., 2020), and E5-small and E5-base (Wang et al., 2022). BGE-small achieved the best performance in this comparison and was used in the main experiments.

E.2.7 HOW EXACT DIFFERS FROM A RAW-HISTORY RAG BASELINE

We would like to clarify how EXACT differs from a raw-history RAG baseline. First, in terms of interpretability, EXACT produces an explicit preference representation through selected attributes, making the inferred personalization signal easier to inspect and understand, whereas RAG retrieves historical examples directly without summarizing them into a structured preference representation. Second, at online inference time, EXACT is lightweight: it only needs the embedding of the current prompt and the associated selected attribute subset from the offline stage. A RAG baseline, by contrast, must retrieve and inject raw historical examples, which requires storing the original prompt/history and typically leads to substantially longer prompts. In our setting, EXACT injects only a small set of preference keywords, whereas RAG few-shot prompting adds several full historical examples, increasing prompt length and inference cost. Third, at the offline stage, EXACT does incur additional computation because it must identify a selected preference subset for each prompt, but this cost is paid once offline. RAG avoids this offline selection step, while shifting the burden to longer and less structured prompt conditioning at test time. In short, we view RAG as an important and meaningful baseline, but it reflects a different tradeoff: direct retrieval of raw user history versus compact, interpretable, and prompt-specific preference abstraction.

E.3 SUPPORTING MATERIALS FOR EXACT

E.3.1 USED ATTRIBUTES FOR CHARACTERIZING USERS' PREFERENCES

We use an interpretable attribute vocabulary to characterize users' preferences, following the attribute design in Drift (Kim et al., 2025a). The attributes cover multiple dimensions of behavior, including *style* (e.g., *Formal*, *Concise*, *Vivid*), *stance and tone* (e.g., *Modest*, *Direct*, *Critical*, *Empathetic*), *reasoning and expertise* (e.g., *Analytic*, *Engineer*, *Code*), and *values and worldview* (e.g., *Principled*, *Utilitarian*, *Hedonist*, *Collectivist*). These attributes are not mutually exclusive and may overlap in practice. For instance, *Formal* often co-occurs with *Respect*, while *Direct* and *Concise* both encourage brevity. Similarly, *Analytic* may align with *Engineer* or *Code* depending on the task context. Table 4 lists the full attribute set used in our experiments. To stay consistent with Drift's experimental setup and focus on controlled comparisons, the main experiments use only this predefined vocabulary; nevertheless, EXACT can be extended to online settings where new attributes arrive after the offline indexing stage via the incremental vocabulary expansion described in Section 2.1. To assess how restrictive this fixed-vocabulary design is in practice, we report two complementary ablations in Appendix E.5.2 and Appendix E.5.2. Reducing the predefined vocabulary to a random 80% subset causes only modest degradation across datasets and backbones, while incremental expansion with newly introduced formatting attributes consistently improves over the baseline EXACT on PRISM; by contrast, continuous weighting over the original attribute set yields little additional gain.

E.3.2 EXAMPLES FOR USER QUERIES

Figures 2–7 retain draft prompt-format illustrations. Their dataset assignments, response-preference labels, and links to experimental records remain unverified. Five of the six illustrations include attributes absent from Table 4, such as *Faithful*, *Structured*, and *Enthusiastic*; they must not be treated as verified outputs of the stated $K = 42$ configuration. We preserve their content for source reconciliation rather than substituting unmeasured attribute selections.

Attribute	Attribute
Base	Creative
Formal	Analytic
Concise	Empathetic
Vivid	Sycophant
Modest	Old-fashioned
Engineer	Meritocratic
Persuasive	Myopic
Emotion	Principled
Humor	Hedonist
Energy	Utilitarian
Code	Realist
Easy	Pessimistic
Direct	Storyteller
Social	Flexible
Western	Spontaneous
Eastern	Collectivist
Respect	Individualistic
Internet Slang	Exclamatory
Proverb	Conspiracy
Critical	Tech Industry Priority
Vague	Eco-friendly

Table 4: Attribute vocabulary used in our experiments to characterize users' preferences.

user_query:
`do you think there are ufo sightings hidden from us?`

selected_attributes $\mathcal{A}$:
`Direct, Concise, Analytic`

original_prompt:
`User: do you think there are ufo sightings hidden from us?`

modified_prompt:
`User: do you think there are ufo sightings hidden from us?`
`Attributes: <Direct, Concise, Analytic>`

Win Response: It is possible some sightings are not fully disclosed due to classified programs, but evidence is inconclusive. Plausible reasons include: (1) national security, (2) ongoing investigations, (3) reporting bias.

Lose Response: There are many reports, some explained and others unexplained; however, there is no conclusive evidence that sightings are intentionally hidden.

Figure 2: **Draft illustration 1 (PRISM attribution unverified).** Prompt modification and a response pair; see the provenance note in Appendix E.3.2.

user_query:
Hello, what do you think about fantasy books?

selected_attributes $\mathcal{A}$:
Enthusiastic, Elaborate, Imaginative

original_prompt:
User: Hello, what do you think about fantasy books?

modified_prompt:
User: Hello, what do you think about fantasy books?
Attributes: <Enthusiastic, Elaborate, Imaginative>

Win Response: Fantasy books are a delightful doorway into other worlds—full of magic systems, strange creatures, and epic quests. They let us explore courage, friendship, and wonder through stories that feel larger than life, often with rich lore and vivid settings that invite the imagination to run wild.

Lose Response: Fantasy books are enjoyable and creative. They often feature imaginative worlds and compelling adventures. If you tell me what you like (cozy vs. epic, romance vs. gritty), I can recommend a few series.

Figure 3: **Draft illustration 2 (PRISM attribution unverified).** Attribute names and response labels require verification.

user_query:
Which are the most important laws to follow?

selected_attributes $\mathcal{A}$:
Comprehensive, Specific, Practical

original_prompt:
User: Which are the most important laws to follow?

modified_prompt:
User: Which are the most important laws to follow?
Attributes: <Comprehensive, Specific, Practical>

Win Response: Practically, prioritize laws that protect safety and rights: (1) laws against violence and threats, (2) theft and fraud statutes, (3) traffic and impaired-driving rules, (4) workplace and housing anti-discrimination laws, and (5) basic civic obligations like taxes. If you share your country/state, I can list the most relevant examples and common pitfalls.

Lose Response: All laws matter because they maintain order. We should follow laws in general and respect authority, though laws can be debated and improved over time.

Figure 4: **Draft illustration 3 (PRISM attribution unverified).** Attribute names and response labels require verification.

user_query:
Summarize the following post for a busy reader.

selected_attributes $\mathcal{A}$:
Concise, Faithful, Neutral

original_prompt:
User: Summarize the following post for a busy reader.
Post: "Our team shipped a new feature after two months of iteration. Early users liked the speed, but some found the UI confusing. We ran A/B tests, simplified the layout, and saw support tickets drop by 30%. Next we will focus on accessibility and documentation."

modified_prompt:
User: Summarize the following post for a busy reader.
Post: "Our team shipped a new feature after two months of iteration. Early users liked the speed, but some found the UI confusing. We ran A/B tests, simplified the layout, and saw support tickets drop by 30%. Next we will focus on accessibility and documentation."
Attributes: <Concise, Faithful, Neutral>

Win Response: The team launched a new feature after two months. Users praised speed but found the UI confusing; after A/B testing and simplifying the layout, support tickets fell by 30%. Next steps include improving accessibility and documentation.

Lose Response: The team worked hard to build an amazing feature. Users loved it, and the company is excited to keep improving things in the future.

Figure 5: **Draft illustration 1 (Summarize attribution unverified).** Attribute-guided summarization prompt with a representative preferred vs. dispreferred summary.

user_query:
Write a short summary highlighting the key takeaway and evidence.

selected_attributes $\mathcal{A}$:
Analytic, Structured, Specific

original_prompt:
User: Write a short summary highlighting the key takeaway and evidence.
Post: "A city piloted free bus fares for six months. Ridership increased 18%, especially on weekends. Surveyed riders cited cost savings and convenience. However, overcrowding rose on two routes and the city added buses to address it."

modified_prompt:
User: Write a short summary highlighting the key takeaway and evidence.
Post: "A city piloted free bus fares for six months. Ridership increased 18%, especially on weekends. Surveyed riders cited cost savings and convenience. However, overcrowding rose on two routes and the city added buses to address it."
Attributes: <Analytic, Structured, Specific>

Win Response: **Takeaway:** Free fares increased public transit use.
Evidence: Ridership rose 18% (largest on weekends) and surveys reported cost savings and convenience.
Caveat: Overcrowding worsened on two routes, requiring additional buses.

Lose Response: The city made buses free and people used them more. There were also some issues, but overall it seemed positive.

Figure 6: **Draft illustration 2 (Summarize attribution unverified).** Attribute-guided summarization emphasizing structure and specificity.

user_query:
Summarize in a warm, supportive tone while staying factual.

selected_attributes $\mathcal{A}$:
Empathetic, Faithful, Clear

original_prompt:
User: Summarize in a warm, supportive tone while staying factual.
Post: "I started learning to cook this year. At first, I burned everything and felt embarrassed. I kept trying simple recipes, asked friends for tips, and now I can make a few meals I'm proud of. I still make mistakes, but it's getting easier."

modified_prompt:
User: Summarize in a warm, supportive tone while staying factual.
Post: "I started learning to cook this year. At first I burned everything and felt embarrassed. I kept trying simple recipes, asked friends for tips, and now I can make a few meals I'm proud of. I still make mistakes, but it's getting easier."
Attributes: <Empathetic, Faithful, Clear>

Win Response: They began learning to cook, struggled at first, and felt embarrassed, but kept practicing simple recipes and sought friends' advice. Now they can make several meals they're proud of, and although mistakes still happen, cooking is becoming easier.

Lose Response: They have completely mastered cooking and never make mistakes anymore, which proves that anyone can become a perfect chef quickly.

Figure 7: **Draft illustration 3 (Summarize attribution unverified).** Attribute-guided summarization balancing warmth with factual faithfulness.

Dataset	User	Prompt (Title)	Response A	Response B
PRISM	RgH765...	**P1:** *I [20M] and girl... (FWB)*	× *"Ex and I broke up. I don't know what to do."* (short)	✓ *"...fell in love... boyfriend in Japan... manly tears..."* (longer / detailed)
PRISM	RgH765...	**P2:** *My [30F] dad [51M] is coming over...*	✓ *"My dad is coming tonight. I'm nervous..."* (short / concise)	× *"6'8" gay, divorced dad... EDIT: ... I'm going to go drink first..."* (longer)

Table 5: **Draft case: length/detail contrast (dataset attribution pending).** User RgH765... prefers a longer, more detailed summary in P1 but a shorter, more concise summary in P2. Colored words mark the salient style cue.

E.4 Qualitative Analysis and Case Studies

E.4.1 Examples of Contextual Preference Shifts

Tables 5 and 6 retain additional draft cases. As in Table 1, their dataset and user mappings require verification: the PRISM-labeled entries are summaries, while the Summarize-labeled entries are open-ended requests. This mismatch with the dataset descriptions must be resolved before drawing empirical conclusions from these examples.

E.5 Additional Ablations

This subsection expands the ablation summary in Section 4.4 of the main paper. To keep the appendix aligned with the main-text narrative, we organize the results into three groups: *algorithmic stability*

Dataset	User	Prompt (Title)	Response A	Response B
Summarize from Human Feedback	`user9`	**4A:** *Write a poem about my white cat.*	✓ *"What kind of poem are you thinking of? ... haiku, sonnet, or free-form?"* (interactive questions)	× *"Here is a 47-word poem about a white cat: ..."* (direct answer)
Summarize from Human Feedback	`user9`	**4B:** *People should not eat meat... strongly encouraged.*	✓ *"Encouraging a meat-free lifestyle can improve health and reduce environmental impact..."* (informational)	× *"Would you like me to go into more detail?"* (follow-up question)

Table 6: **Draft case: interactive vs. informational contrast (dataset attribution pending).** User `user9` prefers interactive clarification for a creative task (4A) but direct informational delivery for an opinion/value task (4B).

and modularity, *vocabulary robustness and extensibility*, and *retrieval and selection design*. Each group below lists the corresponding experiments and their main conclusions.

E.5.1 Algorithmic Stability and Modularity

Algorithmic-group overview. This group evaluates the stability of the optimization procedure and the modularity of EXACT as a decoding-time framework. The corresponding experiments examine greedy vs. exhaustive selection, additional budget-sensitivity statistics, and compatibility with Context Steering.

Greedy vs. exhaustive subset selection. While Theorem 3.1 provides a constant-factor approximation under weak submodularity, we additionally validate the greedy approximation directly on the Summarize from Human Feedback dataset when exhaustive search is feasible (k=3). Table 7 shows that greedy achieves near-optimal objective values across base models, with a high ratio $F(S_k)/F(S^*)$ and a small absolute gap, supporting greedy as an effective and scalable attribute selection procedure in practice.

Metric (k=3)	Llama-3.1-8B	Gemma-2-9B-it	Qwen2.5-7B
$F(S_k)$ (Greedy)	30.34	31.23	30.18
$F(S^*)$ (Overall/Exhaustive)	33.38	32.23	33.20
$r = F(S_k)/F(S^*)$	0.91	0.97	0.91
$\Delta = F(S^*) - F(S_k)$	3.04	1.00	3.02

Table 7: **Greedy vs. exhaustive objective.** We report the objective values achieved by greedy selection ($F(S_k)$) and exhaustive search ($F(S^*)$). The ratio r is computed from the displayed objective values and rounded to two decimals; Δ is the absolute optimality gap.

We further extend this comparison beyond k=3 on PRISM using Qwen2.5-7B-Instruct for a representative user (`thott7XepukYSbOL2QgSlyXd0rgHvr`), and compare greedy selection against exhaustive optimal subsets at $k \in \{3, 5, 10\}$. Table 8 reports the average objective gap $\Delta = F(S^*) - F(S_k)$ (Eq. (4)) and the overlap ratio between the greedy and optimal attribute sets. These draft values require search-log verification: $\binom{42}{10} = 1{,}471{,}442{,}973$ size-10 subsets would be considered in a full search. The candidate restriction, search procedure, and user/dataset mapping must be reconciled before this table can support an optimization-error claim.

To complement Figure 1, Table 9 reports the standard deviations (across three random seeds) of the run-level mean accuracies for Qwen on PRISM at each k.

Budget k	3	5	10
Avg. objective gap $\mathbb{E}[\Delta]$	0.05	0.08	0.11
Overlap ratio $\mathbb{E}[\lvert S_k \cap S^* \rvert / k]$	0.96	0.94	0.88

Table 8: **Draft greedy–exhaustive comparison (search scope and provenance pending).** For a representative user (`thott7XepukYSbOL2QgSlyXd0rgHvr`), we compare greedy-selected subsets S_k with exhaustive optima S^* at larger budgets. We report the average objective gap $\Delta = F(S^*) - F(S_k)$ (Eq. (4)) and the average overlap ratio between the selected sets.

Budget k	1	3	5	10
Std. of accuracy (%)	0.12	0.13	0.13	0.18

Table 9: **Standard deviations for Qwen on PRISM.** The draft reports seed standard deviations for each k in Figure 1. Verification is pending: at $k = 3$, 0.13 conflicts with 0.64 in Table 2. Both values are retained until the run-level results identify the correct scope and statistic.

Compatibility with Context Steering. We additionally test whether retrieval-based personalization can be combined with an external decoding-time steering module. Following Context Steering (CoS) (He et al., 2025), we evaluate two hybrid variants on PRISM: *EXACT+CoS* and *RAG few-shot+CoS*. For EXACT+CoS, the steering context is the retrieved attribute subset selected by EXACT; for RAG few-shot+CoS, the steering context is the retrieved in-context history block used by the RAG baseline. We tune the CoS interpolation weight over $\lambda \in \{0.3, 0.5, 0.8\}$ and report the best result for each backbone under the same evaluation protocol as Table 2.

Method	**Llama-3.1-8B**	**Gemma-2-9B-it**	**Qwen2.5-7B-Instruct**
EXACT + CoS	**67.22**	**65.73**	**66.21**
RAG few-shot + CoS	64.21	63.23	63.31

Table 10: **Compatibility with Context Steering on PRISM.** We combine CoS (He et al., 2025) with EXACT and RAG few-shot, tuning the steering weight over $\lambda \in \{0.3, 0.5, 0.8\}$. EXACT+CoS remains consistently stronger than RAG few-shot+CoS across all three backbones.

Table 10 shows that EXACT remains effective when paired with an external steering module, indicating that its retrieval-based personalization signal is compatible with modular decoding-time control. EXACT+CoS exceeds RAG few-shot+CoS in these three comparisons. Relative to standalone EXACT in Table 2, however, its changes are $+0.60$, -0.20, and -0.20 percentage points for Llama, Gemma, and Qwen. These results demonstrate compatibility, not a uniform benefit from adding CoS.

E.5.2 Vocabulary Robustness and Extensibility

Vocabulary-group overview. This group evaluates whether EXACT depends critically on a fixed predefined attribute library, and whether the representation can be extended when that library is incomplete. The corresponding experiments are sensitivity to moderate vocabulary perturbations and modular fixed-vocabulary extensions.

Sensitivity to the predefined vocabulary. We also perform a direct sensitivity study on the predefined attribute library itself. To test whether EXACT depends critically on the full vocabulary in Table 4, we construct a reduced library by randomly retaining a proportion $\tau = 80\%$ of the original attributes and rerun EXACT under the same evaluation protocol. This ablation keeps the model, retrieval procedure, and optimization pipeline unchanged, and only modifies the candidate pool available to Algorithm 1. It therefore isolates how sensitive the method is to moderate perturbations of the predefined vocabulary.

Table 11 shows that reducing the vocabulary by 20% leads to only small decreases in accuracy across both datasets and all three backbones. The decreases range from 0.07 to 0.89 percentage points. This

Dataset	Method	Llama-3.1-8B	Gemma-2-9B-it	Qwen2.5-7B-Instruct
PRISM	EXACT ($\tau = 80\%$)	66.12	65.22	66.34
	EXACT	**66.62**	**65.93**	**66.41**
Summarize from Human Feedback	EXACT ($\tau = 80\%$)	64.23	65.89	59.84
	EXACT	**65.12**	**66.01**	**60.28**

Table 11: **Sensitivity to the predefined vocabulary.** We compare EXACT under the full attribute library with a reduced library formed by randomly retaining $\tau = 80\%$ of the predefined attributes. Performance drops only modestly, suggesting that EXACT is reasonably robust to moderate perturbations of the candidate vocabulary.

describes the reported 80% subset experiment; it does not establish robustness to arbitrary vocabulary choices or identify which individual attributes matter.

Fixed-vocabulary extensions. We further study whether EXACT is overly constrained by the fixed attribute vocabulary used in the main experiments. To this end, we evaluate two add-on variants within the same pipeline on PRISM. The first is *incremental vocabulary expansion*: after the offline stage has selected a prompt-specific subset from the original library, we introduce a small external set of formatting-oriented attributes, including BULLET-POINT, STEP-BY-STEP, and NARRATIVE, and reapply Algorithm 1 only over these new candidates. Any newly introduced attribute is accepted only if it improves the objective in Eq. (4), so the expanded subset augments the original prompt-level memory without retraining the base model.

The second variant is *continuous weighting*. Instead of treating selected attributes as an unordered set, we assign each attribute a scalar weight based on its normalized marginal gain when it is added during greedy selection. At inference time, these weights are verbalized in the prompt to reflect relative importance, thereby extending the discrete subset representation to a soft preference profile while keeping the retrieval pipeline unchanged. As a lightweight control, we also report an *ordered attributes* variant that preserves only the greedy ranking of selected attributes but does not expose explicit numeric weights.

Method	Llama-3.1-8B	Gemma-2-9B-it	Qwen2.5-7B-Instruct
EXACT (Baseline)	66.62	65.93	66.41
EXACT (Incremental Expansion)	**67.38**	**66.45**	**67.83**
EXACT (Continuous Weights)	66.52	66.21	65.82
EXACT (Ordered Attributes)	66.62	66.12	66.10

Table 12: **Ablation on fixed-vocabulary extensions on PRISM.** We compare the baseline EXACT with three extensions: incremental vocabulary expansion, continuous attribute weighting, and ordered attributes. Incremental expansion consistently improves accuracy, while continuous or ordered variants over the original vocabulary provide limited gains.

Table 12 reports gains of 0.76, 0.52, and 1.42 percentage points from expansion. Accepted additions can increase subset size beyond the original $k = 3$, so these results do not isolate vocabulary coverage from attribute budget. Continuous weighting and ordering show mixed, smaller changes; the table alone does not identify the primary bottleneck.

E.5.3 RETRIEVAL AND SELECTION DESIGN

Retrieval-group overview. This group studies which retrieval-related design choices are most important for context-specific personalization. The corresponding experiments evaluate the semantic-similarity assumption, multi-neighbor retrieval, and retrieval vs. non-retrieval behavior.

Semantic similarity and selected-attribute overlap. EXACT retrieves prompt-specific preference attributes by nearest-neighbor search in the prompt embedding space. This design implicitly assumes that *semantic* similarity between prompts is informative of *preference* similarity. To examine a proxy for this assumption, we perform an offline correlation analysis on both PRISM and Summarize from

Human Feedback. For each user, we form all pairs of deduplicated historical prompts, compute semantic similarity as the cosine similarity between retrieval embeddings, and use selected-attribute overlap as a proxy for preference similarity, computed as the overlap ratio between the corresponding selected attribute subsets (i.e., $|A_i \cap A_j|/k$ for $|A_i| = |A_j| = k$). We then compute the correlation coefficient between semantic similarity and preference similarity and report dataset-level averages in Table 13. These correlations concern attributes selected by EXACT within historical data. They do not independently validate human preference similarity or establish coverage of held-out test contexts.

Dataset	Correlation (semantic vs. attribute overlap)
PRISM	0.873
Summarize from Human Feedback	0.921

Table 13: **Semantic similarity and selected-attribute overlap in history.** We report the average correlation between prompt semantic similarity (embedding cosine) and selected-attribute overlap measured over pairs of deduplicated historical prompts.

Top-1 vs. multi-neighbor retrieval. Retrieving multiple nearest neighbors is a natural extension of prompt-similarity retrieval. To test whether smoothing across neighbors improves robustness, we evaluate a simple top-3 variant of Algorithm 2. Given a test prompt, we retrieve the top-3 most similar historical prompts and aggregate their attribute subsets via a per-attribute majority vote to obtain a single injected subset of size k. Table 14 compares top-1 versus top-3 retrieval under the same evaluation protocol as Table 2. Across both datasets and all backbones, the reported top-3 variant underperforms top-1. This establishes the ordering of these implementations, but does not isolate why aggregation loses accuracy.

Dataset	Backbone	Top-1	Top-3 (maj. vote)
PRISM	Llama-3.1-8B	66.62	64.12
	Gemma-2-9B-it	65.93	64.45
	Qwen2.5-7B-Instruct	66.41	64.21
Summarize from Human Feedback	Llama-3.1-8B	65.12	63.12
	Gemma-2-9B-it	66.01	65.83
	Qwen2.5-7B-Instruct	60.28	59.34

Table 14: **Top-1 vs. top-3 retrieval.** We compare top-1 retrieval with a simple top-3 nearest-neighbor variant using majority-vote aggregation of retrieved attribute subsets.

Retrieval vs. non-retrieval. Our similarity-based attribute injection framework can be instantiated in two ways: a *retrieval* variant that selects attributes conditioned on the test prompt, and a *non-retrieval* variant that applies a single global attribute set to all test prompts. Concretely, *Retrieval* computes the embedding of a test prompt and retrieves the most similar training prompt in the user's history; it then injects the attribute subset associated with that retrieved training instance. In contrast, *Non-retrieval* ignores the test prompt and always injects the single attribute set that achieves the highest objective score on the training data.

Both variants must be evaluated using the strict pairwise ranking accuracy in Eq. (11), with response log-likelihoods under the corresponding attribute-conditioned prompt. The comparison asks whether context-dependent attribute retrieval improves correct preference rankings relative to a fixed user-level attribute set.

Draft result verification. The previous version described this ablation using an increase in the baseline-anchored likelihood gap, which is not the accuracy defined above. Table 15 retains the previously reported numbers for reconciliation; they have not yet been verified under the unified criterion and are not used here as evidence of an accuracy improvement. Per-example scores are required to confirm or recompute these entries and resolve the Qwen value of 60.75 versus 60.28 in Table 2.

Method	Llama-3.1-8B	Gemma-2-9B-it	Qwen2.5-7B
EXACT (with Retrieval)	65.12	66.01	60.75
EXACT (without Retrieval)	57.43	57.21	54.23

Table 15: **Previously reported retrieval-ablation values (metric verification pending).** Values are retained from the earlier draft for Summarize from Human Feedback; they must be confirmed or recomputed under Eq. (11) before being reported as pairwise accuracy.

E.6 Further Case Study

We analyze the user (ID: `d8YBBtVzdVnMLZuzqg88ES4klUw4u7`) from the Summarize From Human Feedback dataset to contrast preference structure under the same pipeline. After filtering low-gain examples and deduplicating by prompt identity, the training index contains 10 unique prompts, each with a greedy-selected attribute subset (k=3). In this case, the most frequent attributes include Internet Slang (4/10) together with expressive styles such as Vivid and Exclamatory (3/10 each), suggesting a more colloquial and high-energy global profile. At inference, hard top-1 retrieval again concentrates on only two training items (each used for 50% of the test prompts), but the two retrieved selected subsets are *qualitatively different*—{Creative, Vivid, Exclamatory} versus {Humor, Easy, Old-fashioned}—showing that the retrieved representation can vary across prompts; this is not an independent measurement of the user's true preference modes.

We also study a third user (ID: `bt1XW9PqRFCxDMGJ3V8V32AyyojIrE`) from the Summarize From Human Feedback dataset. After filtering and deduplication, the training index has 10 unique prompts with greedy-selected subsets (k=3). The training-time selections show a global tendency toward Concise (5/10), often paired with Exclamatory (4/10) and Humor (3/10). At inference, top-1 retrieval is strongly concentrated: 11 test prompts select only two training items as nearest neighbors (72.7% vs. 27.3%). Importantly, these two retrieved selected subsets are *not the same*—{Internet Slang, Concise, Humor} versus {Formal, Concise, Exclamatory}—again showing retrieval of distinct model-selected subsets for the same user, without independently establishing true preference shifts.

F BLINDED HUMAN EVALUATION PROTOCOL

Study status and objective. This appendix retains the original human-evaluation protocol; sample sizes below are design targets, not verified completion counts. The study compares EXACT directly with RAG few-shot on history-conditioned preference fit, with task quality evaluated separately. Criteria and instructions are explicit to support reproducibility (van der Lee et al., 2019; Schoch et al., 2020). The intended measure is third-party assessment from available history, rather than original-user satisfaction or attribute faithfulness.

Draft reconciliation status. The authors supplied three aggregate groups of 400 EXACT–RAG judgments: respectively 170/54/176, 229/34/137, and 194/53/153 wins/ties/losses for EXACT. Pooled counts are 593/141/466 out of 1,200 judgments (49.42%/11.75%/38.83%). These received totals differ from the original 600-judgment target. Rater identities, shared-item correspondence, dataset allocation, displayed histories, and blinding remain to be documented; item-level ratings and quality scores are unavailable in this package. The counts therefore are descriptive author-reported aggregates, not the user-macro analysis below or verified evidence of a completed blinded human study. No agreement, significance, or quality conclusion is inferred from them.

F.1 SAMPLING AND PAIRED GENERATION

We target 100 distinct user–prompt instances from each dataset, using the Llama-3.1-8B backbone from the main experiments and the same checkpoint for both methods. Eligible instances come from the held-out partitions of the three experimental runs. Instances will be deduplicated by dataset, user, and prompt identity. If an instance appears in multiple runs, one run will be selected at random, retaining that run's training history and both candidate generations. Within each dataset, we will randomly order users and sample one available instance per user in successive rounds until reaching the target. If fewer than 100 unique instances are available, all eligible instances will be used. This procedure prioritizes user coverage; actual user counts, instance counts, and run composition will be reported. Sampling and presentation use a fixed seed of 20260925, and the sample list will be frozen before formal annotation, without selection based on method performance or automated judgments.

Both methods will receive the same test request and access the same eligible training history, with a common backbone, retrieval encoder, and prespecified input-token ceiling. Each will retain its own history representation and frozen retrieval configuration. One response per method and instance will be evaluated; compatible existing generations may be reused without selecting among alternative outputs. New generations will use temperature 0.7 and top-p 0.95, with maximum output lengths of 200 tokens for PRISM and 300 for Summarize from Human Feedback. Model identifiers, generation seeds, prompts, retrieval settings, truncation rules, and actual input/output token counts will be retained. This comparison evaluates the complete methods; attributing differences specifically to explicit attributes requires separate controls.

F.2 BLINDING AND ANNOTATION

The original plan assigns all sampled instances to each of three independent annotators, yielding up to 600 judgments across the two datasets. Annotators will be fluent in the language of the materials and independent of the author team. Before formal annotation, the execution guide specifies 10 practice instances (five per dataset) drawn from separate training or development materials to learn the instructions and rating scale. Practice judgments will be excluded from the reported results. Formal annotation will be completed independently, without access to other annotators' judgments or AI assistance.

For each instance, annotators will see the current request, its necessary context or source post, the user's training preference pairs, and two responses labeled A and B. History entries will include the historical prompt and the preferred and dispreferred responses, with entry identifiers for citing evidence. The default display includes the complete training history with only exact duplicate preference records removed; distinct response pairs for the same historical prompt remain visible; any necessary display limit must be fixed during practice and applied uniformly before formal annotation. The exact displayed history will be retained. Method identities, selected attributes, automated scores, and held-out preference labels will be hidden. Response order will be randomized independently for

Score	Overall task-quality criterion
1	Fails the request or contains severe errors that make the response unusable.
2	Partly relevant, with major omissions, errors, or violations of explicit requirements.
3	Basically fulfills the request but has substantive, fixable problems.
4	Fulfills the main requirements correctly and relevantly, with only minor problems.
5	Fully fulfills the request, with no evident material errors or omissions within what the annotator can assess.

Table 16: Rating anchors for the planned human evaluation. These are scale definitions, not observed scores.

each annotator and balanced within each dataset; instance order will also be randomized. Organizers will retain the mapping from presentation identifiers to methods separately from the rating forms. Identical responses will remain eligible.

Preference fit. Annotators will select A, B, *Tie*, or *Insufficient evidence* according to which response better matches the user's preferences inferred from the displayed history and current request. A tie indicates comparable preference fit supported by the available evidence; insufficient evidence indicates that the history does not support a reliable comparison. Annotators will provide a brief explanation and cite relevant history entries when available. They will be instructed to distinguish the user's observed preferences from their own stylistic preferences.

Task quality. Annotators will independently score each response from 1 to 5 for overall task fulfillment, considering relevance, correctness or source faithfulness, completeness, and explicit requirements. Table 16 defines the scale. This is one overall assessment, rather than separate measurements of these dimensions. An unassessable response will receive a missing score with a reason, rather than a zero. Task-quality scores and preference choices may favor different responses.

F.3 Aggregation and Uncertainty

The primary analysis will retain every individual judgment rather than replacing disagreements with majority votes. After restoring method identities, let $z_{uir} \in \{E, R, T, I\}$ denote the preference judgment of annotator r for instance i of user u: EXACT, RAG, tie, or insufficient evidence. For each dataset, let U denote its evaluated users, T_u their instances with completed judgments, and R_{ui} the number of completed judgments for an instance (target: three). Unscored instances and users will be reported separately rather than assigned zero scores. The proportion for category c is

$$\widehat{p}_c = \frac{1}{|U|} \sum_{u \in U} \frac{1}{|T_u|} \sum_{i \in T_u} \frac{1}{R_{ui}} \sum_{r=1}^{R_{ui}} \mathbf{1}[z_{uir} = c], \qquad c \in \{E, R, T, I\}. \tag{36}$$

We will report all four proportions, their raw counts, and the net preference difference $\Delta_{\text{pref}} = 100(\widehat{p}_E - \widehat{p}_R)$ in percentage points. Ties and insufficient-evidence judgments contribute zero to this difference but remain separate reported outcomes and remain in the denominator. High uncertainty in the history must not be interpreted as evidence that the methods are equivalent. A secondary analysis restricted to judgments with sufficient evidence will report its coverage and excluded users. These human preference statistics are distinct from the likelihood-based pairwise accuracy in Eq. (11).

Task-quality analysis will report score distributions and descriptive means. Paired quality differences will use only judgments with both scores present, averaging annotators within an instance, instances within a user, and then users equally; missing scores and any excluded instances or users will be reported. Mean differences treat adjacent scale points as approximately equally spaced. For each dataset, 95% percentile confidence intervals will use 10,000 bootstrap resamples of users, retaining all of a sampled user's instances and judgments together. A user appearing in multiple runs remains a single cluster. These intervals describe uncertainty over users conditional on the annotator panel; judgments are not treated as independent users.

After restoring method identities, agreement will be summarized by pairwise exact agreement and nominal Krippendorff's α over the four preference categories, alongside their frequencies. Degenerate

cases in which α is undefined will be reported as such. Exclusion rules will be fixed before annotation, and any missing assignments or protocol deviations documented. Ratings will not be excluded because they disagree with an automated judge or favor a particular method. The planned sample size is a feasibility target, not a claim of guaranteed statistical power. Results will be reported for both datasets, including inconclusive differences; a nonsignificant quality difference alone does not establish noninferiority.

F.4 Annotator Instructions and Study Records

The following instructions will accompany the rating anchors in Table 16:

> You will see a user's previous response preferences, a new request, and two anonymous responses. Evaluate the responses independently. Do not try to identify the systems or use an AI assistant to provide judgments.
>
> **Preference fit:** Based on the provided history and the current request, which response is more likely to match this user's preferences? Select A, B, Tie, or Insufficient evidence. Use Tie when the evidence supports similar preference fit, and Insufficient evidence when the history does not support a reliable comparison. Do not substitute your own taste or reward length by itself.
>
> **Task quality:** Rate each response from 1 to 5 for how well it fulfills the current request, using the supplied anchors. Consider relevance, correctness or source faithfulness, completeness, and explicit requirements. Rate quality separately from preference fit. If you cannot assess a response, leave its quality score blank and explain why.
>
> Give a brief reason and cite relevant history entries when available. Your quality scores and preference choice need not favor the same response.

Study records will include the sampling manifest, original generations, displayed histories, assignment-level A/B mappings, individual ratings, and the frozen instructions. Recruitment criteria, language and domain background, practice procedure, actual completion counts, time, compensation, informed consent, and the applicable institutional review determination will be documented for the completed study. For the original prospective protocol, these arrangements must precede collection; the completed-study description must instead report the actual procedures and dates, including any deviations. The planned assessment is restricted to one backbone and the sampled users; it cannot establish generality across all models or directly validate the selected attributes as faithful accounts of users' internal preferences.