PERSONALIZED QUERY ENHANCEMENT VIA REWARD-GUIDED FEEDBACK DESCENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Users in real-world interactive systems often submit vague queries that only partially reflect their latent intent. Query enhancement aims to make such queries more informative by adding context that better captures the user's underlying needs. However, a key challenge is that enhancement quality is difficult to measure: direct satisfaction feedback is sparse, while abundant indirect signals are only noisy proxies. In this paper, we study personalized query enhancement under sparse feedback. We formulate user satisfaction through a hierarchical model that links sparse direct satisfaction signals with denser indirect behavioral feedback, yielding a reward model for evaluating query enhancements. Using this reward model, we define objectives for generating and refining personalized enhancements. To iteratively improve the enhancements, we propose reward-guided feedback descent, a novel text-feedback-based query refinement algorithm. A key novelty of our approach is that it uses the reward model and the enhancement objective as a grounding mechanism for determining the preferred candidate numerically and synthesizing textual feedback, which in turn guides the LLM enhancer toward generating higher-quality enhancements. We evaluate our method on both public benchmarks and real-world e-commerce data, demonstrating that the proposed algorithm consistently improves user satisfaction through enhanced queries.

1 INTRODUCTION

Many real-world services are interactive: users submit queries to express their needs, and the system responds by recommending relevant content. These interactions can be multi-turn, as users iteratively refine their queries to better convey their intent. While the long-term objective of such systems is often revenue, their immediate goal is to improve user satisfaction. However, satisfaction is inherently difficult to quantify. It is not fully captured by observable actions such as clicks or purchases, which only provide indirect and partial signals. Rather, satisfaction reflects a holistic judgment of the overall user experience, and directly measuring it requires costly mechanisms such as explicit user surveys.

Query enhancement offers a practical approach to address this problem by predicting a user's intent and suggesting refined queries that better express it. Such enhancements can improve satisfaction not only for users with clear intent, but also for those with vague or unarticulated needs, by helping them surface and refine their underlying goals. However, the latter case is challenging, as it requires optimizing latent satisfaction that is difficult to observe or quantify. Query enhancement is also well-suited to modern LLMs, which can generate high-quality rewrites with appropriate prompts. Yet, defining what constitutes a "good" enhancement for improving user satisfaction remains a challenge.

While a standard approach is to train a reward model and fine-tune an LLM to maximize it (Christiano et al., 2017; Shao et al., 2024), this is often inefficient for natural language tasks like query enhancement. Standard reinforcement learning provides limited learning signals due to its reliance on sparse scalar rewards. Conversely, when clear textual feedback is available, recent text-driven optimization methods have shown strong performance and substantially greater sample efficiency than reward-based optimization in language tasks (Agrawal et al., 2026; Lee et al., 2025). Although our task is well-suited for this paradigm, it crucially lacks the required textual feedback, presenting a key bottleneck due to the inherent scarcity of direct user satisfaction labels. Fortunately, these systems provide an alternative supervision source: users continuously generate *behavioral signals* (e.g., clicks, purchases, and reviews), which offer abundant but indirect evidence of satisfaction.

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.

Motivated by this, we propose a probabilistic model for latent user satisfaction that estimates satisfaction from heterogeneous behavioral signals. Based on this model, we construct a reward function and define a reward-tilted user-query conditional latent intent distribution. We then derive a reward-guided criterion for evaluating query enhancements from this distribution. Crucially, the enhancement objective determines the preferred candidate numerically, while an LLM converts the resulting objective differences into textual feedback for refinement. Rather than relying on unavailable human critiques, this separates behavior-grounded candidate selection from feedback verbalization and turns non-textual behavioral supervision into guidance for iterative text-based optimization.

Our main contributions are summarized as follows:

- (§2) We propose a probabilistic model to estimate latent user intent and satisfaction directly from behavioral signals. From this, we introduce an uncertainty-based query enhancement criterion and derive a reward-guided feedback mechanism for text-based optimization.
- (§3) To address real-world latency constraints, we build an offline pool of candidate enhancements iteratively refined via our feedback mechanism. Serving then requires only lightweight candidate retrieval and branch selection based on our estimated reward.
- (§4-5) We evaluate our approach on public benchmarks and a real-world e-commerce dataset. Notably, our method even outperforms direct online LLM-based enhancement, which is prohibitively expensive and infeasible in latency-constrained e-commerce settings.

2 FORMULATION

We consider a system $\mathcal{S}$ in which a user submits a query to fulfill an underlying (possibly implicit) intent, e.g., browsing within an e-commerce platform, with or without a concrete purchase goal. Our objective is to enhance the query to better reflect the user's true underlying intent, thereby improving user satisfaction. We focus on text-based queries $q \in \Sigma$ and perform direct query enhancement $\Sigma \to \Sigma$, in the same space. Note that our method is not restricted to this particular setting.

Maximizing user satisfaction. The goal of an agent operating on $\mathcal{S}$ is to maximize the user's *satisfaction* $z \in \mathcal{Z}$, where $\mathcal{Z}$ denotes the space of possible satisfaction levels. We model z as a holistic assessment of the user's experience throughout the interaction. In particular, z is not tied to specific outcomes, e.g., a purchase in e-commerce, but reflects the overall usefulness or quality of the experience. As such, z is inherently difficult to observe without explicit user feedback.

Personalization based on user context. The system $\mathcal{S}$ characterizes each user u through two types of user-specific information: (1) *user attributes* D_u, which represent relatively invariant features such as demographic or financial profiles; and (2) *behavioral signals* y, which capture dynamic interaction logs such as clicks, purchases, and reviews. Throughout the text, we treat user attributes D_u as fixed for a given user u and omit explicit dependence on it in our notation. Furthermore, we represent behavioral signals hierarchically as $y \equiv y^{1:L} = (y^1, \ldots, y^L)$; higher levels (i.e., as the index approaches L) correspond to a stronger but less frequent materialization of user satisfaction z.

2.1 LATENT INTENT AND CONCRETE INTERPRETATION

We formalize a mathematical model of user intent when a user u submits a query q to an interactive system $\mathcal{S}$. Based on this model, we derive a target quantity that captures the user's intent and provides an objective for maximizing user satisfaction z.

Definition 2.1 (Universal Latent Intent Space)**.** Let $\mathcal{H}$ denote the *universal latent intent space*. Each $h \in \mathcal{H}$ represents a possible semantic intent corresponding to an underlying user need.

Definition 2.2 (User-Query Conditional Intent Distribution)**.** For a user u and a query q, let $\pi_{u,q}(h)$ be a *user-query conditional intent distribution* with support $\mathcal{H}_{u,q} \subseteq \mathcal{H}$. We treat $\pi_{u,q}$ as a user-specific base distribution determined by the user attribute D_u.

Generally, an intent h is abstract and thus difficult to instantiate precisely. Hence, we assume that each intent h allows an *interpretation* $b \in \mathcal{B}$, where $\mathcal{B}$ is the set of all possible concrete interpretations. In an e-commerce context, for instance, while h might represent an abstract product need envisioned by the user (e.g., an efficient cleaning device for everyday household maintenance), b would correspond to a specific tangible item that realizes this need (e.g., a particular cordless vacuum cleaner model).

Definition 2.3 (User-Query Conditional Intent-Interpretation Mapping; C.1)**.** An *interpretation* $b \in \mathcal{B}$ is a concrete realization of an abstract intent h. We define an *instantiation mapping* $g_{u,q} : \mathcal{H}_{u,q} \to \mathcal{B}$ that assigns each latent intent h to a specific interpretation $b = g_{u,q}(h)$, thereby projecting the user's abstract need into the system's actionable space. For each $b \in \mathcal{B}$, we define the associated intent subset $\mathcal{H}_{u,q}(b) := \{h \in \mathcal{H}_{u,q} : g_{u,q}(h) = b\}$, which forms a partition of the latent intent space $\mathcal{H}_{u,q}$; that is, they are mutually exclusive and collectively exhaustive across all $b \in \mathcal{B}$.

We assume that $\mathcal{B}$ is a fixed interpretation ontology shared across the original and enhanced queries, so each branch $b \in \mathcal{B}$ has a stable semantic identity across queries. The query-conditioned mapping $g_{u,q}$ may change the latent intents assigned to each branch, but it does not redefine the branch itself. Under the instantiation mapping $g_{u,q}$, the prior probability of a concrete interpretation b is simply the total probability mass of all latent intents that instantiate it. By Bayes' theorem, this allows us to directly define the base intent distribution conditioned on a specific interpretation b:

$$\pi_{u,q}(b) = \textstyle\int_{\mathcal{H}_{u,q}(b)} \pi_{u,q}(h)\mathrm{d}h, \quad \pi_{u,q}(h \mid b) = \pi_{u,q}(h)/\pi_{u,q}(b).$$

2.2 Behavioral Reward

At the moment a user submits a query, the system must infer their intent using only the information available, i.e., the query q and fixed user attributes D_u. Due to the single-turn constraint, the system must act on this inferred intent immediately, without the opportunity to wait for the user's actual downstream feedback. Consequently, to evaluate the utility of a given latent intent h, we cannot observe the true future behavior but must instead estimate its expected value using probabilistic models trained on behavioral signals y from historical interaction logs. As the true user satisfaction z is inherently unobservable, we formalize the probabilistic relationship between intents, posterior behaviors, and latent satisfaction to construct a measurable reward signal.

Definition 2.4 (Satisfaction Hierarchy)**.** For a user u and a query q, let the joint distribution of behavioral signals y and latent satisfaction z, conditioned on a latent intent h, factorize as:

$$p_{u,q}(y, z \mid h) = p_{u,q}(z \mid h)p_{u,q}(y \mid z, h), \quad p_{u,q}(y \mid z, h) = \textstyle\prod_{\ell=1}^{L} p_{u,q}(y^{\ell} \mid y^{<\ell}, z, h).$$

Definition 2.5 (Behavioral Reward)**.** The *behavioral reward* $R_{u,q}(h)$ is defined as the expected satisfaction over the joint distribution of behavioral signals and their posterior satisfaction states:

$$R_{u,q}(h) = \mathbb{E}_{y\sim p_{u,q}(\cdot|h),\, z\sim p_{u,q}(\cdot|y,h)}\left[z\right] \overset{\text{tower rule}}{=} \mathbb{E}_{z\sim p_{u,q}(\cdot|h)}\left[z\right],$$

where the marginal likelihoods and the posterior distribution follow the satisfaction hierarchy:

$$p_{u,q}(y \mid h) = \textstyle\int p_{u,q}(y, z \mid h)\mathrm{d}z,\ p_{u,q}(z \mid h) = \int p_{u,q}(y, z \mid h)\mathrm{d}y,$$
$$p_{u,q}(z \mid y, h) = p_{u,q}(y, z \mid h)/p_{u,q}(y \mid h).$$

The tower rule equivalence above shows that introducing hierarchical behavioral signals $y^{1:L}$ does not change the reward target itself; it rather provides a structured probabilistic mechanism for estimating latent satisfaction z from observable behavioral signals.

Definition 2.6 (Reward-Tilted Distributions)**.** The *reward-tilted intent distribution* and the corresponding *reward-tilted interpretation distribution* are defined as:

$$\tilde{\pi}_{u,q}(h) \propto \pi_{u,q}(h)\exp\left(\gamma R_{u,q}(h)\right), \quad \tilde{\pi}_{u,q}(b) = \textstyle\int_{\mathcal{H}_{u,q}(b)} \tilde{\pi}_{u,q}(h)\mathrm{d}h,$$

where $\gamma > 0$ is a scaling hyperparameter that controls the influence of the behavioral reward.

The reward-tilted interpretation distribution $\tilde{\pi}_{u,q}(b)$ also admits a reward-tilted form using an induced interpretation-level reward $R^{\mathcal{B}}_{u,q}(b)$, i.e., $\tilde{\pi}_{u,q}(b) \propto \pi_{u,q}(b)\exp(\gamma R^{\mathcal{B}}_{u,q}(b))$. Specifically, as detailed in Appendix C.1, the *interpretation-level behavioral reward* $R^{\mathcal{B}}_{u,q}(b)$ is defined as follows; it is lower-bounded by the simple expectation of the *intent-level behavioral reward* $R_{u,q}(h)$ over $\mathcal{H}_{u,q}(b)$:

$$R^{\mathcal{B}}_{u,q}(b) = \frac{1}{\gamma}\log\left(\mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)}\left[\exp\left(\gamma R_{u,q}(h)\right)\right]\right) \geq \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)}\left[R_{u,q}(h)\right].$$

Unlike a simple conditional expectation, this log-sum-exp aggregation emphasizes high-reward latent intents within each branch, with the strength of this emphasis controlled by γ.

2.3 QUERY ENHANCEMENT

With the reward-tilted distributions capturing the user's ideal latent intent, we now formalize the mechanism for evaluating and discovering an optimal query enhancement action.

Definition 2.7 (Personalized Query Enhancement)**.** For a user u and a query q, a *personalized query enhancement action* e produces an enhanced query q', defined by the mapping $e : (u, q) \mapsto q'$.

By operating within the latent intent space $\mathcal{H}$, we ensure that the enhancement process targets the user's core semantic needs before mapping them onto actionable interpretations. In our single-turn constrained scenario, a desirable enhancement should induce a *decisive* intent distribution while remaining faithful to the semantic bounds of the original query q. To ensure mathematical consistency, we assume that the enhanced query q' refines the original intent such that its support is contained within the original intent space, i.e., $\mathcal{H}_{u,q'} \subseteq \mathcal{H}_{u,q}$.

Definition 2.8 (Personalized Posterior Sharpening Objective)**.** We define *personalized posterior sharpening objectives (PESO)* at intent and interpretation levels as follows:

$$\mathrm{PESO}^{\mathcal{H}}_{u,q}(e) := -\mathbb{H}\left[\tilde{\pi}_{u,q'}(h)\right] - \lambda D_{\mathrm{KL}}\left(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)\right), \tag{1}$$

$$\mathrm{PESO}^{\mathcal{B}}_{u,q}(e) := -\mathbb{H}\left[\tilde{\pi}_{u,q'}(b)\right] - \lambda D_{\mathrm{KL}}\left(\tilde{\pi}_{u,q'}(b) \parallel \pi_{u,q}(b)\right), \tag{2}$$

Here, $\lambda \geq 0$ controls the strength of the semantic-faithfulness penalty.

While our fundamental goal is to find an enhancement e that maximizes the intent-level objective (Eq. 1), directly optimizing it is computationally intractable due to the unobservable nature of the latent space $\mathcal{H}$. Under a common intent-to-interpretation coarse-graining, the intent-level PESO decomposes into the interpretation-level objective and a conditional residual term.

Proposition 1 (Intent–Interpretation Relationship in PESO)**.** *The intent-level PESO decomposes as*

$$\mathrm{PESO}^{\mathcal{H}}_{u,q}(e) = \mathrm{PESO}^{\mathcal{B}}_{u,q}(e) - \mathbb{E}_{b\sim\tilde{\pi}_{u,q'}}[\mathbb{H}[\tilde{\pi}_{u,q'}(h \mid b)] + \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h \mid b)\|\pi_{u,q}(h \mid b))].$$

Thus, the interpretation-level objective captures the coarse-grained PESO, while the residual reflects uncertainty and distributional mismatch within each branch. The proof is provided in Appendix C.2.

While PESO provides a convincing information-theoretic measure, its connection to behavioral utility is only implicit through the reward-tilted distribution. To make this utility component explicit and directly encourage higher expected behavioral reward, we introduce an alternative formulation.

Definition 2.9 (Personalized Branch-Utility Alignment Objective)**.** We define *personalized branch-utility alignment objectives (PUBAO)* at intent and interpretation levels as follows:

$$\mathrm{PUBAO}^{\mathcal{H}}_{u,q}(e) := \mathbb{E}_{h\sim\tilde{\pi}_{u,q'}}\left[\log \pi_{u,q'}(h) + \gamma R_{u,q'}(h)\right] - \lambda D_{\mathrm{KL}}\left(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)\right), \tag{3}$$

$$\mathrm{PUBAO}^{\mathcal{B}}_{u,q}(e) := \mathbb{E}_{b\sim\tilde{\pi}_{u,q'}}\left[\log \pi_{u,q'}(b) + \gamma R^{\mathcal{B}}_{u,q'}(b)\right] - \lambda D_{\mathrm{KL}}\left(\tilde{\pi}_{u,q'}(b) \parallel \pi_{u,q}(b)\right), \tag{4}$$

At first glance, minimizing uncertainty (as in PESO) and maximizing expected utility (as in PUBAO) might appear to be two fundamentally distinct optimization paradigms. Remarkably, however, under our reward-tilted formulation, these two objectives are deeply intertwined.

Proposition 2 (Structural Connection between PESO and PUBAO)**.** *By the definition of the reward-tilted distribution, the PUBAO and PESO objectives differ only by a log-partition term.*

$$\mathrm{PUBAO}^{\mathcal{H}}_{u,q}(e) = \mathrm{PESO}^{\mathcal{H}}_{u,q}(e) + \log Z^{\mathcal{H}}_{u,q'}, \quad \textit{where } Z^{\mathcal{H}}_{u,q'} = \textstyle\int_{\mathcal{H}_{u,q'}} \pi_{u,q'}(h)\exp(\gamma R_{u,q'}(h))\,\mathrm{d}h,$$

$$\mathrm{PUBAO}^{\mathcal{B}}_{u,q}(e) = \mathrm{PESO}^{\mathcal{B}}_{u,q}(e) + \log Z^{\mathcal{B}}_{u,q'}, \quad \textit{where } Z^{\mathcal{B}}_{u,q'} = \textstyle\sum_{b\in\mathcal{B}} \pi_{u,q'}(b)\exp\left(\gamma R^{\mathcal{B}}_{u,q'}(b)\right)$$

See Appendix C.3 for the proof. This relationship has practical implications beyond enhancement evaluation. While both PESO and PUBAO share the semantic faithfulness term, PESO summarizes the post-enhancement effect through a single posterior-sharpening term. PUBAO splits this effect into query-induced branch plausibility and reward-aligned branch utility, distinguishing whether an interpretation is merely made decisive or also behaviorally desirable (cf. C.2; connection to the free energy principle). Its finer structure provides the diagnostic basis for the natural-language feedback used in our iterative refinement procedure (cf. Section 4; empirical validation).

3 IMPLEMENTATION

Building on the formulation in Section 2, we seek to identify an enhancement e that yields $q' = e(u, q)$ as a refined version of a query q from a user u, maximizing PESO or PUBAO metrics. Our practical implementation operates at the interpretation level: Section 3.1 details the construction of the reward-tilted interpretation distribution, and Section 3.3 describes the query enhancement process.

3.1 REWARD-TILTED INTERPRETATION DISTRIBUTION

Categorical modeling of interpretation. Assuming a discrete interpretation space $\mathcal{B}$, e.g., a finite product catalog in e-commerce, the user-specific base distribution $\pi_{u,q}(b)$ is modeled as a categorical distribution. By Definition 2.2, $\pi_{u,q}(b)$ is conditioned on the query q and user attributes D_u, thus:

$$\log \pi_{u,q}(b) = \log p(b, D_u, q) - \log p(D_u, q).$$

We parameterize the log-joint probability $\log p(b, D_u, q)$ as an energy-based model that captures the set of pairwise interactions between the query q, the interpretation b, and the user attributes D_u. By mapping these elements into a shared d-dimensional embedding space, we define the joint energy as:

$$\log p(b, D_u, q) = \langle b, D_u \rangle + \langle b, q \rangle + \langle q, D_u \rangle,$$

where the terms represent the user's inherent preference, semantic relevance, and query context bias, respectively. Under this formulation, the term $\log p(D_u, q)$ naturally serves as the log-partition function, i.e., $\mathrm{logsumexp}_{b' \in \mathcal{B}}(\log p(b', D_u, q))$. Notably, since the query context bias term $\langle q, D_u \rangle$ is independent of the interpretation b, it cancels out during the softmax normalization process for a given (u, q) pair. Consequently, the categorical probability of an interpretation $b \in \mathcal{B}$ is computed as:

$$\pi_{u,q}(b) = \mathrm{softmax}\left(\log p(b, D_u, q)\right) \propto \exp\left(\langle b, D_u \rangle + \langle b, q \rangle\right). \tag{5}$$

Reward modeling. The reward model takes a query q, user attributes D_u, and an interpretation b to predict the reward $\hat{R}_{u,q}(b)$. If satisfaction labels z were available for all training examples, this would reduce to a standard supervised learning problem. In our setting, however, only a small fraction of the training data contains z, while the majority of examples provide only behavioral signals $y^{1:L}$. To address this, we construct the reward model as a two-stage predictor. In the first stage, a model $f_y(q, D_u, b)$ predicts the behavioral signals $\hat{y}^{1:L}$ from the inputs (q, D_u, b). In the second stage, another model $f_z(q, D_u, b, \hat{y}^{1:L})$ predicts the reward $\hat{R}_{u,q}(b)$. The first stage leverages the relatively abundant behavioral signals available at scale, whereas the second stage uses only the subset with satisfaction-derived reward labels. This two-stage model provides a deterministic plug-in approximation to the behavioral hierarchy in Section 2, while the learned score $\hat{R}_{u,q}(b)$ estimates conditional-mean satisfaction and serves as a proxy for the theoretical branch reward $R^{\mathcal{B}}_{u,q}(b)$.

Building upon the base interpretation distribution in Eq. 5, we construct a practical reward-tilted interpretation distribution by substituting the learned conditional-mean proxy $\hat{R}_{u,q}(b)$ for the theoretical branch reward $R^{\mathcal{B}}_{u,q}(b)$. Specifically, the learned proxy shifts each branch logit by $\gamma \hat{R}_{u,q}(b)$:

$$\tilde{\pi}_{u,q}(b) \propto \exp\left(\langle b, D_u \rangle + \langle b, q \rangle + \gamma \hat{R}_{u,q}(b)\right). \tag{6}$$

3.2 QUERY ENHANCEMENT WITH REWARD-GUIDED FEEDBACK DESCENT

We describe how to construct a query enhancer e from the learned reward model $\hat{R}_{u,q}$. We use an LLM-based enhancer $e(\cdot \mid q, u, \mathcal{F}_t)$ that takes a query q, user information u, and feedback $\mathcal{F}_t$ at iteration t. Starting from the empty set $\mathcal{F}_0$, the enhancer generates an improved query q_1 from the original query $q_0 := q$ and a natural language description of user attributes D_u.

We adopt the feedback descent (FD) algorithm (Lee et al., 2025) to further improve the enhancer. In the terminology of Lee et al. (2025), the enhancer e corresponds to the proposing model $\mathcal{M}$, while FD additionally requires an evaluator E that compares candidate proposals and produces textual feedback. Rather than relying on textual critiques from a separate evaluator, we instead derive feedback directly from our enhancement objectives (Eq. 2, 4). At each iteration t, the enhancer proposes a challenger rewrite $\tilde{q}_t \sim e(\cdot \mid q, u, \mathcal{F}_t)$ against the current best rewrite q_t. Both candidates are scored by $S_{u,q}$, instantiated by either $\mathrm{PESO}^{\mathcal{B}}_{u,q}$ or $\mathrm{PUBAO}^{\mathcal{B}}_{u,q}$, and the higher-scoring candidate is retained as q_{t+1}:

$$q_{t+1} = \begin{cases} \tilde{q}_t, & S_{u,q}(\tilde{q}_t) > S_{u,q}(q_t), \\ q_t, & \text{otherwise.} \end{cases}$$

Note that objective comparison fixes the winner before feedback generation, so the LLM does not participate in candidate selection and only verbalizes the decomposed objective differences into actionable feedback h_t. The shared KL term $-\lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(b) \parallel \pi_{u,q}(b))$ signals drift from the original intent. PESO adds a posterior-sharpening diagnostic, $-\mathbb{H}[\tilde{\pi}_{u,q'}(b)]$, for detecting overly broad rewrites, whereas PUBAO decomposes this effect into branch plausibility, $\log \pi_{u,q'}(b)$, and reward-aligned utility, $\gamma \hat{R}_{u,q'}(b)$. We append h_t to the feedback history, $\mathcal{F}_{t+1} := \texttt{concat}(\mathcal{F}_t, h_t)$, and repeat the process for T iterations to obtain the finalized enhanced query $q' := q_T$. See Appendix H for the feedback-generation prompts and Appendix G for qualitative refinement examples.

3.3 Tractable Enhancement via Discretization

Without latency constraints, the enhancer LLM can directly apply T steps of FD to map a query q from a user u to an enhanced query q'. However, in many realistic scenarios where even milliseconds of latency matter such as search engines, real-time LLM-based enhancement is often impractical. To address this, we build an offline enhancement pool over prototypical personas and branches to pre-compute enhancements. By shifting generation offline, serving relies solely on lightweight retrieval and reward-based selection, avoiding LLM overhead without sacrificing refinement quality.

Reward models. We instantiate the behavioral reward model with a lightweight tabular estimator such as a Random Forest (Breiman, 2001). This model accommodates nonlinear interactions among query, user, interpretation, and behavioral features under sparse and heterogeneous supervision while remaining efficient for large-scale offline scoring.

Prototypical persona mapping. We approximate the user attribute space with a finite set of N prototypical personas, $\mathcal{U} = \{u_1, \ldots, u_N\}$. For any incoming user u, we assign the nearest persona $u_* \in \mathcal{U}$ based on the similarity of their attribute embeddings. The number of personas N is determined following standard clustering heuristics, such as the silhouette score (Rousseeuw, 1987).

Pre-defined enhancement pool. Instead of learning a generative mapping e over an unconstrained space, we pre-define a finite pool of reference enhanced queries tailored for each persona $u \in \mathcal{U}$ and each interpretation $b \in \mathcal{B}$. For each pair $(u, b) \in \mathcal{U} \times \mathcal{B}$, we first generate an initial enhanced query and then refine it using the reward-guided FD procedure described in Section 3.2. Consequently, the resulting entry $q'_{u,b}$ is designed to clearly represent b within the context of that persona. Importantly, offline refinement and serving-time selection operate at different granularities. Offline FD compares alternative rewrites for a fixed persona–branch pair, whereas serving compares different branches for the same incoming user–query pair using the learned reward within the posterior top-K set.

Test-time retrieval from the enhancement pool. At test time, enhancement reduces to efficient retrieval from the offline pool. Given a user u and query q, we map u to the nearest persona $u_* \in \mathcal{U}$. To avoid the costly evaluation of the reward across all interpretations $\mathcal{B}$ during serving, we use the base distribution $\pi_{u_*,q}(b)$ as a lightweight first-stage retriever for the top-K candidates. We then select the final enhancement q'^* from the pre-computed pool using the learned reward function $\hat{R}_{u_*,q}$:

$$\mathcal{B}_q = \mathrm{TopK}_{b \in \mathcal{B}}\, \pi_{u_*,q}(b), \quad b^* = \arg\max_{b \in \mathcal{B}_q} \hat{R}_{u_*,q}(b), \quad q'^* = q'_{u_*,b^*}.$$

4 Real-World E-commerce Data

Through empirical analysis on a dataset collected from a real-world e-commerce system, we provide evidence for the following claims: Hierarchical behavioral signals enable learning a robust reward model even when latent satisfaction scores are scarce (§4.2). The reward-guided FD algorithm effectively improves personalized query enhancement (§4.3).

4.1 Experimental Setup

Dataset. We used a real-world e-commerce dataset consisting of user search sessions, product interactions, and post-purchase outcomes. Each instance contains a user query (q), user attributes (D_u), a product interpretation (b), and hierarchical behavioral signals ($y^{1:L}$), including exposure, click, cart, wishlist, purchase, review, repurchase, and refund signals. We split the data by (session ID, q) groups to prevent leakage across train, validation, and test sets. This dataset cannot be publicly released due to privacy and copyright concerns. De-identified examples are provided in Appendix F.

Table 1: Reward-model evaluation under varying amounts of latent satisfaction supervision. y supervision denotes how the auxiliary behavioral predictor f_y is trained: not used, trained on the same subset as z, or trained on all available behavioral logs.

z supervision	y supervision	MSE ↓	Spearman ↑	AUROC ↑	z supervision	y supervision	MSE ↓	Spearman ↑	AUROC ↑
0.1%	None	0.0597	0.1826	0.5711	10%	None	0.0528	0.4180	0.6905
	Same subset	**0.0367**	0.2099	0.5841		Same subset	0.0521	0.4401	0.7049
	All logs	0.0550	**0.2793**	**0.6294**		All logs	**0.0485**	**0.5069**	**0.7586**
1%	None	0.0563	0.3273	0.6354	Full	None	0.0498	0.4677	0.7142
	Same subset	**0.0503**	0.3443	0.6471		Same subset	**0.0470**	**0.5334**	**0.7713**
	All logs	0.0521	**0.4334**	**0.7329**		All logs			

LLM as a judge for latent satisfaction. Obtaining direct user satisfaction scores z at scale is prohibitively expensive. Since such scores are nevertheless required as sparse supervision and for offline evaluation, we adopt an LLM-as-a-judge framework (Zheng et al., 2023) to obtain reference satisfaction scores. We separately conduct human evaluations to assess the alignment of our evaluation and selection mechanisms with human judgments. See Appendix B for more details.

Metric. We evaluate an enhanced query by the interpretation branch that it retrieves. For a given query q and enhancement q', let $b(q') = \arg\max_{b \in \mathcal{B}} \pi_{u,q'}(b)$ denote the top branch induced by q'. We report two learned-reward metrics: $R_1(q') = \hat{R}_{u,q'}(b(q'))$ and $R_2(q') = \hat{R}_{u,q}(b(q'))$. Here, R_1 evaluates the enhanced query as a standalone system query, measuring whether q' induces a high-reward interpretation, while R_2 evaluates the retrieved branch under the original user–query context, measuring whether the enhancement improves routing for the original query. For the no-enhancement baseline, both metrics reduce to $\hat{R}_{u,q}(b(q))$ since $q' = q$. We also report the mean reference satisfaction score $\bar{z}$ and its standardized improvement over the original-query baseline, $\Delta\bar{z}/\sigma_0 = (\bar{z} - \bar{z}_0)/\sigma_0$, where $\bar{z}_0$ and σ_0 denote the baseline mean and standard deviation.

4.2 Reward model evaluation.

We first evaluate whether hierarchical behavioral signals help estimate latent satisfaction when satisfaction supervision is sparse. We vary the fraction of reference satisfaction labels z available for training and compare three ways of using behavioral supervision $y^{1:L}$. The *None* setting directly predicts z from (q, D_u, b) without auxiliary behavioral prediction. The *Same subset* setting trains the behavioral predictor f_y only on the corresponding z-labeled subset. The *All logs* setting trains f_y using all available behavioral logs before fitting the satisfaction predictor f_z.

Table 1 shows that behavioral supervision improves reward estimation across z-supervision regimes. The improvement is visible in both regression and ranking metrics: models with behavioral auxiliary prediction generally achieve lower MSE and consistently higher Spearman correlation and AUROC. This trend is especially clear in the sparse-label regime, supporting the use of behavioral signals as intermediate supervision for predicting the sparse reference-satisfaction target from limited z labels.

4.3 Enhancement Evaluation

We next evaluate whether reward-guided feedback descent improves the offline enhancement pool itself. Starting from the initial pool, we apply up to three FD refinement steps, retaining the higher-scoring candidate at each step for subsequent refinement. We evaluate the selected queries using R_1, R_2, the reference satisfaction score $\bar{z}$, and the standardized satisfaction improvement $\Delta\bar{z}/\sigma_0$.

Table 2: Offline pool refinement under reward-guided FD.

Method	R_1	R_2	$\bar{z}$	$\Delta\bar{z}/\sigma_0$
Original query	8.459	8.459	7.599	–
Initial pool	8.475	8.084	7.576	-0.0265
1-step FD	8.644	8.431	7.709	+0.1274
2-step FD	8.683	8.478	7.741	+0.1644
3-step FD	**8.701**	**8.499**	**7.747**	**+0.1722**

Table 2 shows that reward-guided FD steadily improves the offline enhancement pool across both reward metrics and latent satisfaction. The largest gain occurs after the first refinement step, while subsequent iterations provide smaller but consistent improvements. A two-sided paired t-test against the original query confirms significant improvements in $\bar{z}$ at all FD steps ($p < 0.001$), reaching $\Delta\bar{z}/\sigma_0 = 0.1722$ after three steps. These gains indicate that FD improves not only the reward-based enhancement objectives but also the estimated standalone satisfaction of the enhanced queries.

Baselines. We compare PESO and PUBAO as FD objectives and evaluate production-style and FD-style baselines. Production baselines include BM25+RM3 (Robertson et al., 1994; Abdul-Jaleel et al.,

Table 4: Comparison with production-style and FD-style baselines on the e-commerce dataset. Paired p-values are from two-sided paired t-tests on satisfaction scores against the original-query baseline.

Method	$R_1 \uparrow$	$R_2 \uparrow$	$\bar{z} \uparrow$	$\Delta\bar{z}/\sigma_0 \uparrow$	Paired p
Original query	8.459	8.459	7.599	–	–
BM25 + RM3	8.556	8.347	7.620	+0.0249	0.027
Rocchio-PRF	8.554	8.209	7.609	+0.0110	0.008
Persona-aware PRF	8.531	8.436	7.690	+0.1060	< 0.001
Original query + reranker	8.415	8.415	7.693	+0.1093	< 0.001
`Static` (1-step)	8.509	8.188	7.627	+0.0329	0.049
`Persona-blind` (1-step)	8.582	8.278	7.702	+0.1194	< 0.001
PUBAO-FD (1-step)	8.644	8.431	7.709	+0.1274	< 0.001
Rubric (1-step)	8.576	8.225	7.808	+0.2420	< 0.001
`Text-feature` (3-step)	8.489	8.461	7.640	+0.0474	< 0.001
`Compare` (3-step)	8.592	8.310	7.712	+0.1306	< 0.001
PUBAO-FD (3-step)	**8.701**	**8.499**	**7.747**	**+0.1722**	< 0.001
Rubric (3-step)	8.589	8.222	7.942	+0.3982	< 0.001

2004), which expands queries using terms from pseudo-relevant results; Rocchio-PRF (Rocchio Jr, 1971), which shifts the query representation toward the pseudo-relevant centroid; Persona-aware PRF, which additionally incorporates persona context into expansion; and Original-query reranking, which fixes the query and reranks the posterior branches using a pretrained relevance reranker. For FD-style baselines, `Static` replaces feedback with a generic instruction, `Persona-blind` removes persona information, and `Text-feature` directly provides all textualized behavioral signals instead of reward-derived feedback. `Compare` lets the LLM select the pairwise winner, providing a close analogue of Feedback Descent (Lee et al., 2025). Finally, ***Rubric*** is an oracle baseline that uses the reference satisfaction rubric for both candidate scoring and feedback generation.

Results. Table 3 compares PESO and PUBAO as FD objectives across refinement steps. Both objectives improve with additional refinement, while PUBAO consistently performs better at every step. A two-sided paired t-test further shows that PUBAO significantly outperforms PESO at each refinement step ($p < 0.001$). At three steps, PUBAO improves over PESO by 0.109 in R_1, 0.099 in R_2, and 0.035 in $\bar{z}$, while reaching $\Delta\bar{z}/\sigma_0 = 0.1722$. This consistent advantage of PUBAO across multiple steps supports the contribution of the explicit reward-aligned utility term.

Table 3: Ablation on the FD objective: PESO vs. PUBAO across FD steps.

Step	Objective	R_1	R_2	$\bar{z}$	$\Delta\bar{z}/\sigma_0$
1	PESO	8.539	8.347	7.669	+0.0806
	PUBAO	**8.644**	**8.431**	**7.709**	**+0.1274**
2	PESO	8.580	8.384	7.697	+0.1138
	PUBAO	**8.683**	**8.478**	**7.741**	**+0.1644**
3	PESO	8.592	8.400	7.712	+0.1310
	PUBAO	**8.701**	**8.499**	**7.747**	**+0.1722**

Table 4 compares PUBAO-FD with production-style and FD-style baselines. PUBAO-FD outperforms both baseline groups and achieves the best R_1, R_2, and $\bar{z}$ among non-oracle methods, reaching $\Delta\bar{z}/\sigma_0 = 0.1722$. A two-sided paired t-test against the original query confirms a statistically significant improvement in $\bar{z}$ ($p < 0.001$). These results support personalized, reward-grounded feedback over conventional query-enhancement and alternative FD baselines. Note that ***Rubric*** oracle attains higher $\bar{z}$ using the reference satisfaction rubric, which is not available in the target setting.

5 Public Benchmark Study: LongMemEval

5.1 Experimental Setup

Dataset and protocol. We evaluate on LongMemEval (Wu et al., 2024), a public benchmark for long-term conversational memory QA. We use the harder LongMemEval-M setting, where each question is paired with roughly 500 memory sessions and only a few contain answer-relevant evidence. Because LongMemEval provides no official train/test partition for reward-model training, we construct a fixed 300/200 split stratified by question type. We use the training split only for benchmark-specific reward-model training and evaluate all methods on the held-out test split. To ensure a fair comparison, we reimplement and evaluate all methods under the same fixed split and generation pipeline. The exact split and implementation details are provided in Appendix D.2.

Setup. We map the memory history, current question, and candidate memory sessions to the user context, input query, and interpretation space in our framework, respectively. The benchmark-specific

Table 5: LongMemEval results for first-stage retrieval (left) and shared reranking (right).

Method	R@1	R@5	NDCG@5	R@10	QA Acc.	Method	R@1	R@5	NDCG@5	R@10	QA Acc.
Original query	0.104	0.592	0.615	0.640	0.605	Original query	0.207	0.756	0.807	0.787	0.690
RM3	0.085	0.573	0.587	0.689	0.565	RM3	0.207	0.756	0.816	0.799	0.710
Rocchio	0.092	0.646	0.637	0.732	0.660	Rocchio	0.213	0.787	0.824	0.835	0.685
Key Expansion	0.183	0.720	0.753	0.799	0.685	Key Expansion	0.226	0.829	0.863	0.890	0.750
PBR	0.171	0.726	0.759	0.787	0.685	PBR	0.232	0.799	0.853	0.860	0.735
Direct-rewrite	**0.220**	0.720	0.764	0.829	0.705	Direct-rewrite	**0.238**	0.829	**0.871**	0.890	0.730
Reward-Guided FD	0.189	**0.762**	**0.792**	**0.848**	**0.715**	**Reward-Guided FD**	**0.238**	**0.842**	**0.871**	**0.896**	**0.780**

reward model is trained using answer-session annotations from the training split to estimate whether a candidate session contains useful evidence. Neither gold answers nor answer-session identifiers are exposed during test-time query generation or reranking. Each method retrieves memory sessions, after which the top five are passed to a fixed GPT-5.2 reader and evaluated with a GPT-4o judge.

Retrieval settings. We evaluate two settings to separate query-enhancement quality from downstream reranker effects. The first-stage setting uses the base retriever directly, while the shared-reranking setting applies the same `BAAI/bge-reranker-v2-m3` cross-encoder reranker to each method's retrieved candidate set before selecting the final sessions. This allows us to test whether the benefit of query enhancement persists when all methods share the same stronger downstream reranking stage. For Reward-Guided FD, the reward model is used only to construct the enhanced query q' and is not applied again after the shared reranking stage.

Baselines and evaluation. We reuse RM3 and Rocchio from Section 4 and additionally compare Key Expansion (Wu et al., 2024), PBR (Zhang et al., 2026), and direct LLM rewriting. RM3 and Rocchio provide classical pseudo-relevance-feedback baselines using top-retrieved sessions, Key Expansion provides a strong lexical expansion baseline, and PBR explicitly incorporates user context into query expansion. Direct rewriting uses an LLM to rewrite the question without reward-guided iterative refinement, providing the closest generation-based baseline. We report Recall@1, Recall@5, NDCG@5, Recall@10, and downstream QA accuracy in both retrieval settings.

5.2 Results

Without cross-encoder reranking, Reward-Guided FD achieves the best Recall@5, NDCG@5, Recall@10, and QA accuracy, while Direct-rewrite performs best at Recall@1. Its strongest gains appear in broader evidence retrieval, reaching Recall@5 of 0.762 and QA accuracy of 0.715. Classical expansion methods such as RM3 and Rocchio remain clearly below the stronger personalized and LLM-based baselines, indicating that simple pseudo-relevance feedback is insufficient.

With shared reranking, Reward-Guided FD remains best or tied across the retrieval metrics and reaches the highest QA accuracy of 0.780. The improvement persists when all methods share the reranker, suggesting that the gains arise from better query enhancement rather than a method-specific ranking advantage. Because the reward model is trained using only benchmark-native supervision, these results demonstrate that the framework can be adapted beyond the e-commerce setting.

6 Conclusion

We introduced reward-guided feedback descent for personalized query enhancement under sparse satisfaction feedback. From a probabilistic behavioral reward model, we derived PESO and PUBAO to evaluate candidate enhancements and expose decomposed objective differences as actionable textual feedback. This design separates numerical candidate selection from LLM feedback verbalization, enabling iterative refinement of an offline enhancement pool while serving requires only lightweight retrieval and reward-based branch selection. Experiments on real-world e-commerce data show consistent improvements over production-style and FD-style baselines, while LongMemEval demonstrates gains in evidence retrieval and downstream question answering both before and after shared reranking. Overall, our results suggest that behavioral reward models can serve not only as ranking signals, but also as structured guidance for iterative language refinement.

Limitations. Our e-commerce study is offline and proprietary, limiting reproducibility and causal claims about deployed outcomes; LongMemEval provides complementary public validation. Behavioral signals are imperfect proxies for satisfaction and may reflect exposure or position bias. Personalization may use sensitive user attributes, requiring privacy and fairness safeguards.

AI USE STATEMENT

In this work, generative AI models were used as experimental components for personalized query generation and refinement, objective-grounded feedback verbalization, reference satisfaction scoring, and downstream evaluation, as described in the paper. In particular, GPT-5.2 was used for LLM-based enhancement components, reference satisfaction scoring, and the LongMemEval reader, while GPT-4o was used for downstream LongMemEval answer evaluation. We also used generative AI tools to assist with research methodology and experiment refinement, software implementation and data processing, and drafting and editing parts of the manuscript. All AI-assisted code, analyses, experimental outputs, and manuscript content were reviewed and verified by the authors for correctness and consistency with the underlying experiments. The authors take full responsibility for the final content of this work, including all text, claims, code, analyses, and experimental results produced with the aid of generative AI.

ETHICS STATEMENT

Our e-commerce experiments use de-identified historical interaction data and user-profile information, including potentially sensitive attributes, under restricted access and data-handling procedures. Only de-identified examples are included in the paper, and the proprietary dataset is not released because of privacy and copyright constraints. The human evaluations assess query quality and interpretation alignment using research stimuli without exposing direct user identifiers or collecting additional sensitive user information. Personalized systems based on behavioral and profile data may nevertheless inherit privacy, fairness, and proxy-related biases, and appropriate access control, consent, monitoring, and fairness safeguards are necessary before deployment.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the probabilistic formulation, Reward-Guided Feedback Descent procedure, experimental protocols, baseline implementations, evaluation metrics, and statistical testing in the main paper and appendices. Appendix D provides additional implementation details for the e-commerce and LongMemEval experiments, while Appendix E reports robustness and diagnostic analyses. The supplementary code includes the exact question IDs for the fixed 300/200 LongMemEval train/test split. Theoretical assumptions and proofs are provided in Appendix C.2 and Appendix C.3, and the LLM prompts and de-identified data examples are provided in Appendix H and Appendix F. Although the proprietary e-commerce data cannot be released, the public LongMemEval and BESPOKE evaluations provide independently accessible settings for reproducing and further evaluating the proposed framework.

REFERENCES

Nasreen Abdul-Jaleel, James Allan, W Bruce Croft, Fernando Diaz, Leah Larkey, Xiaoyan Li, Mark D Smucker, and Courtney Wade. Umass at trec 2004: Novelty and hard. In *Proceedings of the Thirteenth Text REtrieval Conference*, 2004.

L. A. Agrawal, S. Tan, D. Soylu, N. Ziems, R. Khare, K. Opsahl-Ong, A. Singhvi, H. Shandilya, M. J. Ryan, M. Jiang, C. Potts, K. Sen, A. G. Dimakis, I. Stoica, D. Klein, M. Zaharia, and O. Khattab. GEPA: reflective prompt evolution can outperform reinforcement learning. In *Proceedings of The 14th International Conference on Learning Representations (ICLR 2026)*, 2026.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

P. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

Karl Friston, Francesco Rigoli, Dimitri Ognibene, Christoph Mathys, Thomas Fitzgerald, and Giovanni Pezzulo. Active inference and epistemic value. *Cognitive neuroscience*, 6(4):187–214, 2015.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1762–1777, 2023.

Pranav Kasela, Marco Braga, Gabriella Pasi, and Raffaele Perego. Se-pqa: Personalized community question answering. In *Companion Proceedings of the ACM Web Conference 2024*, pp. 1095–1098, 2024.

Hyunseo Kim, Sangam Lee, Kwangwook Seo, and Dongha Lee. Bespoke: Benchmark for search-augmented large language model personalization via diagnostic feedback. *arXiv preprint arXiv:2509.21106*, 2025.

K. Kobalczyk, N. Astorga, T. Liu, and M. V. D. Schaar. Active task disambiguation with LLMs. In *Proceedings of The 13th International Conference on Learning Representations (ICLR 2025)*, 2025.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Clam: Selective clarification for ambiguous questions with generative language models. *arXiv preprint arXiv:2212.07769*, 2022.

Yoonho Lee, Joseph Boen, and Chelsea Finn. Feedback descent: Open-ended text optimization via pairwise comparison. *arXiv preprint arXiv:2511.07919*, 2025.

Belinda Z Li, Alex Tamkin, Noah Goodman, and Jacob Andreas. Eliciting human preferences with language models. *arXiv preprint arXiv:2310.11589*, 2023.

Xiaopeng Li, Lixin Su, Pengyue Jia, Suqi Cheng, Junfeng Wang, Dawei Yin, and Xiangyu Zhao. Agent4ranking: semantic robust ranking via personalized query rewriting using multi-agent llms. *ACM Transactions on Information Systems*, 43(6):1–33, 2025.

Jiongnan Liu, Zhicheng Dou, Guoyu Tang, and Sulong Xu. Jdsearch: A personalized product search dataset with real queries and full interactions. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2945–2952, 2023.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5303–5315, 2023.

Shengyu Mao, Yong Jiang, Boli Chen, Xiao Li, Peng Wang, Xinyu Wang, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Rafe: Ranking feedback improves query rewriting for rag. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 884–901, 2024.

Beren Millidge, Alexander Tschantz, and Christopher L Buckley. Whence the expected free energy? *Neural Computation*, 33(2):447–482, 2021.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 188–197, 2019.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7957–7968, 2023.

Stephen E Robertson and K Sparck Jones. Relevance weighting of search terms. *Journal of the American Society for Information science*, 27(3):129–146, 1976.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. Okapi at TREC-3. In Donna K. Harman (ed.), *Proceedings of The Third Text REtrieval Conference, TREC 1994, Gaithersburg, Maryland, USA, November 2-4, 1994*, volume 500-225 of *NIST Special Publication*, pp. 109–126. National Institute of Standards and Technology (NIST), 1994. URL http://trec.nist.gov/pubs/trec3/papers/city.ps.gz.

Joseph John Rocchio Jr. Relevance feedback in information retrieval. *The SMART retrieval system: experiments in automatic document processing*, 1971.

Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

Alireza Salemi and Hamed Zamani. Lamp-qa: A benchmark for personalized long-form question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 1139–1159, 2025.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. K. Li, Y. Wu, and D. Guo. Deepseekmath: pushing the limits of mathematical reasoning in open language models, 2024. arXiv:2402.03300.

Yuda Song, Lili Chen, Fahim Tajwar, Remi Munos, Deepak Pathak, J Andrew Bagnell, Aarti Singh, and Andrea Zanette. Expanding the capabilities of reinforcement learning via text feedback. *arXiv preprint arXiv:2602.02482*, 2026.

Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9414–9423, 2023.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory, 2024. URL https://arxiv.org/abs/2410.10813.

Yingyi Zhang, Pengyue Jia, Derong Xu, Yi Wen, Xianneng Li, Yichao Wang, Wenlin Zhang, Xiaopeng Li, Weinan Gan, Huifeng Guo, et al. Personalize before retrieve: Llm-based personalized query expansion for user-centric retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 16406–16414, 2026.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A Related Work

Query reformulation and enhancement. Query reformulation has long been studied to bridge the gap between underspecified user queries and the information required by retrieval systems (Rocchio Jr, 1971; Robertson & Jones, 1976). Classical pseudo-relevance feedback methods use top-ranked results as relevance evidence: Rocchio shifts the query representation toward the pseudo-relevant centroid, while BM25+RM3 expands the query using terms estimated from BM25-retrieved documents (Rocchio Jr, 1971; Robertson et al., 1994; Abdul-Jaleel et al., 2004). Recent LLM-based approaches further treat query text itself as an optimization target. Query2Doc uses LLM-generated pseudo-documents for query expansion (Wang et al., 2023), while HyDE embeds hypothetical documents for zero-shot retrieval (Gao et al., 2023). Rewrite-Retrieve-Read adapts a query rewriter to better align the query with downstream retrieval and reading modules (Ma et al., 2023). PBR further incorporates user context into personalized query expansion before retrieval (Zhang et al., 2026). RaFe uses ranking feedback from a reranker for LLM-based query rewriting, while Agent4Ranking studies iterative personalized query rewriting within a ranking framework (Mao et al., 2024; Li et al., 2025). Unlike these approaches, our method uses sparse satisfaction supervision and behavioral signals to derive both candidate selection and directional textual feedback for iterative personalized refinement.

Ambiguity resolution and preference elicitation. Another related line of work addresses ambiguous or underspecified user requests through clarification. CLAM studies when language models should ask clarifying questions rather than directly answering ambiguous questions (Kuhn et al., 2022). Active Task Disambiguation formalizes task ambiguity and uses Bayesian experimental design to select clarifying questions that reduce uncertainty over possible task specifications (Kobalczyk et al., 2025). Similarly, GATE uses language models to elicit human preferences and task specifications through free-form interaction, showing that model-generated questions or edge cases can uncover informative user preferences (Li et al., 2023). These approaches acquire additional information by interacting with the user. Our setting is different: we consider one-shot query enhancement in an interactive system, where the system must act immediately using the original query, user context, and historical behavioral signals, without asking the user additional questions.

Personalization datasets and behavioral feedback. Recent benchmarks have emphasized personalization, long-term user context, and behavioral interaction in language and retrieval systems. LongMemEval evaluates long-term conversational memory through evidence retrieval and question answering over extensive interaction histories (Wu et al., 2024). BESPOKE evaluates search-augmented LLM personalization using authentic user histories, detailed information needs, fine-grained preference scores, and diagnostic feedback (Kim et al., 2025). LaMP-QA and SE-PQA provide additional public settings for personalized question answering using historical user information and personalized information needs (Salemi & Zamani, 2025; Kasela et al., 2024). Amazon product-review datasets provide large-scale reviews, product metadata, and logs, but are centered on reviews and associated products rather than user-issued search queries (Ni et al., 2019). JDsearch provides real user queries, behavioral interactions, and product metadata, although anonymized query and product tokens make it better suited to structural personalization studies (Liu et al., 2023). Our work instead targets a deployment-oriented setting where direct satisfaction labels are sparse and most supervision comes from naturally occurring behavioral signals. In our e-commerce setting, each example couples a user query, user attributes, candidate interpretation branches, and hierarchical behavioral signals such as exposure, click, purchase, review, repurchase, and refund. We use these dense but indirect signals to estimate latent satisfaction and construct a reward model for personalized query enhancement.

Text-feedback-based optimization. A growing body of work uses natural-language feedback as a richer learning signal than scalar rewards. RL from Text Feedback (RLTF) incorporates textual critiques into reinforcement learning through self-distillation and feedback modeling, allowing the policy to internalize feedback during training (Song et al., 2026). ProTeGi performs automatic prompt optimization by generating natural-language "gradients" that criticize the current prompt and editing the prompt in the opposite semantic direction (Pryzant et al., 2023). GEPA uses natural-language reflection over rollout trajectories to evolve prompts and learn high-level rules from trial and error, often with fewer rollouts than reinforcement-learning-based optimization (Agrawal et al., 2026). Feedback Descent generalizes this idea to open-ended text artifacts by iteratively improving candidates through pairwise comparisons and structured textual feedback (Lee et al., 2025).

Our method is closest in spirit to Feedback Descent, but differs in how the comparison and feedback are grounded. In standard Feedback Descent, an evaluator typically both compares candidate artifacts and provides textual critiques. In our setting, however, reliable textual critiques are not naturally available, and an LLM-only evaluator may conflate linguistic plausibility with user satisfaction. We therefore first determine the winner and loser using an external behavioral-reward objective, instantiated by PESO or PUBAO, and then ask the LLM to verbalize the metric-induced preference into actionable feedback. The feedback is grounded in decomposed diagnostics such as KL faithfulness, posterior sharpening, branch plausibility, and reward-aligned utility. This decouples candidate evaluation from feedback verbalization, allowing text-based optimization to exploit the directional richness of natural-language feedback while remaining grounded in non-linguistic behavioral evidence.

B Human Evaluation

We evaluate human preferences at three distinct stages: offline query refinement, serving-time branch selection, and end-to-end query comparison. Study 1 evaluates query-level decisions while fixing the user and interpretation branch, whereas Study 2 evaluates branch-level decisions while fixing the user and incoming query. Study 3 compares the final queries produced by our method and baseline methods. Study 1 uses 300 pairwise query comparisons collected in the original human evaluation involving 15 annotators. Study 2 and Study 3 involve 12 annotators divided into four groups, with three annotators in each group. Each study includes 120 items, and each item is evaluated by three annotators. The annotation targets are human preference and human-rated relevance.

B.1 Study 1: Offline query-refinement decisions.

For a fixed user u and interpretation branch b, we examine whether objective-based selectors agree with human preferences between candidate queries. This evaluates the decision used to guide offline refinement: choosing between queries while holding the intended interpretation fixed. We compare PUBAO, PESO, and an RF-only selector based on the learned reward score $\hat{R}$.

Table 6: Agreement with human preferences in offline refinement decisions.

Selector	Agreement (%)	95% CI
PUBAO	65.6	[59.7, 71.1]
PESO	67.2	[61.3, 72.6]
RF-only $\hat{R}$	59.2	[53.4, 64.8]

Table 6 shows numerically higher agreement for PESO and PUBAO than for the RF-only selector on this query-level task. Serving-time selection makes a different decision, comparing branches for the same incoming query, which we evaluate separately below.

B.2 Study 2: Serving-time branch selection.

We fix each user–query pair (u, q) and compare posterior-only selection with reward reranking. Following the serving procedure, let u_* denote the persona assigned to user u, and let $\mathcal{B}_q = \mathrm{TopK}_{b\in\mathcal{B}}\, \pi_{u_*,q}(b)$ denote the candidate set with $K = 200$. The posterior-only baseline and the reward-based serving selector choose

$$b_\pi = \arg\max_{b\in\mathcal{B}} \pi_{u_*,q}(b), \qquad b^* = \arg\max_{b\in\mathcal{B}_q} \hat{R}_{u_*,q}(b). \tag{7}$$

Each item contains four candidate branches rated on a 1–5 scale by three annotators. Writing $\bar{h}_{u,q}(b)$ for the mean human rating of branch b for the user–query pair (u, q), we compare the selected branches through

$$\Delta h_{u,q} = \bar{h}_{u,q}(b^*) - \bar{h}_{u,q}(b_\pi). \tag{8}$$

Positive, negative, and zero differences indicate a win, a loss, and a tie for reward reranking, respectively. The selectors choose the same branch on 8 of the 120 items, leaving 112 items with distinct selections for the paired comparison.

As shown in Table 7, reward reranking wins 69.5% of non-tied comparisons. The mean human rating increases by 0.351 points. The paired differences are statistically significant under both the Wilcoxon test ($p = 0.0016$) and the paired-t test ($p = 0.0094$), supporting reward reranking for serving-time branch selection.

Table 7: Human comparison of reward reranking against posterior-only selection.

Metric	Result
Reward Wins / losses / ties	66 / 29 / 17
Win rate excluding ties	69.5% [59.6, 77.8]

B.3 Study 3: End-to-end query comparison.

We conduct a blind four-arm comparison of the original query, word-based RM3, word-based Rocchio, and three-step PUBAO-FD. The original query appears as a candidate and is not displayed separately as a reference. The 12 annotators evaluate 120 items, with three annotators per item each selecting one of four candidate queries, yielding $120 \times 3 = 360$ individual choices. A majority choice is obtained for 90 items, while the remaining 30 receive three votes for different candidates. Confidence intervals are estimated using an item-level cluster bootstrap that retains all three judgments for each resampled item, with the annotator panel held fixed.

Table 8: Blind end-to-end query preferences, with 95% item-level cluster-bootstrap confidence intervals in brackets.

	Individual choices		Item-level majorities	
Method	**Count**	**Share (%)**	**Count**	**Share (%)**
PUBAO-FD	145/360	**40.3** [35.0, 45.6]	47/90	**52.2** [42.6, 62.1]
RM3	91/360	25.3 [20.8, 30.0]	21/90	23.3 [15.6, 31.4]
Rocchio	70/360	19.4 [16.1, 22.8]	13/90	14.4 [8.0, 21.5]
Original	54/360	15.0 [11.4, 18.6]	9/90	10.0 [4.3, 16.3]

As shown in Table 8, PUBAO-FD receives the largest share of both individual choices (40.3%) and items with a majority decision (52.2%). For the pre-specified comparison with the original query, PUBAO-FD accounts for $145/199 = 72.9\%$ of the judgments selecting either PUBAO-FD or the original query. These descriptive results favor PUBAO-FD under the evaluated annotation protocol.

C Theoretical Remarks and Proofs

Remark C.1 (User-Query Conditional Intent-Interpretation Mapping; 2.3)**.** In this framework, the mapping $g_{u,q}$ is conditioned on the user-query context, reflecting that the same latent intent h may be instantiated differently depending on the user u and the specific query q submitted. In an e-commerce context, for instance, a latent intent h representing 'protective winter outerwear' might be mapped to a heavy down parka (b) for a user u in a subarctic climate, but instantiated as a lightweight waterproof windbreaker (b') for a user u' in a milder region.

Remark C.2 (Connection to Free Energy Principle)**.** PUBAO conceptually aligns with expected free energy (EFE; Friston et al., 2015; Millidge et al., 2021) style action selection, where choosing an enhancement e balances the expected utility of a branch against a KL penalty that enforces semantic faithfulness to the original query q. However, unlike the standard EFE, our formulation omits future observation models and information-seeking terms because no additional user responses are observed. Consequently, PUBAO is a specialized adaptation of EFE principles for one-shot query enhancement.

C.1 Derivation of the Interpretation-Level Behavioral Reward

We derive the interpretation-level reward $R^{\mathcal{B}}_{u,q}(b)$ that makes the reward-tilted interpretation distribution $\tilde{\pi}_{u,q}(b)$ consistent with the reward-tilted intent distribution $\tilde{\pi}_{u,q}(h)$. Recall Definition 2.6:

$$\tilde{\pi}_{u,q}(h) = \frac{\pi_{u,q}(h)\exp(\gamma R_{u,q}(h))}{Z^{\mathcal{H}}_{u,q}}, \qquad Z^{\mathcal{H}}_{u,q} = \int_{\mathcal{H}_{u,q}} \pi_{u,q}(h)\exp(\gamma R_{u,q}(h))\; dh.$$

The induced distribution over interpretations b is obtained by marginalizing over the corresponding intent subset $\mathcal{H}_{u,q}(b)$. Substituting the definition of $\tilde{\pi}_{u,q}(h)$ into this marginalization yields

$$\begin{aligned}\tilde{\pi}_{u,q}(b) &= \int_{\mathcal{H}_{u,q}(b)} \tilde{\pi}_{u,q}(h)\, dh \\ &= \frac{1}{Z^{\mathcal{H}}_{u,q}} \int_{\mathcal{H}_{u,q}(b)} \pi_{u,q}(h)\exp(\gamma R_{u,q}(h))\; dh.\end{aligned}$$

Using $\pi_{u,q}(b) = \int_{\mathcal{H}_{u,q}(b)} \pi_{u,q}(h)\, dh$ and $\pi_{u,q}(h \mid b) = \pi_{u,q}(h)/\pi_{u,q}(b)$, we can rewrite $\tilde{\pi}_{u,q}(b)$ as

$$\tilde{\pi}_{u,q}(b) = \frac{\pi_{u,q}(b)}{Z^{\mathcal{H}}_{u,q}} \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[\exp(\gamma R_{u,q}(h))\right].$$

To express this in the branch-level tilted form $\tilde{\pi}_{u,q}(b) \propto \pi_{u,q}(b) \exp\big(\gamma R^{\mathcal{B}}_{u,q}(b)\big)$, the interpretation-level reward must satisfy

$$\exp\big(\gamma R^{\mathcal{B}}_{u,q}(b)\big) = \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[\exp(\gamma R_{u,q}(h))\right].$$

Therefore,

$$R^{\mathcal{B}}_{u,q}(b) = \frac{1}{\gamma} \log \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[\exp(\gamma R_{u,q}(h))\right].$$

This shows that the interpretation-level reward is a log-sum-exp aggregation of intent-level rewards within the interpretation branch, rather than the simple conditional expectation of $R_{u,q}(h)$. Finally, since $\log x$ is concave, Jensen's inequality gives

$$\log \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[\exp(\gamma R_{u,q}(h))\right] \geq \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[\log \exp(\gamma R_{u,q}(h))\right] = \gamma \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[R_{u,q}(h)\right].$$

Dividing both sides by $\gamma > 0$ yields

$$R^{\mathcal{B}}_{u,q}(b) \geq \mathbb{E}_{h\sim\pi_{u,q}(\cdot|b)} \left[R_{u,q}(h)\right].$$

Thus, the branch-level reward emphasizes high-reward latent intents within the branch, with the strength of this emphasis controlled by γ.

C.2 Proof of Proposition 1 (Intent–Interpretation Relationship in PESO)

Under the common intent-to-interpretation coarse-graining assumed in Proposition 1, let $b = g(h)$ denote the shared interpretation assigned to an intent h under the original and enhanced distributions. The interpretation b is therefore a deterministic coarse-graining of h under both distributions. Thus, the reward-tilted distribution and the original distribution induce the joint factorizations

$$\tilde{\pi}_{u,q'}(h, b) = \tilde{\pi}_{u,q'}(b)\tilde{\pi}_{u,q'}(h \mid b),$$
$$\pi_{u,q}(h, b) = \pi_{u,q}(b)\pi_{u,q}(h \mid b).$$

Because b is determined by h, applying the chain rule for entropy gives

$$\mathbb{H}[\tilde{\pi}_{u,q'}(h)] = \mathbb{H}[\tilde{\pi}_{u,q'}(b)] + \mathbb{E}_{b\sim\tilde{\pi}_{u,q'}} \mathbb{H}[\tilde{\pi}_{u,q'}(h \mid b)].$$

Likewise, by the chain rule for KL divergence,

$$D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)) = D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(b) \parallel \pi_{u,q}(b)) + \mathbb{E}_{b\sim\tilde{\pi}_{u,q'}} D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h \mid b) \parallel \pi_{u,q}(h \mid b)).$$

Substituting these two decompositions into the intent-level PESO gives

$$\begin{aligned}
\mathrm{PESO}^{\mathcal{H}}_{u,q}(e) &= -\mathbb{H}[\tilde{\pi}_{u,q'}(h)] - \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)) \\
&= -\mathbb{H}[\tilde{\pi}_{u,q'}(b)] - \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(b) \parallel \pi_{u,q}(b)) \\
&\quad - \mathbb{E}_{b\sim\tilde{\pi}_{u,q'}} \left[\mathbb{H}[\tilde{\pi}_{u,q'}(h \mid b)] + \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h \mid b) \parallel \pi_{u,q}(h \mid b))\right].
\end{aligned}$$

The first two terms are exactly the interpretation-level objective $\mathrm{PESO}^{\mathcal{B}}_{u,q}(e)$. Therefore,

$$\mathrm{PESO}^{\mathcal{H}}_{u,q}(e) = \mathrm{PESO}^{\mathcal{B}}_{u,q}(e) - \mathbb{E}_{b\sim\tilde{\pi}_{u,q'}} \left[\mathbb{H}[\tilde{\pi}_{u,q'}(h \mid b)] + \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h \mid b) \parallel \pi_{u,q}(h \mid b))\right],$$

which proves the claim.

Interpretation. If the enhancement primarily shifts probability mass across interpretations while approximately preserving the conditional micro-intent structure, $\tilde{\pi}_{u,q'}(h \mid b) \approx \pi_{u,q}(h \mid b)$, the conditional KL term becomes negligible. The interpretation-level objective therefore most closely tracks the intent-level objective when the remaining within-branch entropy is also small or approximately stable across candidate enhancements.

C.3 Proof of Proposition 2 (Structural Connection between PESO and PUBAO)

We prove the result for the intent-level objective; the interpretation-level case follows identically. By Definition 2.6, the reward-tilted intent distribution is

$$\tilde{\pi}_{u,q'}(h) = \frac{\pi_{u,q'}(h)\exp(\gamma R_{u,q'}(h))}{Z^{\mathcal{H}}_{u,q'}}, \qquad Z^{\mathcal{H}}_{u,q'} = \int_{\mathcal{H}_{u,q'}} \pi_{u,q'}(h)\exp(\gamma R_{u,q'}(h))\, dh.$$

Taking logarithms gives $\log \pi_{u,q'}(h) + \gamma R_{u,q'}(h) = \log \tilde{\pi}_{u,q'}(h) + \log Z^{\mathcal{H}}_{u,q'}$. Substituting this identity into the intent-level PUBAO objective yields

$$\begin{aligned}
\mathrm{PUBAO}^{\mathcal{H}}_{u,q}(e) &= \mathbb{E}_{h\sim\tilde{\pi}_{u,q'}}\left[\log \pi_{u,q'}(h) + \gamma R_{u,q'}(h)\right] - \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)) \\
&= \mathbb{E}_{h\sim\tilde{\pi}_{u,q'}}\left[\log \tilde{\pi}_{u,q'}(h) + \log Z^{\mathcal{H}}_{u,q'}\right] - \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)) \\
&= -\mathbb{H}[\tilde{\pi}_{u,q'}(h)] + \log Z^{\mathcal{H}}_{u,q'} - \lambda D_{\mathrm{KL}}(\tilde{\pi}_{u,q'}(h) \parallel \pi_{u,q}(h)) \\
&= \mathrm{PESO}^{\mathcal{H}}_{u,q}(e) + \log Z^{\mathcal{H}}_{u,q'}.
\end{aligned}$$

Thus, PUBAO and PESO differ by the log-partition term at the intent level. The same argument applies to the interpretation-level objective. Using

$$\tilde{\pi}_{u,q'}(b) = \frac{\pi_{u,q'}(b)\exp\left(\gamma R^{\mathcal{B}}_{u,q'}(b)\right)}{Z^{\mathcal{B}}_{u,q'}}, \qquad Z^{\mathcal{B}}_{u,q'} = \sum_{b\in\mathcal{B}} \pi_{u,q'}(b)\exp\left(\gamma R^{\mathcal{B}}_{u,q'}(b)\right),$$

we obtain $\log \pi_{u,q'}(b) + \gamma R^{\mathcal{B}}_{u,q'}(b) = \log \tilde{\pi}_{u,q'}(b) + \log Z^{\mathcal{B}}_{u,q'}$, and substituting this into the branch-level PUBAO objective gives

$$\mathrm{PUBAO}^{\mathcal{B}}_{u,q}(e) = \mathrm{PESO}^{\mathcal{B}}_{u,q}(e) + \log Z^{\mathcal{B}}_{u,q'}.$$

Therefore, at both intent and interpretation levels, PUBAO and PESO differ by the corresponding log-partition term.

Candidate-dependent rankings. At either objective level $\mathcal{X} \in \{\mathcal{H}, \mathcal{B}\}$, $Z^{\mathcal{X}}_{u,q'}$ depends on the candidate enhancement q', so this relationship does not imply identical PESO and PUBAO rankings. Let $q^\star$ be the unique PESO maximizer with margin $m = \min_{q'\neq q^\star}\{\mathrm{PESO}(q^\star) - \mathrm{PESO}(q')\}$. A sufficient condition for preserving the PESO maximizer is

$$\max_{q'\neq q^\star}\left\{\log Z^{\mathcal{X}}_{u,q'} - \log Z^{\mathcal{X}}_{u,q^\star}\right\} < m.$$

Thus, PUBAO may select a different enhancement when candidate-dependent variation in the log-partition term is large enough to overcome the PESO margin.

D Experimental Details

This section provides additional implementation and evaluation details for the two main experimental settings in Sections 4 and 5.

D.1 E-commerce Experimental Details

Data split and model training. We split the e-commerce data by (session ID, q) groups so that interactions associated with the same session–query pair do not cross train, validation, and test partitions. The behavioral predictor f_y is trained on 483K interactions from 59K groups, whereas the satisfaction predictor f_z uses only the 10% subset with reference satisfaction labels z in the main setting. The main enhancement evaluation uses a group-disjoint held-out set of 4,210 interactions, while reward-model evaluation uses a separate held-out test set of 917 examples. The deployed interpretation space contains 40 prototypical personas and 6,348 interpretation branches.

Reward model and refinement. We instantiate f_y as a 400-tree multi-output Random Forest and f_z as a 600-tree regressor trained with squared-error loss on z, using 112 input features. All LLM-based enhancement components and LLM baselines use GPT-5.2. We set the reward-tilting coefficient to $\gamma = 1.5$, the faithfulness coefficient to $\lambda = 0.1$, and the serving candidate cutoff to $K = 200$. Reward-Guided FD runs for three refinement steps and the higher-scoring candidate at each step becomes the incumbent for the next iteration. Appendix H provides the prompts used for pool initialization, challenger generation, satisfaction judging, and feedback generation.

Statistical testing. All e-commerce methods share the same routing, reward-scoring, and satisfaction-evaluation pipeline. We use two-sided paired t-tests on matched evaluation instances as the primary significance test against the original-query baseline and report Welch's t-tests as a complementary analysis. The paired tests reported in Section 4.3 therefore measure within-instance changes in reference satisfaction rather than differences between independently sampled groups.

D.2 LongMemEval Experimental Details

Split and supervision. We use LongMemEval-M (Wu et al., 2024), which contains 500 questions associated with long conversational histories. Because the benchmark does not define a train/test partition for reward-model training, we construct a fixed 300/200 split stratified by question type and use the same split for all methods. Only the 300-question training partition is used to fit the benchmark-specific reward model, while the 200-question partition is held out for retrieval and downstream QA evaluation. Answer-session annotations from the training split supervise whether candidate memory sessions contain useful evidence, while neither gold answers nor held-out answer-session identifiers are exposed during test-time query generation or reranking.

Retrieval and downstream evaluation. For each question, an enhanced query retrieves candidate memory sessions using the official flat-BM25 retriever. The first-stage setting evaluates BM25 retrieval directly, while shared reranking applies `BAAI/bge-reranker-v2-m3` to the top $N = 50$ sessions and retains $k = 5$. For our method, the reward model is used only to construct q' and is not reapplied after reranking. The final five sessions are passed to a fixed GPT-5.2 reader and evaluated with a GPT-4o judge using Recall@K, NDCG@5, and QA accuracy.

Baseline implementation. RM3 and Rocchio use pseudo-relevant sessions returned by the common flat-BM25 retriever, while Key Expansion (Wu et al., 2024) performs lexical query expansion. PBR (Zhang et al., 2026) incorporates conversational user context into personalized query expansion, and Direct-rewrite uses GPT-5.2 to rewrite the input question once without reward-guided iterative feedback. We reimplement and evaluate all methods using the same fixed split, retrieval pipeline, and downstream reader to ensure a controlled comparison.

E Additional Experiments and Analyses

This section provides additional analyses of the reward-model approximation, robustness to reward-model choice, and a diagnostic evaluation of our method on BESPOKE.

E.1 Validation of the Two-Stage Reward Approximation

Our practical reward model uses the deterministic two-stage plug-in approximation described in Section 3. To examine how closely it follows the hierarchical formulation, we implement a hierarchical-masked variant in which purchase-conditioned outcomes such as review, repurchase, and refund are modeled only when they can materialize and are marginalized at inference. All downstream reward-model settings are kept unchanged. The two-stage and hierarchical-masked estimators produce nearly identical reward values, with Pearson correlation 0.998, Spearman correlation 0.996, and mean absolute error 0.049. They also preserve downstream method ordering, supporting the simpler plug-in estimator used in the experiments. This diagnostic evaluates the behavioral plug-in approximation; the conditional-mean proxy for the theoretical branch reward is described in Section 3.1.

E.2 Robustness to Reward-Model Choice

We test whether the improvements depend on the Random Forest used during candidate selection by fixing the generated queries and re-evaluating them with independently trained LightGBM, XGBoost, and non-parametric kNN estimators. For kNN, features are standardized and projected to 32 PCA components using transformations fitted only on the training split, followed by distance-weighted k-NN with $k = 100$.

Table 9: Reward gains over the original query under alternative evaluators.

Method	RF	LightGBM	XGBoost	kNN
BM25 + RM3	+0.100	+0.183	+0.107	-0.158
Rocchio-PRF	+0.091	+0.150	+0.097	-0.408
Persona-aware PRF	+0.069	+0.138	+0.042	-0.027
PUBAO-FD	**+0.231**	**+0.267**	**+0.268**	**+0.161**

Table 10: Diagnostic results on BESPOKE.

Method	NeedAlign ↑	Depth ↑	Tone ↑	Style ↑	Recall ↑
Raw query	4.00	4.20	6.76	5.72	0.230
User-profile prefix	4.56	5.08	6.80	6.04	0.286
Naive one-shot LLM rewrite	4.28	5.20	6.60	6.44	0.319
Sample-and-rerank by reward model	4.04	4.84	6.64	6.24	0.279
Single-shot PUBAO enhancement	4.36	5.12	6.56	6.08	0.315
PESO-guided enhancement	4.40	5.20	6.60	6.12	0.275
PUBAO without faithfulness ($\lambda = 0$)	4.33	4.86	6.37	5.96	0.273
PUBAO without reward tilt ($\gamma = 0$)	4.28	5.00	6.56	6.04	0.308
PUBAO with multi-turn enhancement	4.24	5.12	6.68	6.24	0.339
5-axis Reward-Guided FD	4.76	5.56	6.92	6.48	0.355

Table 9 shows that PUBAO-FD remains strongest under all four evaluators and achieves a positive gain under each. The parametric evaluators yield consistent conclusions, while kNN provides a complementary nonparametric test that avoids out-of-range extrapolation. All kNN predictions remain between 4.81 and 9.90, yet PUBAO-FD achieves a positive gain while the other baselines remain weaker. This provides a stricter check against pure out-of-range reward extrapolation because kNN predictions are convex combinations of observed neighbor labels.

E.3 Diagnostic Study on BESPOKE

Setup. We additionally evaluate on BESPOKE (Kim et al., 2025), a public personalized information-seeking benchmark based on user-specific chat and search histories. Each example contains a query, user context, explicit gold information needs, and a gold response, and we use the fixed test split of 50 user–query pairs. Because BESPOKE does not contain post-query behavioral traces, its response-level personalization and recall judgments provide proxy supervision for the reward model. We construct the interpretation space by clustering gold information items from the training split into semantic groups representing recurring personalized information needs.

Methods and evaluation. We compare raw and persona-prefixed queries, one-shot rewriting, reward-based sample-and-rerank, PESO/PUBAO variants, ablations removing faithfulness or reward tilting, multi-turn PUBAO enhancement, and five-axis Reward-Guided FD. Each enhanced query is passed to the same search-augmented answer generator, and we evaluate the resulting response using NeedAlign, Depth, Tone, Style, and Recall. NeedAlign, Depth, Tone, and Style are rescaled to 1–10, while Recall is measured in $[0, 1]$ against the gold information needs.

Results. Table 10 shows that simple personalization and one-shot rewriting improve several metrics over the raw query, whereas scalar reward-based sample-and-rerank is less consistent, suggesting that scalar selection alone does not fully exploit personalized supervision. PESO and PUBAO exhibit complementary behavior, while removing either faithfulness or reward tilting generally degrades performance, supporting both components of the objective. Five-axis Reward-Guided FD achieves the best overall result across all five metrics. This experiment is primarily diagnostic because its feedback dimensions are explicitly aligned with the evaluation axes, but it shows that decomposed reward signals can effectively guide refinement when corresponding diagnostic dimensions are available.

F DATA SAMPLES

In Section 4, training examples are constructed from de-identified user-session data and an internal user-interest profiling system. This system represents user context through category-level preference scores inferred from demographic characteristics, interests, preferences, and behavioral tendencies.

Because the original data contain company-internal schema information and user behavioral logs, we provide only a small number of de-identified examples in Tables 11, 12, and 13. Direct identifiers, including session id, user id, and item id, are replaced with pseudonymous tokens. Prices are displayed after masking the lower three digits. For each user, we expose only the top preference entries as tuples of category index, category code, masked category name, and preference score. To indicate the broad format of the taxonomy without revealing its proprietary structure, only a small number of coarse illustrative top-level category names are shown, such as Travel, Electronics, and Gen Z Leisure Culture. Fine-grained leaf category names are not exposed. Product names are de-branded and generalized with placeholder tokens such as `[Brand_A]`, preserving only broad product information needed to understand the example.

Table 11: A de-identified training sample with a review outcome.

Field	Value
pseudonymized_session	`sess_98dd32c1`
pseudonymized_user	`user_ac1ae989`
age_decade / gender	50s / F
session_date	2025-08-02
time_window	09:58 → 10:09
event_counts	impression=4, click=1, wishlist=0, cart=0, purchase=1
interest_profile_top10	[idx=638, code=S0905, preference_category=<masked>] 0.9997; [idx=423, code=P101705, preference_category=<masked>] 0.9963; [idx=418, code=P1017, preference_category=<masked>] 0.9892; [idx=436, code=P10210402, preference_category=<masked>] 0.9794; [idx=417, code=P10, preference_category=<masked>] 0.9736; [idx=553, code=P17, preference_category=<masked>] 0.9618; [idx=194, code=I2309, preference_category=<masked>] 0.9578; [idx=187, code=I23, preference_category=<masked>] 0.9560; [idx=396, code=P070102, preference_category=<masked>] 0.9442; [idx=476, code=P1022, preference_category=<masked>] 0.9433
events	`[00]` **IMPRESSION** at 09:58, query='shampoo 1+1' item_id=`item_a6402852`, price=(40,***, 13,***) item='[Brand_A] perfume shampoo, large-size hair-care product' category='Fragrance/Body/Hair > Hair-care > Shampoo' `[01]` **IMPRESSION** at 09:58, query='shampoo 1+1' item_id=`item_0b541e46`, price=(36,***, 36,***) item='[Brand_B] protein shampoo, large-size daily hair-care product' category='Fragrance/Body/Hair > Hair-care > Shampoo' `[02]` **CLICK** at 09:58, query='shampoo 1+1' item_id=`item_76fbbd4a`, price=(30,***, 11,***) item='[Brand_C] hair treatment, damaged-hair care product' category='Fragrance/Body/Hair > Hair-care > Treatment' `[03]` **IMPRESSION** at 10:00, query='shampoo 1+1' item_id=`item_0b541e46`, price=(36,***, 36,***) item='[Brand_B] protein shampoo, large-size daily hair-care product' category='Fragrance/Body/Hair > Hair-care > Shampoo' `[04]` **IMPRESSION** at 10:00, query='shampoo 1+1' item_id=`item_a6402852`, price=(40,***, 13,***) item='[Brand_A] perfume shampoo, large-size hair-care product' category='Fragrance/Body/Hair > Hair-care > Shampoo' `[05]` **PURCHASE** at 10:03, query='shampoo 1+1' item_id=`item_a6402852`, price=(40,***, 13,***) item='[Brand_A] perfume shampoo, large-size hair-care product' category='Fragrance/Body/Hair > Hair-care > Shampoo' **additional behavioral signal: review=3⋆**

Table 12: A de-identified training sample with a repurchase outcome.

Field	Value
pseudonymized_session	sess_9ea59d6a
pseudonymized_user	user_c2cf7d1c
age_decade / gender	50s / F
session_date	2026-01-16
time_window	14:27 → 14:28
event_counts	impression=6, click=1, wishlist=0, cart=0, purchase=1
interest_profile_top10	[idx=240, code=I34, preference_category=<masked>] 0.9956; [idx=620, code=S0705, preference_category=<masked>] 0.9945; [idx=486, code=P1201, preference_category=<masked>] 0.9938; [idx=615, code=S07, preference_category=<masked>] 0.9923; [idx=472, code=P102116, preference_category=<masked>] 0.9900; [idx=241, code=I3401, preference_category=<masked>] 0.9862; [idx=201, code=I2602, preference_category=<masked>] 0.9845; [idx=213, code=I261301, preference_category=<masked>] 0.9796; [idx=243, code=I340102, preference_category=<masked>] 0.9793; [idx=200, code=I26, preference_category=<masked>] 0.9750
events	[00] **IMPRESSION** at 14:27, query='lutein zeaxanthin astaxanthin' item_id=item_44feddf3, price=(135,***, 28,***) item='[Brand_D] lutein-zeaxanthin supplement, multi-box option' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [01] **IMPRESSION** at 14:27, query='lutein zeaxanthin astaxanthin' item_id=item_8c8db1d7, price=(90,***, 30,***) item='[Brand_E] eye-health supplement, capsule-type product' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [02] **IMPRESSION** at 14:27, query='lutein zeaxanthin astaxanthin' item_id=item_203edb82, price=(29,***, 29,***) item='[Brand_F] lutein-zeaxanthin supplement, monthly supply' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [03] **IMPRESSION** at 14:27, query='lutein zeaxanthin astaxanthin' item_id=item_aee645bb, price=(105,***, 29,***) item='[Brand_G] lutein-zeaxanthin supplement, multi-box option' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [04] **IMPRESSION** at 14:27, query='lutein zeaxanthin' item_id=item_b73cfbde, price=(60,***, 36,***) item='[Brand_H] lutein-zeaxanthin supplement, capsule-type product' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [05] **IMPRESSION** at 14:27, query='lutein zeaxanthin' item_id=item_44feddf3, price=(135,***, 28,***) item='[Brand_D] lutein-zeaxanthin supplement, multi-box option' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [06] **CLICK** at 14:27, query='lutein zeaxanthin' item_id=item_b73cfbde, price=(60,***, 36,***) item='[Brand_H] lutein-zeaxanthin supplement, capsule-type product' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' [07] **PURCHASE** at 14:28, query='lutein zeaxanthin' item_id=item_b73cfbde, price=(60,***, 36,***) item='[Brand_H] lutein-zeaxanthin supplement, capsule-type product' category='Health/Diet > Health-functional foods > Lutein / Eye-nutrition' **additional behavioral signal: repurchase=1**

Table 13: A de-identified training sample with review and refund signals.

Field	Value
pseudonymized_session	sess_c35ea98a
pseudonymized_user	user_56d40c78
age_decade / gender	40s / M
session_date	2026-03-29
time_window	18:26 → 18:27
event_counts	impression=8, click=2, wishlist=0, cart=0, purchase=1
interest_profile_top10	[idx=24, code=I030201, preference_category=<masked>] 0.9998; [idx=301, code=L17, preference_category=<masked>] 0.9992; [idx=304, code=L1703, preference_category=<masked>] 0.9992; [idx=23, code=I0302, preference_category=<masked>] 0.9989; [idx=620, code=S0705, preference_category=<masked>] 0.9984; [idx=554, code=P1701, preference_category=<masked>] 0.9979; [idx=517, code=P15, preference_category=<masked>] 0.9969; [idx=615, code=S07, preference_category=<masked>] 0.9968; [idx=476, code=P1022, preference_category=<masked>] 0.9960; [idx=553, code=P17, preference_category=<masked>] 0.9960
events	[00] **IMPRESSION** at 18:26, query='blueberries' item_id=item_35c42e17, price=(16,***, 8,***) item='fresh blueberries, domestic produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [01] **IMPRESSION** at 18:26, query='blueberries' item_id=item_08e6a238, price=(33,***, 33,***) item='[Brand_I] frozen blueberries, imported produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [02] **CLICK** at 18:26, query='blueberries' item_id=item_08e6a238, price=(33,***, 33,***) item='[Brand_I] frozen blueberries, imported produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [03] **IMPRESSION** at 18:26, query='blueberries' item_id=item_08e6a238, price=(33,***, 33,***) item='[Brand_I] frozen blueberries, imported produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [04] **IMPRESSION** at 18:26, query='blueberries' item_id=item_35c42e17, price=(16,***, 8,***) item='fresh blueberries, domestic produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [05] **IMPRESSION** at 18:26, query='blueberries' item_id=item_21da8fb9, price=(23,***, 20,***) item='[Brand_J] frozen fruit mix, blueberry option' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [06] **IMPRESSION** at 18:26, query='blueberries' item_id=item_5646c04f, price=(9,***, 8,***) item='[Brand_K] frozen blueberries, bulk pack' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [07] **IMPRESSION** at 18:26, query='blueberries' item_id=item_0db16d80, price=(26,***, 21,***) item='frozen blueberries, multi-bag pack' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [08] **IMPRESSION** at 18:26, query='blueberries' item_id=item_88ca733a, price=(25,***, 19,***) item='[Brand_K] frozen blueberries, bundle pack' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [09] **CLICK** at 18:26, query='blueberries' item_id=item_a7c0485d, price=(34,***, 24,***) item='ready-to-eat frozen blueberries, packed produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' [10] **PURCHASE** at 18:27, query='blueberries' item_id=item_a7c0485d, price=(34,***, 24,***) item='ready-to-eat frozen blueberries, packed produce' category='Rice/Fruit/Agri-fish > Fruit > Berry fruit > Blueberry' **additional behavioral signal: review=1⋆, refund=1**

G QUALITATIVE E-COMMERCE EXAMPLES

The following examples illustrate how our reward-guided feedback descent procedure refines an initial pool candidate into a more specific, behaviorally rewarding, and semantically faithful query enhancement. The original Korean queries are translated into English and lightly generalized for confidentiality, while all company, brand, and product identifiers are anonymized following the protocol used for the e-commerce dataset. For each example, q denotes the original query, q' denotes the initial candidate selected from the offline pool, and q'' denotes the refined query obtained after one PUBAO-guided refinement step using objective-derived textual feedback.

Across the three examples, the initial pool candidate may introduce useful specificity but can also exhibit substantial semantic drift, as reflected by its large KL value. The PUBAO-guided refinement produces a more decisive branch distribution, generally reducing entropy while improving reward alignment and semantic faithfulness. Consequently, the refined query achieves a higher PUBAO value by jointly balancing branch plausibility, behavioral reward, and deviation from the original query.

Table 14: Qualitative examples of reward-guided query refinement on the e-commerce dataset.

Example 1: Original query: *instant rice*; anonymized product: *[Brand_A] cooked white rice, 210 g, 18-pack*

Query version	Enhanced query	Sharpening	Reward	KL	Entropy	PESO	PUBAO
Original (q)	instant rice	0.401	8.057	0.163	5.241	-5.257	6.665
Initial pool (q')	chewy instant rice for home meals	0.554	8.376	12.138	3.904	-5.118	7.368
1-step FD (q'')	chewy white instant rice for home use	0.995	9.188	11.716	0.048	-1.220	12.543

Example 2: Original query: *diet*; anonymized product: *[Brand_B] grain-enzyme dietary supplement, 2 g, 90 sachets*

Query version	Enhanced query	Sharpening	Reward	KL	Entropy	PESO	PUBAO
Original (q)	diet	0.623	8.340	0.155	3.300	-3.316	9.040
Initial pool (q')	digestive enzyme stick for bread or noodle meals	0.526	7.980	12.901	4.149	-5.439	6.400
1-step FD (q'')	dietary-fiber diet enzyme stick	0.780	8.450	4.807	1.927	-2.408	10.250

Example 3: Original query: *dumplings*; anonymized product: *[Brand_C] assorted dim sum dumplings*

Query version	Enhanced query	Sharpening	Reward	KL	Entropy	PESO	PUBAO
Original (q)	dumplings	0.650	8.650	0.130	3.062	-3.075	9.780
Initial pool (q')	easy-to-steam dim sum set for home	0.850	8.700	19.650	1.312	-3.277	9.570
1-step FD (q'')	assorted dim sum dumplings	0.993	8.970	3.451	0.062	-0.408	13.050

H PROMPT TEMPLATES

The e-commerce pipeline uses four LLM-facing prompts at different offline stages. The latent-satisfaction judge in Figure 2 assigns reference satisfaction labels using the query, user context, product information, and available behavioral outcomes. The initial personalized-query generator in Figure 1 creates the candidates used to initialize the offline enhancement pool.

During iterative refinement, the proposer in Figure 3 generates a challenger using the current winner and the accumulated feedback. After an external objective selects the winner, the critic in Figure 4 explains the preference and provides a concrete editing direction for the following iteration. These prompts are used only for offline label construction, pool initialization, and pool refinement, while serving uses the resulting precomputed pool without online LLM generation.

P1 — Initial Personalized-Query Generator

[Usage]

This prompt generates an initial personalized query from the provided product and user information before iterative refinement begins.

[Prompt]

You are a personalized query enhancer for an e-commerce search box.

Given one product and one user profile containing interests and consumption patterns, generate exactly one natural and personalized query that the user would plausibly type to find the product.

Rules

1. Keep the query within the product category and make it suitable for product search rather than reviews, recipes, usage instructions, or related content.
2. Use only preferences and behavioral signals grounded in the provided user profile.
3. Produce a concise and natural search phrase.
4. Do not copy the product name verbatim or make only a trivial modification to it.
5. Select a meaningful product attribute and combine it with a genuinely relevant user signal.
6. Express the selected product attribute and user signal through a plausible search context.
7. Do not copy internal profile labels, raw specifications, negative-review keywords, or unsupported attributes into the query.
8. Provide a brief rationale identifying the product attribute and user signal used.
9. Return only the required JSON object.

Output Format

```
{
  ''product_cluster_id'': <int>,
  ''user_cluster_id'': <int>,
  ''enhanced_query'': ''<personalized query>'',
  ''rationale'': ''<product attribute and user signal used>'',
  ''signal_type'': ''<type of user signal>''
}
```

Figure 1: **Prompt used to initialize the offline enhancement pool.** The generator combines grounded product attributes and user information to produce an initial personalized query.

P2 — LLM Judge for Latent Satisfaction

[Usage]

This prompt assigns a reference satisfaction label using query quality, user--product compatibility, and available post-interaction outcomes.

[Prompt]

You are a strict judge of latent satisfaction for an e-commerce search and recommendation system.

Estimate how likely the user is to be satisfied when the enhanced query leads to the given product.

Use original query, enhanced query, user profile, product information, and any available reviews, ratings, repurchase, or refund information.

Query Quality

Evaluate whether the enhanced query preserves the original intent, uses grounded user information, improves specificity, remains plausible, and matches the product.

Unsupported personalization or substantial drift from the original intent should lower the quality score.

Realized Outcome

Evaluate the evidence of user satisfaction from the available behavioral outcomes.

Positive reviews and repeated purchases support a high score, whereas negative reviews or customer-initiated refunds support a low score.

Overall Satisfaction

Combine query quality and realized outcomes into the final satisfaction score.

Rely more strongly on realized outcomes when reliable, and rely more on query quality when outcome evidence is absent or uncertain.

A very low value in either query quality or realized satisfaction should constrain the final score.

Service Confound

Set service_confound to true only when a negative outcome is mainly caused by delivery, logistics, customer service, or another operational issue rather than query or product mismatch.

Output Format

```
{
  ''rubric_reasoning'': ''<query-quality assessment>'',
  ''rubric_scores'': {
    ''intent_product_preservation'': <int>,
    ''personalization_grounding'': <int>,
    ''hooking_attractiveness'': <int>,
    ''specificity_gain_retrieval'': <int>,
    ''plausibility_non_clickbait'': <int>,
    ''user_product_fit'': <int>
  },
  ''grounding_violation'': <bool>,
  ''quality_score'': <int>,
  ''outcome_reasoning'': ''<outcome assessment>'',
  ''outcome_score'': <int>,
  ''fusion_reasoning'': ''<overall assessment>'',
  ''meta_overall'': <int>,
  ''service_confound_reasoning'': ''<confound assessment>'',
  ''service_confound'': <bool>
}
```

Figure 2: **LLM-judge prompt used to obtain latent-satisfaction.** The judge combines query quality, user–product compatibility, and available behavioral outcomes into the final satisfaction score.

P3 — Feedback-Descent Proposer

[Usage]

This prompt generates a challenger query using the current winner, the product and user context, and feedback from the previous comparison.

[Prompt]

You are a personalized query enhancer for an e-commerce search box.

Generate one challenger query by improving the current winner according to the provided feedback.

Use the same product-grounding and personalization rules applied when generating the initial query.

Preserve the current winner's useful intent and grounded personalized attributes.

Apply only the minimal change described by the editing instruction.

Do not repeat weaknesses identified in the previous losing query.

Feedback must not override the product category, product information, user profile, or original-query intent.

If the feedback conflicts with grounded information, ignore the conflicting part and make a conservative edit.

Do not copy product names, brand names, model names, or raw specifications unless they are necessary to preserve the original intent.

Return only the required JSON object.

Output Format

```
{
  ''product_cluster_id'': <int>,
  ''user_cluster_id'': <int>,
  ''enhanced_query'': ''<challenger query>'',
  ''rationale'': ''<rationale for the edit>'',
  ''signal_type'': ''<type of user signal>''
}
```

Figure 3: **Proposer prompt used during reward-guided feedback descent.** The proposer preserves the current winner's successful properties while applying the supplied editing direction.

P4 — Feedback-Descent Critic

[Usage]

This prompt explains a winner selected by the external objective and provides an editing direction for generating the next challenger.

[Prompt]

You are a feedback generator for personalized query enhancement.

You receive the original query, user profile, product information, two candidate queries, the selected winner, and diagnostic values describing their differences.

Do not independently select or reverse the winner because the preference has already been determined.

Do not generate a new query.

Explain why the winner is better using the most decisive diagnostic differences.

Then provide one concrete editing direction that preserves the winner's strengths while improving its weakest aspect.

The editing direction may recommend clearer intent, less unsupported drift, stronger use of a grounded user signal, or better alignment with the product branch.

Do not recommend copying product names, brand names, model names, raw specifications, or internal feature labels.

Return only the required JSON object.

Output Format

```
{
  ''winner query'':  ''<exact winner query>'',
  ''loser query'':  ''<exact loser query>'',
  ''feedback'':  ''<two or three sentences explaining why the winner is better>'',
  ''edit_instruction'':  ''<one editing direction for the next challenger>''
}
```

Figure 4: **Critic prompt used to synthesize objective-grounded textual feedback.** The critic explains an externally determined preference and provides a directional edit without generating the next query.