COLD-START PERSONALIZATION VIA TRAINING-FREE PRIORS FROM STRUCTURED WORLD MODELS

Avinandan Bose[1,2,*], **Shuyue Stella Li**[1,2,*], **Faeze Brahman**[3], **Pang Wei Koh**[2,3], **Simon Shaolei Du**[2], **Yulia Tsvetkov**[2,†], **Maryam Fazel**[2,†], **Lin Xiao**[1,†], **Asli Celikyilmaz**[1,†]
[1]Meta, [2]University of Washington, [3]Allen Institute for AI
[*]Co-first author, [†]Equal Advising
{avibose,stelli}@cs.washington.edu

ABSTRACT

Cold-start personalization requires inferring user preferences through interaction when no user-specific historical data is available. The core challenge is a routing problem: each task admits dozens of preference dimensions, yet individual users care about only a few, and which ones matter depends on who is asking. With a limited question budget, asking without structure will miss the dimensions that matter. Reinforcement learning is the natural formulation, but in multi-turn settings its terminal reward fails to exploit the factored, per-criterion structure of preference data, and in practice learned policies collapse to static question sequences that ignore user responses. We propose decomposing cold-start elicitation into offline structure learning and online Bayesian inference. PEP (**P**reference **E**licitation with **P**riors) learns a structured world model of preference correlations offline from complete profiles, then performs training-free Bayesian inference online to select informative questions and predict complete preference profiles, including dimensions never asked about. The framework is modular across downstream solvers and requires only simple belief models. Across medical, mathematical, social, and commonsense reasoning, PEP achieves 80.8% alignment between generated responses and users' stated preferences versus 68.5% for RL, with 3–5$\times$ fewer interactions. When two users give different answers to the same question, PEP changes its follow-up 39–62% of the time versus 0–28% for RL. It does so with $\sim$10K parameters versus 8B for RL, showing that the bottleneck in cold-start elicitation is the capability to exploit the factored structure of preference data.

1 INTRODUCTION

Cold-start personalization requires inferring user preferences through interaction when no user-specific historical data is available. Consider two users who ask an AI assistant, "I have a headache, what should I do?" A pregnant woman must *avoid* ibuprofen (contraindicated in pregnancy); a marathon runner *wants exactly that* as fast-acting relief before tomorrow's race. The optimal response for one is a safety-critical failure for the other. The pregnant user cannot even articulate this need, because she doesn't know ibuprofen harms her fetus. The assistant must elicit it. Cold-start is not a first-session problem; it recurs whenever a task's preference-relevant dimensions fall outside the span of prior interactions, even for users with extensive history. In a large-scale evaluation spanning medical, mathematical, and commonsense reasoning, Li et al. (2026) find that frontier models fail to ask appropriate clarifying questions even when explicitly prompted, with 29% of elicitation attempts worsening alignment versus generic responses.

Why is elicitation so difficult? A single task can involve 20–30 preference dimensions (Li et al., 2025c; 2026). For the headache query, these span medication safety, time urgency, clinical detail, emotional tone, prior treatment history, holistic versus conventional preference, and many more. But individual users care about only 2–4 of these, and *which* 2–4 depends on who is asking: the pregnant woman cares about safety and reassurance; the runner about urgency and performance. With a limited interaction budget (Radlinski et al., 2019), the assistant cannot ask about all dimensions. This is a routing problem: the assistant must navigate a large space of possible preference dimen-

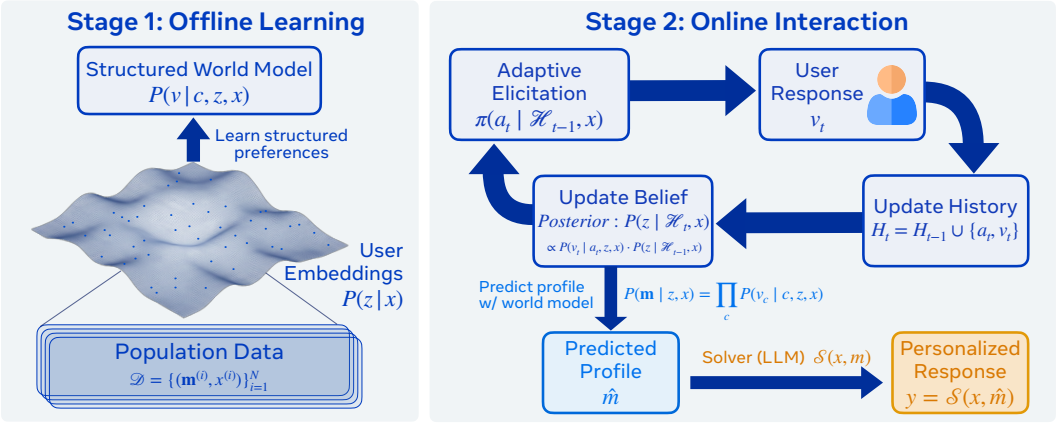

Figure 1: Overview of PEP. **Offline:** We learn a structured world model from population data capturing preference correlations through latent user embeddings. **Online:** For new users, we adaptively select questions, update beliefs after each response, and predict the full preference profile, including preferences not asked about, to generate personalized responses.

sions to find the sparse subset relevant to *this* user, within a handful of questions. Fixed or random questioning strategies will miss the relevant dimensions with high probability.

Preferences, however, are not independent across dimensions or across users. A user concerned about medication safety is likely also seeking reassurance rather than clinical depth. These correlations are robust in our data and consistent across domains (§4), and they mean that a well-chosen question can reveal far more than the single dimension it directly queries, because observing one preference updates beliefs about many others. Collaborative filtering exploits analogous structure in recommender systems, using matrix factorization and latent factor models to infer missing ratings from partial observations (Koren et al., 2009; Salakhutdinov & Mnih, 2008), with active extensions that elicit user preferences to address cold-start through strategically selected queries (Boutilier et al., 2012; Jin & Si, 2012; Golbandi et al., 2011; Elahi et al., 2018). Our setting differs in two key respects: the preference dimensions vary per task rather than forming a fixed item catalog, and the output is a free-form natural language response whose quality depends on accurately capturing the user's full preference profile rather than ranking a set of products.

With benchmarks like PrefDisco (Li et al., 2026) providing complete preference profiles and automated evaluation validated against human judgments, preference elicitation becomes a natural multi-turn reinforcement learning (RL) problem: the agent learns a questioning policy through interaction, receiving reward based on how well the final personalized response aligns with the user's preferences (Wang et al., 2025; Wei et al., 2025; Wu et al., 2025a). But RL's feedback structure is mismatched to this problem. The reward is sparse and terminal: only the final response is evaluated after T questions, providing a single scalar that does not decompose across turns. To determine which questions were informative, the agent must solve credit assignment across all T sequential decisions (Kaelbling et al., 1998; Ross et al., 2008). In practice, this causes RL to converge to static, generic question sequences: on AIME, the trained RL policy asks identical questions to every user regardless of their responses, achieving 0% adaptivity. Critically, the same datasets that provide RL's terminal rewards contain complete preference profiles with dense, per-criterion labels. RL fails because its training signal collapses factored supervision into a single scalar.

We propose PEP (**P**reference **E**licitation with **P**riors), which decomposes cold-start elicitation into offline structure learning and online Bayesian inference, sidestepping credit assignment entirely (Figure 1). Offline, we learn a structured world model from complete preference profiles, capturing how preference dimensions correlate through latent user embeddings. Online, for each new user, we maintain a posterior over their latent embedding and update it via Bayes' rule as responses arrive, with questions selected to maximize information gain about the user's complete profile, including preferences never directly asked about. The framework is training-free at test time and modular, allowing any black-box LLM to serve as the downstream solver. It admits sample complexity guar-

antees polynomial in the number of criteria and independent of the interaction budget T, whereas RL scales exponentially (Appendix A).

We evaluate PEP on four reasoning domains from PrefDisco (Li et al., 2026): medical (MedQA), mathematical (AIME), commonsense (CSQA), and social (SocialIQA). These tasks are unambiguous in *what* to solve but admit diverse user preferences in *how* to reason and respond, as illustrated by the headache example above. PEP elicits preferences through T=5 questions, predicts a complete preference profile $\hat{m}$ over all criteria including those never queried, and generates a personalized response (Figure 1). The response is judged against the user's ground-truth preferences via PrefAlign, a rubric-based evaluation validated against human judgments. PEP achieves 80.8% alignment between generated responses and users' stated preferences versus 68.5% for RL (GRPO), with 3–5$\times$ fewer interactions. When two users give different answers to the same question, PEP changes its follow-up question 39–62% of the time versus 0–28% for RL. Ablations confirm that the learned world model is the primary driver of these gains, with adaptive question selection saving one additional question per interaction. PEP achieves this with $\sim$10K parameters (Bayesian linear regression) versus 8B for RL baselines, demonstrating that the bottleneck in cold-start elicitation is not model capacity but whether the method exploits the factored structure of preference data.

Contributions. (1) We formalize the distinction between factored and entangled supervision for preference elicitation, explaining why end-to-end RL collapses to static questioning when trained with sparse terminal rewards. (2) We propose PEP, a modular framework that decomposes cold-start elicitation into offline world model learning with dense supervision and online Bayesian inference, adapting active collaborative filtering to LLM personalization. It requires no retraining at test time and integrates with any black-box solver. (3) We provide extensive empirical evaluation across four reasoning domains demonstrating that this decomposition yields substantially higher preference alignment, interaction efficiency, and adaptivity than end-to-end RL.

2 PROBLEM FORMULATION

We consider an interactive setting where an assistant must personalize its response to a user's task by inferring their task-specific preferences. The task x may be unambiguous (e.g., "solve this equation"), but how to reason through and respond is not: different users prefer different tradeoffs across criteria such as detail, rigor, pedagogy, or brevity. These preferences are latent: users rarely state them upfront, and the relevant criteria vary across tasks.

Preference profiles. We represent a user's preferences as a profile $\mathbf{m}^* \subseteq \{(c, v) : c \in \mathcal{C}(x), v \in \mathcal{V}(c)\}$, recording which criteria the user cares about and what values they assign. The criteria set $\mathcal{C}(x)$ is a spanning set of relevant preference dimensions for a task. Each criterion c admits a set of possible preference values $\mathcal{V}(c)$, which may represent categorical options, ordinal levels, or natural language descriptions. The set $\mathcal{V}(c)$ includes a distinguished element representing indifference. In practice, users care about only a small subset of available criteria, so $|\mathbf{m}^*| \ll |\mathcal{C}(x)|$.

Elicitation as a POMDP. The assistant interacts with the user to uncover their preferences before generating a response. We formalize this as a partially observable Markov decision process (POMDP):

- **State:** Hidden state is the user's preference profile $\mathbf{m}^*$.
- **Actions:** At each turn t, the assistant's action is to select a criterion to query: $a_t \in \mathcal{C}(x) \setminus \{a_1, \ldots, a_{t-1}\}$.
- **Observations:** The user responds with their preference $v_t \in \mathcal{V}(a_t)$ for the queried criterion, yielding observation $o_t = (a_t, v_t)$.
- **History:** The observable history after t turns is $\mathcal{H}_t = \{(a_1, v_1), \ldots, (a_t, v_t)\}$.
- **Horizon:** The assistant has a fixed budget of T questions.
- **Policy:** A policy $\pi(a \mid \mathcal{H}_t)$ maps the current history to a distribution over which criterion to ask next.

This frames preference elicitation as exploring a structured environment where user preferences are the hidden state and questions and user responses provide partial observations.

Profile prediction and response generation. After T turns, the assistant predicts a complete preference profile $\hat{\mathbf{m}}$ from the partial observations $\mathcal{H}_T$, including preferences for criteria that were never

queried. The predicted profile then conditions the downstream response $y = \mathcal{S}(x, \hat{\mathbf{m}})$, where $\mathcal{S}$ is a fixed LLM that generates a response given the task and predicted preferences.

Objective. The quality of the final assistant response is measured by its preference alignment with the user's true preferences: $\textsc{PrefAlign}(y, \mathbf{m}^*)$ (Li et al., 2026). This score is *user-conditioned*: the same response y may score highly for one user and poorly for another. This rules out one-size-fits-all approaches because a response optimized for the population mean $\bar{\mathbf{m}} = \mathbb{E}[\mathbf{m}^*]$ will fail users whose preferences deviate from the average. Furthermore, preferences are *task-conditioned*: the same user may prefer rigorous derivations for mathematics but intuitive explanations for biology. The assistant faces a cold-start problem for each new task.

Our goal is to learn a policy that maximizes expected preference alignment across a distribution of users and tasks:

$$\max_{\pi} \; \mathbb{E}_{x \sim \mathcal{D},\, \mathbf{m}^* \sim P(\mathbf{m}|x)} \; \mathbb{E}_{\tau \sim \pi} \left[\textsc{PrefAlign}(y, \mathbf{m}^*) \right], \tag{1}$$

where the trajectory $\tau = (\mathcal{H}_T, \hat{\mathbf{m}}, y)$ is induced by the policy's question selections, the resulting observations, the predicted profile, and the solver's response.

Learning from world models versus sparse rewards. An effective elicitation policy requires a world model: knowledge of how preference dimensions correlate across users. Reinforcement learning learns this world model implicitly while learning the questioning policy, using only sparse terminal rewards. The following result shows this joint learning problem has exponential sample complexity:

Proposition 1 (Abbreviated)**.** *Learning an effective questioning policy via RL requires exploring the space of question sequences, which grows combinatorially in $|\mathcal{C}(x)|$ (# of criteria). With only terminal feedback, sample complexity scales exponentially in the budget T. In contrast, learning preference correlations from complete profiles requires samples polynomial in $|\mathcal{C}(x)|$ and independent of T.*

Appendix A provides a formalized version of this argument under standard assumptions. This exponential scaling arises because RL must solve credit assignment across sequential decisions without explicit access to correlation structures.

3 Methodology

We introduce PEP, which is a framework that decomposes cold-start preference elicitation into two components: (i) a *belief model* that captures population-level preference structure from complete profiles (Figure 1 left), and (ii) a *selection strategy* that uses this structure to adaptively query users under an interaction budget (Figure 1 right). PEP is training-free in that it performs no parameter updates for new users; adaptation occurs solely through Bayesian updates using a fixed, offline-learned belief model and can be plug-and-play with any solver mode for response generation.

World model. We model statistical regularities and correlations in user preferences through a task-conditioned population distribution $P(\mathbf{m} \mid x)$. By learning this distribution from complete profiles offline, we can construct a prior $P(v|c, x)$ for each criterion that encodes population-level correlations. At test time, partial observations allow inference over unobserved criteria via this learned structure.

3.1 Belief Models

The belief model maintains a posterior over user preferences given interaction history. Recall from Section 2 that $\mathcal{H}_t = \{(a_1, v_1), \ldots, (a_t, v_t)\}$ denotes the observed history after t interactions, where $a_i \in \mathcal{C}(x)$ is the criterion queried at turn i and $v_i \in \mathcal{V}(a_i)$ is the user's response. For each unqueried criterion $c \in \mathcal{C}(x) \setminus \{a_1, \ldots, a_t\}$, the belief model outputs

$$b_t(v \mid c) := P(v \mid c, x, \mathcal{H}_t), \quad v \in \mathcal{V}(c). \tag{2}$$

This conditional distribution captures both the predicted value and the uncertainty in that prediction.

Latent variable formulation. We assume a latent variable z (user embedding) mediates dependencies between criteria, yielding

$$P(\mathbf{m} \mid z, x) = \prod_c P(v_c \mid c, z, x). \tag{3}$$

This structure provides three benefits: (i) compact representation of the joint preference distribution without modeling exponentially many configurations; (ii) a coherent belief state where the posterior $P(z \mid \mathcal{H}_t, x)$ summarizes everything about the user after t observations; (iii) well-defined information gain—querying any criterion informs z, which propagates to all others. When the posterior is tractable, information gain is computed exactly; otherwise, it can be estimated via sampling or variational methods.

Offline learning. Given a dataset $\mathcal{D} = \{(\mathbf{m}^{(i)}, x^{(i)})\}_{i=1}^N$ of complete preference profiles, we learn task-conditioned priors

$$b_0(v \mid c) := P(v \mid c, x) \quad \text{for } v \in \mathcal{V}(c). \tag{4}$$

These encode population-level preference correlations before any user-specific interaction.

Bayesian updates. After observing (a_t, v_t), the posterior over the latent variable z updates via Bayes' rule:

$$P(z \mid \mathcal{H}_t, x) \propto P(v_t \mid a_t, z, x) \cdot P(z \mid \mathcal{H}_{t-1}, x), \tag{5}$$

which induces updated predictive distributions $b_t(v \mid c)$ for all unobserved criteria. This Bayesian updating allows the model to maintain beliefs over individual users while leveraging the world model learned from population data.

Instantiations. The framework admits any belief model that maintains a posterior over z. The input representation $\boldsymbol{\phi}(\mathcal{H}_t)$ can range from simple indicator encodings to neural networks, while the output head maintains a structured form (e.g., Gaussian, mixture) for tractable information gain. This separation allows expressive representations while preserving principled uncertainty quantification. We use two collaborative-filtering instantiations.

Example 1: Bayesian Linear Regression. The embedding $z = \boldsymbol{\phi}(\mathcal{H}_t)$ encodes observed criteria and values. Each criterion is modeled as

$$v_c = \boldsymbol{\beta}_c^\top z + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2), \tag{6}$$

with the weights $\boldsymbol{\beta}_c$ learned offline. Predictions integrate over the learned posterior: $b_t(v \mid c) = \int P(v \mid \boldsymbol{\beta}_c, z) P(\boldsymbol{\beta}_c \mid \mathcal{D})\, d\boldsymbol{\beta}_c$.

Example 2: Gaussian Mixture Model. The latent variable $z \in \{1, \ldots, K\}$ indexes user types with task-conditioned priors $P(z \mid x)$ and type-specific likelihoods $P(v \mid c, z, x)$. The posterior is

$$P(z \mid \mathcal{H}_t, x) \propto P(z \mid x) \prod_{(a_i, v_i) \in \mathcal{H}_t} P(v_i \mid a_i, z, x). \tag{7}$$

Predictions marginalize over user types: $b_t(v \mid c) = \sum_z P(v \mid c, z, x) P(z \mid \mathcal{H}_t, x)$.

Both models admit closed-form posteriors; Appendix B.1 presents derivations and discusses methods for approximating posteriors for models that do not have tractable forms.

3.2 Adaptive Question Selection

The selection strategy chooses which criterion to query next given the current posterior. At turn t, the system must select which criterion to query next: $a_t \in \mathcal{C}(x) \setminus \{a_1, \ldots, a_{t-1}\}$.

Strategies. Any function that scores candidate criteria given the current posterior $P(z|\mathcal{H}_{t-1}, x)$ can be used. Strategies differ in whether they are are adaptive (i.e., conditioning on user responses) through this posterior. We consider three examples.

Example 1: Random Selection. A criterion is chosen uniformly among those remaining:

$$a_t \sim \text{Uniform}(\mathcal{C}(x) \setminus \{a_1, \ldots, a_{t-1}\}). \tag{8}$$

This is a non-adaptive strategy that ignores H_{t-1}, though the belief model still updates for the final prediction of $\hat{m}$. This isolates the contribution of adaptive selection from belief model inference.

Algorithm 1 PEP

Require: Task x, budget T, belief model conditional on user latent z, strategy SELECT, solver $\mathcal{S}$
Ensure: Personalized response y

$\mathcal{C} \leftarrow \mathcal{C}(x)$
$P(z|x) \leftarrow \text{INITIALIZEPRIOR}(x)$
$\mathcal{H}_0 \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  $a_t \leftarrow \text{SELECT}(P(z \mid \mathcal{H}_{t-1}), \mathcal{C} \setminus \{a_1, \ldots, a_{t-1}\})$
  $v_t \leftarrow \text{QUERYUSER}(a_t)$
  $\mathcal{H}_t \leftarrow \mathcal{H}_{t-1} \cup \{(a_t, v_t)\}$
  $P(z \mid \mathcal{H}_t, x) \leftarrow \text{UPDATEPOSTERIOR}(P(z \mid \mathcal{H}_{t-1}, x) \times P(v_t \mid a_t, z, x))$
end for
$\hat{\mathbf{m}} \leftarrow \text{PREDICTPROFILE}(P(z \mid \mathcal{H}_T))$
$y \leftarrow \mathcal{S}(x, \hat{\mathbf{m}})$
RETURN y

Example 2: Uncertainty Sampling. The criterion with highest marginal entropy ($\mathbb{H}[\cdot]$) is queried:

$$a_t = \arg \max_{c \in \mathcal{C}(x) \setminus \{a_1, \ldots, a_{t-1}\}} \mathbb{H}[b_t(\cdot \mid c)]. \tag{9}$$

Since $b_t(v|c) = P(v|c, x, H_t)$ depends on the observed history through the posterior over z, this strategy is adaptive to user responses. This reduces uncertainty where it is highest but ignores correlations.

Example 3: Information Gain. The criterion that maximally reduces uncertainty about the latent variable is queried:

$$a_t = \arg \max_{c \in \mathcal{C}(x) \setminus \{a_1, \ldots, a_{t-1}\}} I(v_c; z \mid \mathcal{H}_t, x), \text{ where} \tag{10}$$

$$I(v_c; z \mid \mathcal{H}_t, x) = \mathbb{H}[z \mid \mathcal{H}_t, x] - \mathbb{E}_{v \sim b_t(\cdot|c)} \mathbb{H}[z \mid \mathcal{H}_t, (c, v), x] \tag{11}$$

is the mutual information between the response to criterion c and the latent z. Rather than scoring criteria by individual uncertainty, this adaptive strategy scores by informativeness about the latent structure connecting all preferences. Stochastic variants of $\arg\max$ are also possible (Bose et al., 2024a).

Entropy and information gain are computed in closed form with full derivations in Appendix B.2, along with bounds and approximation methods for when posteriors are not tractable. The $\arg\max$ is computed by enumeration over remaining criteria, which is tractable since $|\mathcal{C}(x)|$ is typically small (10-20). The quantities for entropy and information-gain can be computed independently for each criteria in parallel.

Elicitation procedure. Algorithm 1 summarizes PEP. The policy $\pi(a \mid \mathcal{H}_t, x)$ is implicitly defined by combining the belief update with the selection strategy.

Solver. The solver $\mathcal{S}$ is treated as a fixed black box that generates a response given task x and predicted preference profile $\hat{\mathbf{m}}$. This modular design separates elicitation from response generation, allowing PEP to focus on efficient preference inference.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We design experiments to test our central hypothesis: *effective preference elicitation requires learning population-level preference structure separately from adaptive individual inference.* Our experiments address three questions: (1) Does PEP achieve better preference alignment? (2) Why does it outperform? (3) Which components are essential?

Datasets. We evaluate on four benchmarks covering diverse reasoning domains from PrefDisco (Li et al., 2026): MedQA (medical reasoning), AIME (mathematical problem-solving), CommonsenseQA (commonsense reasoning), and SocialIQA (social reasoning). Each domain contains 100

problems, with 50 users per problem. We filter the benchmark to ensure a challenging, long-tail distribution of preferences. We remove common cross-sample criteria (present in $\geq 10\%$ of tasks) to focus strictly on task-specific cold-start discovery, and exclude extremely rare criteria (≤ 3 users) to ensure statistical validity. This results in 20–30 diverse criteria per task. On average, users care about only 2–4 criteria. This sparsity creates a sparse discovery problem: since $\mathcal{C}(x)$ is a superset of user needs, random querying is ineffective, requiring our model to leverage learned correlations to prune the search space. We use an 80/20 train/test split at the problem level, so evaluation is performed on tasks never seen during training.

All evaluation settings and protocols, including criteria realism, user simulation and judge reliability, strictly follow the PrefDisco benchmark (Li et al., 2026), which provides extensive human evaluation confirming reliability. We adopts these protocols to ensure comparability and reproducibility.

User simulation. We evaluate under a passive user protocol: users answer minimally without volunteering information. This reflects documented human-AI interaction patterns where users provide minimal information unless prompted (Sundar, 2020) and isolates the method's information-seeking capability rather than relying on user initiative. We explore other user types in Appendix D.1 and find higher risks of reward hacking with collaborative users.

Protocol and information access. All methods follow the same protocol: sequentially select $T = 5$ criteria to ask about, observe user responses, and output a predicted preference profile over all criteria. This predicted profile is then passed to a shared solver (GPT-4.1) to generate a personalized response; the solver is fixed across all methods, isolating the effect of preference elicitation (see Appendix D.3 for solver prompt). All methods select from the same set of available criteria and observe the same user responses during elicitation.

Baselines. We compare against state-of-the-art methods for interactive preference elicitation spanning the spectrum of supervision and adaptivity:

1. **Prompting**: Llama-3.1-8B-Instruct given criteria descriptions and preference level descriptions instructed to ask informative questions. Uses the same base model and system prompt as GRPO for fair comparison.

2. **CollabLLM**: Llama-3.1-8B-Instruct trained with offline-DPO using multiturn reward (Wu et al., 2025b), targeting general interactivity rather than preference elicitation.

3. **Population Average**: Uses population-mean preferences without elicitation, no interaction needed.

4. **GRPO**: Llama-3.1-8B-Instruct given criteria descriptions and preference level descriptions trained with GRPO (Shao et al., 2024) using terminal PrefAlign rewards on personalized response generated using solver. Best checkpoint selected on validation set.

5. **PEP**: Bayesian Linear Regression belief model with adaptive acquisition (Information Gain or Uncertainty variants, selected per dataset on validation). Evaluated over 20 trials.

These methods differ primarily in supervision structure. **Population Average**, **PEP**, and **GRPO** all observe complete user preference profiles, receiving equivalent supervision quantity but different in organization. Population Average and PEP receives these as factored per-criterion labels, enabling direct estimation of preference structures; while GRPO observes them through interaction trajectories where user responses reveal the same underlying preferences, learning from which requires jointly discovering preference structure and questioning policy from terminal rewards. **CollabLLM** trains on trajectories without preference labels; **Prompting** receives no task-specific supervision. LLM-based baselines (Prompting, CollabLLM, GRPO) additionally have access to natural language descriptions of each criterion during elicitation. Our experiments test a focused hypothesis: whether separating preference structure learning from online policy learning yields practical benefits over end-to-end approaches. Full implementation details appear in Appendix E.

Evaluation metrics.

1. *Preference alignment*: We use PrefAlign (Li et al., 2026), a weighted alignment score computed by rubric-based LLM judge (GPT-4.1) to assess how well the response matches the user's ground-truth preferences. We normalize scores relative to Generic (no preference information, score

Table 1: Preference alignment (% of Oracle, higher is better). Generic = 0%, Oracle = 100%. Mean ± std over 20 trials.

Method	MedQA	AIME	SocialIQA	CSQA
Prompting	22.3±2.1	29.1±2.3	31.4±2.2	18.2±1.9
CollabLLM	20.3±1.2	26.4±1.3	24.8±1.0	20.0±2.0
Pop. Average	73.2±1.4	74.3±1.8	82.1±1.2	72.4±1.3
GRPO	71.4±2.8	76.2±2.4	71.3±3.1	55.2±2.9
PEP	**77.4**±1.2	**80.1**±1.4	**87.3**±1.8	**78.2**±1.1

Table 2: Adaptivity: percent of cases where different user responses lead to different follow-up questions (higher is better) over 20 trials. PEP adapts 2× more often than baselines.

Dataset	Prompting	CollabLLM	GRPO	PEP
MedQA	17.3±2.1	28.3±4.2	21.4±3.2	**49.2**±3.1
AIME	29.1±2.4	44.3±3.1	0.0±0.0	**39.4**±4.2
SocialIQA	28.7±3.1	75.0±12.5	27.8±2.9	**61.8**±3.7
CSQA	22.4±2.3	37.0±4.8	21.6±2.8	**43.3**±3.4

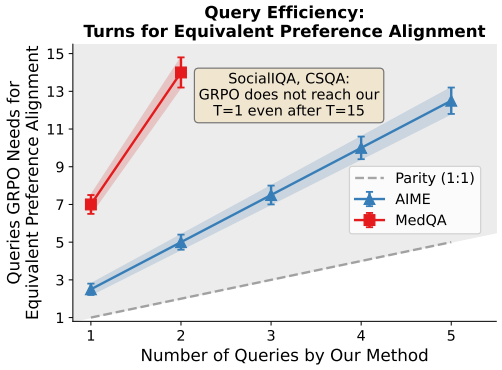

Figure 2: Query efficiency comparison between PEP and GRPO. Points above the dashed parity line indicate PEP requires fewer queries than GRPO to achieve equivalent preference alignment.

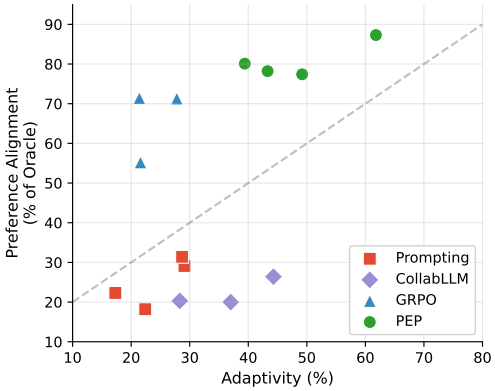

Figure 3: Adaptivity versus preference alignment across methods and datasets. Higher adaptivity is associated with better alignment.

S_{generic}) and Oracle (complete ground-truth profile, score S_{oracle}), reporting % of Oracle:

$$\% \text{ of Oracle} = \frac{S_{\text{method}} - S_{\text{generic}}}{S_{\text{oracle}} - S_{\text{generic}}} \times 100. \tag{12}$$

2. *Query efficiency*: We measure how many turns a method requires to reach a given preference alignment threshold, capturing the user interaction cost.
3. *Adaptivity*: Given identical history $\mathcal{H}_{t-1}$ and current question a_t, when two users provide different answers v_t, how often does a_{t+1} differ?

$$\text{Adaptivity} = P(a_{t+1}^{(1)} \neq a_{t+1}^{(2)} \mid \mathcal{H}_{t-1}, a_t, v_t^{(1)} \neq v_t^{(2)}). \tag{13}$$

Effective elicitation should adapt questions based on user responses; methods that ask fixed question sequences regardless of answers cannot personalize to individuals.

4.2 PEP Achieves Better Preference Alignment

Table 1 presents preference alignment across all methods and datasets. PEP consistently and significantly outperforms all baselines, achieving 77–87% of oracle performance compared to 55–76% for GRPO, 72–82% for population average, 20–26% for CollabLLM, and 18–31% for prompting. These results validate our hypothesis that separating structure learning from adaptive inference is effective. Additionally, PEP achieves these gains with approximately 10K parameters (400 criteria per domain × 25 regression weights per criterion; see Appendix E for details), demonstrating that the bottleneck in preference elicitation is inference structure, not model capacity.

4.3 Fine-Grained Performance Gain Analysis

We decompose PEP's gains into three factors: (1) inference from partial observations, (2) adaptive question selection, and (3) learned preference structure.

Table 3: **Qualitative comparison on CSQA.** Users with opposite preferences (User A: casual, engaging; User B: formal, thorough) answer questions about "Why might someone walk to work instead of driving?" Ground-truth preference shown in green (User A) and blue (User B). GRPO asks identical questions; PEP adapts to responses. Scale: 1=strongly avoid, 5=strongly prefer.

	GRPO (Fixed Sequence)			**PEP (Adaptive)**			
Turn	**Criterion**	**User A** (casual)	**User B** (formal)	**User A Criterion**	**Resp.**	**User B Criterion**	**Resp.**
Q1	Conversational Tone	5	1	Conversational Tone	5	Conversational Tone	1
Q2	Academic Citations	no pref	no pref	Humor/Wit	4	Formal Definitions	5
Q3	Technical Terminology	1	3	Real-World Examples	5	Structured Format	4
Q4	Abstract Principles	no pref	2	Storytelling	3	Comprehensive Coverage	5
Q5	Theoretical Framework	no pref	no pref	Analogies	no pref	Precise Language	3
Preferences Discovered		**1/3**	**0/3**	**3/3**		**3/3**	

4.3.1 QUERY EFFICIENCY

To analyze query efficiency, we compare how many questions each method requires to reach equivalent preference alignment (Figure 2). We focus on comparison against GRPO as the strongest RL baseline for clarity. PEP achieves 2.5× efficiency on AIME, 7× on MedQA, and >15× on SocialIQA and CSQA, where GRPO fails to match PEP's single-question performance even after 15 turns. By learning preference correlations offline from dense supervision, PEP infers unobserved preferences from partial observations. GRPO must simultaneously learn correlations and questioning policy from sparse terminal rewards, requiring direct elicitation of each preference it uses.

4.3.2 ADAPTIVE QUESTION SELECTION

Effective elicitors should adapt to individual users rather than following a fixed sequence. We measure adaptivity: how often different user responses lead to different next questions (Eq. 13). Population Average is excluded as it asks no questions. Table 2 shows PEP adapts 2× more than baselines (39–62% vs 0–44%). GRPO tends to exploit majority patterns and converge to static generic questioning. A fixed sequence of questions were generated to all users, resulting in an adaptivity score of 0% on AIME. GRPO's adaptivity correlates with cross-sample criteria overlap: on MedQA and AIME (5–8% overlap), it achieves 21–28% adaptivity and 71–76% alignment. On SocialIQA and CSQA (3% overlap), adaptivity remains low (22–28%) but performance drops to 55–71%, falling below population average. High overlap enables learning fixed question sequences for common patterns; low overlap yields sparse, diverse preferences where memorized sequences fail.

CollabLLM achieves 28–44% adaptivity by generating free-form follow-up questions. However, it targets general interactivity rather than preference elicitation, cycling through generic topics ("What is your background?" then "How do you prefer to learn?") rather than querying specific criteria informed by prior responses. This yields high adaptivity but only 20–26% preference alignment: changing questions is not sufficient; an effective elicitor must change questions *in a way that leverages learned preference structure* to target criteria the user actually cares about.

PEP maintains 39–62% adaptivity across domains by learning correlations through latent variables rather than memorizing task-specific patterns. Figure 3 shows high adaptivity generally leads to better personalization (i.e., high preference alignment scores), with GRPO-AIME as an expected outlier with low adaptivity but high alignment given the high cross-sample preference overlap.

4.3.3 QUALITATIVE COMPARISON

Table 3 compares GRPO and PEP on two users with opposite preferences. GRPO asks identical questions regardless of responses, discovering 1/3 ground-truth preferences for each user. PEP adapts: when User B rejects conversational tone, the world model infers formal preferences and pivots accordingly, discovering 3/3 preferences for both users.

4.4 PEP COMPONENTS ABLATIONS

We ablate PEP's key components to validate our two-stage design: (1) offline world model learning and (2) online adaptive inference (Figure 4). In the first **offline world model learning** stage, removing the latent structure that captures correlations between preferences causes performance to plateau

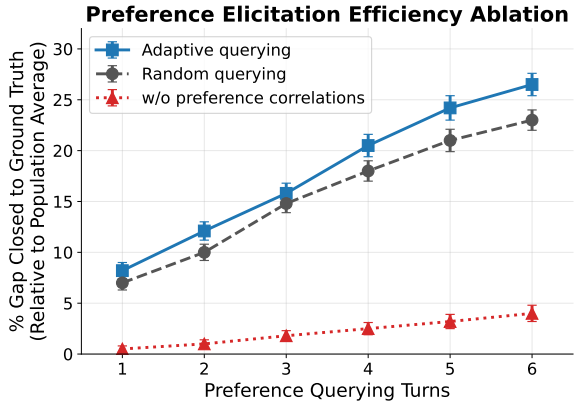

Figure 4: Ablation study on (i) modeling preference correlations: without the latent structure that captures correlations between preferences (red), elicitation yields very slow improvement over population average, compared to modeling preference correlations (blue, gray); and (ii) adaptive querying: adaptive selection (blue) closes 24% of the gap at T=5, outperforming random querying (gray) at T=6, effectively saving one question.

near population average, regardless of how many questions are asked (red). With learned correlations, each observation propagates information through the world model (blue, gray), confirming that population-level structure is essential for cold-start personalization. In the second **online adaptive inference** stage, Adaptive Querying (blue) consistently outperforms Random Selection (gray), closing 24% of the gap at T=5 compared to 21% for random. Adaptive querying at T=5 matches Random at T=6, saving one question per user interactions.

5 Related Work

The cold-start problem and preference elicitation have been extensively studied in recommender systems, where collaborative filtering learns latent structure from population data to infer preferences for new users (Koren et al., 2009; Salakhutdinov & Mnih, 2008; Liang et al., 2018; Elahi et al., 2018), and active extensions select which items to query to reduce uncertainty (Boutilier et al., 2012; Jin & Si, 2012; Golbandi et al., 2011). Recent work combines collaborative filtering with LLMs for recommendation (Zheng et al., 2024; Hou et al., 2024; Liao et al., 2023; Lin et al., 2024). Our work adapts these ideas to a different setting: rather than ranking items from a fixed catalog, we elicit structured preference profiles that condition an LLM to generate personalized free-form responses, with preference dimensions that vary per task. LLM-based preference elicitation has been explored in conversational recommenders (Austin et al., 2024; Martin et al., 2024), task specification (Li et al., 2023; Handa et al., 2024), and online RLHF (Zhang et al., 2024). PrefDisco (Li et al., 2026) shows 29% of elicitation attempts worsen alignment versus generic responses. RLHF and DPO (Ouyang et al., 2022; Rafailov et al., 2023) align to aggregated preferences without interactive discovery. Recent work adapts post-hoc through per-user reward modeling (Poddar et al., 2024; Li et al., 2025c; Bose et al., 2025). See extended related work in Appendix C.

6 Conclusion

We addressed cold-start preference elicitation by decomposing the problem into offline learning of preference structure and online Bayesian inference. Our proposed method, PEP, learns a structured world model of preference correlations from complete profiles with dense, per-criterion supervision, then performs training-free inference at test time to select informative questions and predict complete preference profiles from minimal interaction. By exploiting the factored structure of preference data that end-to-end RL discards, PEP achieves substantially higher alignment with fewer interactions using simple belief models. The framework is modular, allowing the learned world model to augment any solver, including API-based or domain-specific models, without retraining. Future work should extend to natural language elicitation, where questions and responses are free-form rather than structured over predefined criteria, and to automatic discovery of preference dimensions from data.

LIMITATIONS

While our framework demonstrates significant gains in cold-start elicitation, several avenues for future research remain. We do not address longitudinal learning across sessions or helping users form preferences they have not yet considered.

We evaluate with simulated users following the validated protocols of PrefDisco (Li et al., 2026), which were designed to reflect empirically observed human–AI interaction patterns and have been validated against human judgments. Our evaluation therefore isolates the elicitation strategy itself rather than confounding effects from user verbosity or cooperation. Nonetheless, real-world deployment would introduce additional challenges such as noisy, inconsistent, or evolving preferences.

As the world model is trained on population-level preference data, it is crucial to ensure that the learned correlations do not encode or propagate social biases. Furthermore, while the inference happens locally at test-time to preserve user privacy, there might still exist privacy and security risks from the limited turns of interaction with the model. Real world applications must be augmented with rigorous de-identification and consent protocols.

IMPACT STATEMENT

This paper presents work whose goal is to advance personalized AI systems. Effective preference elicitation enables AI assistants to adapt to individual needs, improving user satisfaction and task outcomes across diverse populations.

Potential positive impacts include more equitable access to personalized assistance, as the method requires minimal user effort, no interaction history, and no technical expertise. The training-free design enables deployment without collecting per-user training data, enhancing privacy.

Potential risks include over-reliance on inferred preferences when users' actual needs differ, and bias amplification from population training data. Systems should provide mechanisms for users to correct inferred preferences and transparently disclose when responses are personalized based on inference rather than explicit statements.

REPRODUCIBILITY STATEMENT

All experiments in this work use the open-source PrefDisco dataset. The full set of user simulation prompts is provided in Appendix D, and all hyperparameters and training configurations are detailed in Appendix E.

ACKNOWLEDGMENT

This research was developed in part with funding from the Defense Advanced Research Projects Agency's (DARPA) SciFy program (Agreement No. HR00112520300). The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. The work of MF was supported in part by awards NSF CCF 2212261, NSF CCF 2312775, NSF TRIPODS II DMS-2023166, the Meta AIM program, and the Moorthy Family Professorship at UW.

REFERENCES

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

Saleh Afzoon, Ruben Lacroix Nouri, Omri Leshem, Yarden Kalver, Ella Chan, and Yoram Bachrach. PersoBench: Benchmarking personalization in language models. *arXiv preprint arXiv:2410.14171*, 2024.

Hassan Ashtiani, Shai Ben-David, Nicholas JA Harvey, Christopher Liaw, Abbas Mehrabian, and Yaniv Plan. Near-optimal sample complexity bounds for robust learning of gaussian mixtures via compression schemes. *Journal of the ACM (JACM)*, 67(6):1–42, 2020.

Karl Johan Åström. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Raul Astudillo, Zhiyuan Jerry Lin, Eytan Bakshy, and Peter Frazier. qeubo: A decision-theoretic acquisition function for preferential bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pp. 1093–1114. PMLR, 2023.

David Austin, Anton Korikov, Armin Toroghi, and Scott Sanner. Bayesian optimization with llm-based acquisition functions for natural language preference elicitation. In *18th ACM Conference on Recommender Systems*, RecSys '24, pp. 74–83. ACM, October 2024. doi: 10.1145/3640457.3688142. URL http://dx.doi.org/10.1145/3640457.3688142.

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International conference on machine learning*, pp. 263–272. PMLR, 2017.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

David Barber and Felix Agakov. The im algorithm: a variational approach to information maximization. *Advances in neural information processing systems*, 16(320):201, 2004.

Freddie Bickford Smith, Andreas Kirsch, Sebastian Farquhar, Yarin Gal, Adam Foster, and Tom Rainforth. Prediction-oriented bayesian active learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 7331–7348. PMLR, 2023.

Avinandan Bose, Mihaela Curmei, Daniel Jiang, Jamie H Morgenstern, Sarah Dean, Lillian Ratliff, and Maryam Fazel. Initializing services in interactive ml systems for diverse users. *Advances in Neural Information Processing Systems*, 37:57701–57732, 2024a.

Avinandan Bose, Zhihan Xiong, Aadirupa Saha, Simon Shaolei Du, and Maryam Fazel. Hybrid preference optimization for alignment: Provably faster convergence rates by combining offline preferences with online exploration. *arXiv preprint arXiv:2412.10616*, 2024b.

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

Craig Boutilier, Richard S Zemel, and Benjamin Marlin. Active collaborative filtering. *arXiv preprint arXiv:1212.2442*, 2012.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *40th annual symposium on foundations of computer science (Cat. No. 99CB37039)*, pp. 634–644. IEEE, 1999.

Omar Darwiche Domingues, Pierre Ménard, Emilie Kaufmann, and Michal Valko. Episodic reinforcement learning in finite mdps: Minimax lower bounds revisited. In *Algorithmic Learning Theory*, pp. 578–598. PMLR, 2021.

Mehdi Elahi, Matthias Braunhofer, Tural Gurbanov, and Francesco Ricci. User preference elicitation, rating sparsity and cold start., 2018.

Adam Foster, Desi R Ivanova, Ilyas Malik, and Tom Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design. In *International Conference on Machine Learning*, pp. 3384–3395. PMLR, 2021.

Nadav Golbandi, Yehuda Koren, and Ronny Lempel. Adaptive bootstrapping of recommender systems using decision trees. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pp. 595–604, 2011.

David Ha and Jürgen Schmidhuber. World models. 2018. doi: 10.5281/ZENODO.1207631. URL https://zenodo.org/record/1207631.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Kunal Handa, Yarin Gal, Ellie Pavlick, Noah Goodman, Jacob Andreas, Alex Tamkin, and Belinda Z Li. Bayesian preference elicitation with language models. *arXiv preprint arXiv:2403.05534*, 2024.

Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, pp. 720–730, 2023.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Daniel Hsu, Sham M Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on learning theory*, pp. 9–1. JMLR Workshop and Conference Proceedings, 2012.

David Z Huang, Francisco Javier Marmolejo-Cossio, Tetiana Lechner, and David C Parkes. Accelerated preference elicitation with LLM-based proxies. *arXiv preprint arXiv:2501.14625*, 2025.

Felix Huber, Sebastian Rojas Gonzalez, and Raul Astudillo. Bayesian preference elicitation for decision support in multi-objective optimization. *Journal of Multi-Criteria Decision Analysis*, 32 (3):e70019, 2025.

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

Chi Jin, Zeyuan Allen-Zhu, Sebastien Bubeck, and Michael I Jordan. Is q-learning provably efficient? *Advances in neural information processing systems*, 31, 2018.

Rong Jin and Luo Si. A bayesian approach toward active learning for collaborative filtering. *arXiv preprint arXiv:1207.4146*, 2012.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. University of London, University College London (United Kingdom), 2003.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

David D Lewis and Jason Catlett. Heterogeneous uncertainty sampling for supervised learning. In *Machine Learning Proceedings 1994*, pp. 148–156. Morgan Kaufmann, 1994.

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

Belinda Z Li, Been Kim, and Zi Wang. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *arXiv preprint arXiv:2503.22674*, 2025a.

Shuyue Stella Li, Jimin Mun, Faeze Brahman, Pedram Hosseini, Bryceton G Thomas, Jessica M Sin, Bing Ren, Jonathan S Ilgen, Yulia Tsvetkov, and Maarten Sap. Alfa: Aligning llms to ask good questions a case study in clinical reasoning. In *Second Conference on Language Modeling*, 2025b.

Shuyue Stella Li, Melanie Sclar, Hunter Lang, Ansong Ni, Jacqueline He, Puxin Xu, Andrew Cohen, Chan Young Park, Yulia Tsvetkov, and Asli Celikyilmaz. Prefpalette: Personalized preference modeling with latent attributes. In *Second Conference on Language Modeling*, 2025c.

Shuyue Stella Li, Avinandan Bose, Faeze Brahman, Simon Shaolei Du, Pang Wei Koh, Maryam Fazel, and Yulia Tsvetkov. Personalized reasoning: Just-in-time personalization and why llms fail at it. *The Fifteenth International Conference on Learning Representations*, 2026.

Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan Ilgen, Emma Pierson, Pang Wei W Koh, and Yulia Tsvetkov. Mediq: Question-asking llms and a benchmark for reliable interactive clinical reasoning. *Advances in Neural Information Processing Systems*, 37:28858–28888, 2024.

Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pp. 689–698, 2018.

Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, and Xiang Wang. Llara: Aligning large language models with sequential recommenders. *CoRR*, 2023.

Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. In *Proceedings of the ACM Web Conference 2024*, pp. 3497–3508, 2024.

Dennis V Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Xianda Martin, Yutong Yue, Jinsung Kim, et al. Asking clarifying questions for preference elicitation with large language models. *arXiv preprint arXiv:2510.12015*, 2024.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 93–102. IEEE, 2010.

Ian Osband and Benjamin Van Roy. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Rock Yuren Pang, KJ Feng, Shangbin Feng, Chu Li, Weijia Shi, Yulia Tsvetkov, Jeffrey Heer, and Katharina Reinecke. Interactive reasoning: Visualizing and controlling chain-of-thought reasoning in large language models. *arXiv preprint arXiv:2506.23678*, 2025.

Matteo Papini, Damiano Binaghi, Giuseppe Canonaco, Matteo Pirotta, and Marcello Restelli. Stochastic variance-reduced policy gradient. In *International conference on machine learning*, pp. 4026–4035. PMLR, 2018.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. *Advances in Neural Information Processing Systems*, 37:52516–52544, 2024.

Filip Radlinski, Krisztian Balog, Bill Byrne, and Karthik Krishnamoorthi. Coached conversational preference elicitation: A case study in understanding movie preferences. In Satoshi Nakamura, Milica Gasic, Ingrid Zukerman, Gabriel Skantze, Mikio Nakano, Alexandros Papangelis, Stefan Ultes, and Koichiro Yoshino (eds.), *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pp. 353–360, Stockholm, Sweden, September 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5941. URL https://aclanthology.org/W19-5941/.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Stephane Ross, Joelle Pineau, Sebastien Paquet, and Brahim Chaib-draa. Online planning algorithms for POMDPs. volume 32, pp. 663–704, 2008.

Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, pp. 880–887, 2008.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

Burr Settles. Active learning literature survey. Number 1648, 2009.

H Sebastian Seung, Manfred Opper, and Haim Sompolinsky. Query by committee. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 287–294, 1992.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

David Silver and Joel Veness. Monte-carlo planning in large POMDPs. *Advances in Neural Information Processing Systems*, 23, 2010.

S Shyam Sundar. Rise of machine agency: A framework for studying the psychology of human–ai interaction (haii). *Journal of computer-mediated communication*, 25(1):74–88, 2020.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Blaise Thomson and Steve Young. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn llm agents, 2025. URL https://arxiv.org/abs/2510.14967.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level reward design, 2025. URL https://arxiv.org/abs/2505.11821.

Jason D Williams and Steve Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech & Language*, 21(2):393–422, 2007.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. Collabllm: From passive responders to active collaborators. *arXiv preprint arXiv:2502.00640*, 2025a.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. Collabllm: From passive responders to active collaborators, 2025b. URL https://arxiv.org/abs/2502.00640.

Steve Young, Milica Gašić, Blaise Thomson, and Jason D Williams. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE*, 101(5):1160–1179, 2013.

Shenao Zhang, Donghan Yu, Hiteshi Sharma, Ziyi Yang, Shuohang Wang, Hany Hassan, and Zhaoran Wang. Self-exploring language models: Active preference elicitation for online alignment. *arXiv preprint arXiv:2405.19332*, 2024.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do llms recognize your preferences? evaluating personalized preference following in llms. *arXiv preprint arXiv:2502.09597*, 2025.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1435–1448. IEEE, 2024.

A QUERY COMPLEXITY ANALYSIS

We analyze the query complexity, the total number of user interactions required, for reinforcement learning versus belief model estimation. The key distinction is not the number of users, but the *structure of feedback* available from each interaction.

A.1 SETUP

Let $C = |\mathcal{C}(x)|$ denote the number of criteria, T the elicitation budget, K the number of latent user types, and $V = \max_c |\mathcal{V}(c)|$ the maximum number of preference values per criterion. We assume access to a simulator that can sample users from the population and respond to preference queries.

A.2 REINFORCEMENT LEARNING: SPARSE FEEDBACK

In RL, the agent learns by running episodes. Each episode consists of:

1. Sample a user with profile $\mathbf{m}^* \sim P(\mathbf{m} \mid x)$
2. Ask T questions, receiving responses $v_1, \ldots, v_T$
3. Receive terminal reward $\text{PREFALIGN}(y, \mathbf{m}^*)$

Queries per episode. Each episode requires T queries to the user.

Feedback structure. The agent receives T preference values during elicitation, but these are *inputs* to the policy, not learning signals. The only supervision is the terminal reward, a single scalar summarizing how well the final response matched the user's preferences. This reward does not decompose: it provides no indication of which questions were informative or which were wasteful.

The credit assignment problem. To learn which questioning strategies work, RL must solve a credit assignment problem: given a terminal reward, determine which of the T sequential decisions contributed positively (Sutton et al., 1998). With sparse, terminal-only feedback, this requires extensive exploration.

Exploration complexity. The number of distinct T-step questioning sequences is:

$$\frac{C!}{(C-T)!} = C \cdot (C-1) \cdot \ldots \cdot (C-T+1) = O(C^T) \tag{14}$$

For $C = 20$ and $T = 5$, this exceeds 10^6 sequences. To determine which sequences are effective for which user types, RL must explore a substantial fraction of this space.

Sample complexity lower bounds. For episodic RL with horizon T and sparse rewards, standard lower bounds show that the number of episodes required scales polynomially in the state-action space and the horizon (Azar et al., 2017; Jin et al., 2018; Domingues et al., 2021). In our setting, the effective state space (possible histories) grows as $O((C \cdot V)^T)$. Even with function approximation, policy gradient methods suffer from variance that scales with the horizon when rewards are sparse (Schulman et al., 2015; Papini et al., 2018), requiring:

$$N_{\text{episodes}} = \Omega\left(\frac{T^2}{\epsilon^2} \cdot f(C, V, T)\right) \tag{15}$$

where $f(C, V, T)$ captures the complexity of the policy class and grows with the history space.

Total RL queries. The total number of queries is:

$$Q_{\text{RL}} = N_{\text{episodes}} \times T \tag{16}$$

Due to the credit assignment problem with sparse feedback, N_{episodes} scales poorly with both the history space and the horizon T. Critically, *even with unlimited access to a user simulator*, RL cannot circumvent this: the bottleneck is not user availability but the structure of feedback.

A.3 BELIEF MODEL: DENSE FEEDBACK

Our approach learns a belief model from complete preference profiles. Data collection consists of:

1. Sample a user with profile $\mathbf{m}^* \sim P(\mathbf{m} \mid x)$
2. Query the user on all criteria they care about (or a representative subset)
3. Record the complete profile $\mathbf{m}^*$

Queries per profile. Each profile requires $O(C)$ queries in the worst case, or $O(|\mathbf{m}^*|)$ queries if we only elicit criteria the user cares about. Since $|\mathbf{m}^*| \ll C$ in practice, this is often much smaller.

Feedback structure. Each query yields direct supervision: we observe the user's preference value for that specific criterion. This is *dense* feedback, as every query produces a labeled data point. There is no credit assignment problem because we are not learning from a delayed reward signal.

Learning complexity. The belief model parameters consist of:

- For GMM: K mixture weights and $K \times C \times V$ emission probabilities
- For BLR: C weight vectors of dimension d, with prior covariance

Standard results for mixture model estimation show that N complete profiles suffice to achieve ϵ-accurate parameter estimates when (Dasgupta, 1999; Moitra & Valiant, 2010; Ashtiani et al., 2020):

$$N_{\text{profiles}} = \tilde{O}\left(\frac{K \cdot C \cdot V}{\epsilon^2}\right) \tag{17}$$

For Bayesian linear regression, posterior concentration results give (Hsu et al., 2012; Abbasi-Yadkori et al., 2011):

$$N_{\text{profiles}} = \tilde{O}\left(\frac{C \cdot d}{\epsilon^2}\right) \tag{18}$$

Total belief model queries. The total number of queries is:

$$Q_{\text{belief}} = N_{\text{profiles}} \times O(C) = \tilde{O}\left(\frac{K \cdot C^2 \cdot V}{\epsilon^2}\right) \tag{19}$$

This is *polynomial in* C and *independent of* T.

A.4 COMPARISON

	RL	**Belief Model**
Feedback per interaction	Terminal reward (1 scalar)	Preference value (per criterion)
Credit assignment	Required across T decisions	Not needed
Queries per sample	T	$O(C)$ or $O(\|\mathbf{m}^*\|)$
Number of samples	Exponential in T	Linear in K, C, V
Dependence on T	Exponential	Independent

A.5 DISCUSSION

The query complexity gap reflects a fundamental difference in problem structure, not merely a difference in data efficiency.

RL solves a harder problem. Reinforcement learning attempts to discover effective questioning strategies through trial and error, using only terminal feedback. This requires solving credit assignment: determining which decisions in a T-step sequence contributed to the outcome. With sparse rewards, this is provably difficult (Kakade, 2003; Osband & Van Roy, 2016).

Belief models exploit dense supervision. Our approach sidesteps credit assignment entirely by learning from complete preference profiles. Each criterion query yields direct supervision about that criterion's distribution in the population. The sequential decision problem at test time is then solved by Bayesian inference, not by learning from rewards.

The key insight. Both approaches ultimately require interacting with users. The difference is *what we learn from those interactions*:

- RL: "This 5-question sequence with this user yielded reward 0.7." (Sparse, entangled)
- Belief model: "This user prefers formal explanations, high detail, and worked examples." (Dense, factored)

The factored structure of preference profiles enables efficient supervised learning; the entangled structure of episode rewards necessitates difficult credit assignment.

Practical implications. In settings where complete preference profiles can be collected (e.g., through user studies or annotation), the belief model approach is preferable. The offline data collection cost, $O(C)$ queries per user, is comparable to a few RL episodes, but yields far more learning signal. At test time, the learned belief model enables efficient Bayesian inference without further training.

B Information Gain Computation

B.1 Posterior Inference

Given observations $\mathcal{H}_t = \{(a_1, v_1), \ldots, (a_t, v_t)\}$, the belief model maintains a posterior over the user embedding z. We derive the update for both instantiations.

Bayesian Linear Regression. The weights $\boldsymbol{\beta}_c$ have a Gaussian posterior learned offline: $P(\boldsymbol{\beta}_c \mid \mathcal{D}) = \mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$. At test time, the user embedding $z = \boldsymbol{\phi}(\mathcal{H}_t)$ is constructed deterministically from observations. When a new observation (a_t, v_t) arrives, z is updated by incorporating the new observation into the feature encoding. The predictive distribution for criterion c is:

$$b_t(v \mid c) = \mathcal{N}(\boldsymbol{\mu}_c^\top z,\ z^\top \boldsymbol{\Sigma}_c z + \sigma^2) \tag{20}$$

The mean depends on the current user embedding z, while the variance reflects both the uncertainty in $\boldsymbol{\beta}_c$ (learned offline) and observation noise.

Gaussian Mixture Model. The user embedding $z \in \{1, \ldots, K\}$ has a categorical posterior with weights $\pi_k := P(z = k \mid \mathcal{H}_t, x)$. When observing (c, v), the posterior updates as:

$$\pi_k^{(c,v)} = \frac{\pi_k \cdot P(v \mid c, z = k, x)}{\sum_{k'} \pi_{k'} \cdot P(v \mid c, z = k', x)} \tag{21}$$

User types consistent with the observation receive higher weight.

Extensions to intractable posteriors. For more expressive belief models where the posterior is intractable, several approximation strategies exist:

Monte Carlo estimation. Samples $z^{(1)}, \ldots, z^{(M)} \sim P(z \mid \mathcal{H}_t)$ can be drawn via MCMC or importance sampling, then used to approximate expectations.

Variational approximation. A tractable distribution $q(z)$ (e.g., Gaussian, mixture) is fit to approximate $P(z \mid \mathcal{H}_t)$ by minimizing KL divergence.

B.2 Information Gain and Selection

Given the posterior over the user embedding $P(z \mid \mathcal{H}_t)$, information gain for querying criterion c is:

$$I(v_c; z \mid \mathcal{H}_t) = \mathbb{H}[z \mid \mathcal{H}_t] - \mathbb{E}_{v \sim b_t(\cdot \mid c)} \mathbb{H}[z \mid \mathcal{H}_t, (c, v)] \tag{22}$$

This measures how much observing the user's preference for c would reduce uncertainty about the user embedding z. Computing this requires: (i) the entropy of the current posterior, (ii) the predictive distribution $b_t(v \mid c)$ over possible responses, and (iii) the entropy of the updated posterior for each possible response v.

Bayesian Linear Regression. Information gain measures how much querying criterion c reduces predictive uncertainty about other criteria. Since the user embedding $z = \boldsymbol{\phi}(\mathcal{H}_t)$ grows richer with each observation, we quantify informativeness by the expected reduction in predictive variance across unobserved criteria:

$$I(c \mid \mathcal{H}_t) = \sum_{c' \neq c} \left[\mathrm{Var}[v_{c'} \mid \mathcal{H}_t] - \mathbb{E}_{v \sim b_t(\cdot \mid c)} \mathrm{Var}[v_{c'} \mid \mathcal{H}_t, (c, v)] \right] \tag{23}$$

The predictive variance $\mathrm{Var}[v_{c'} \mid \mathcal{H}_t] = z^\top \boldsymbol{\Sigma}_{c'} z + \sigma^2$ depends on how the current user embedding z interacts with the learned weight uncertainty $\boldsymbol{\Sigma}_{c'}$.

Gaussian Mixture Model. The entropy of the categorical posterior over user types is:

$$\mathbb{H}[z \mid \mathcal{H}_t] = -\sum_{k=1}^{K} \pi_k \log \pi_k \tag{24}$$

The expected posterior entropy is:

$$\mathbb{E}_{v \sim b_t(\cdot \mid c)} \mathbb{H}[z \mid \mathcal{H}_t, (c, v)] = \sum_{v \in \mathcal{V}(c)} b_t(v \mid c) \left(-\sum_{k=1}^{K} \pi_k^{(c,v)} \log \pi_k^{(c,v)} \right) \tag{25}$$

where $b_t(v \mid c) = \sum_k \pi_k P(v \mid c, z = k, x)$ is the marginal predictive distribution. Information gain is the difference between current and expected posterior entropy. All quantities involve sums over K types and $|\mathcal{V}(c)|$ possible values, yielding $O(K \cdot |\mathcal{V}(c)|)$ per criterion.

Selection by enumeration. For both models, we compute information gain for each $c \in \mathcal{C}(x) \setminus \{a_1, \ldots, a_{t-1}\}$ and select the maximizer. Since $|\mathcal{C}(x)|$ is typically 10–25, enumeration is tractable.

Approximations for intractable models. For the intractable posteriors discussed above, information gain can be approximated using bounds. The Barber-Agakov bound (Barber & Agakov, 2004) provides:

$$I(v_c; z \mid \mathcal{H}_t) \geq \mathbb{E}_{v,z} \left[\log q(z \mid v, \mathcal{H}_t) \right] + \mathbb{H}[z \mid \mathcal{H}_t] \tag{26}$$

where $q(z \mid v, \mathcal{H}_t)$ is a learned variational posterior. This approach underlies methods such as BALD (Houlsby et al., 2011) in Bayesian active learning.

C Extended Related Work

Active learning and Bayesian design. Active learning selects informative examples efficiently (Settles, 2009; Lewis & Catlett, 1994; Seung et al., 1992). Bayesian experimental design formalizes this through expected information gain (EIG) maximization (Lindley, 1956; Chaloner & Verdinelli, 1995), with recent extensions to prediction-oriented objectives (Bickford Smith et al., 2023), neural acquisition functions (Foster et al., 2021), multi-objective optimization (Astudillo et al., 2023; Huber et al., 2025), hybrid preference learning (Bose et al., 2024b).

Preference elicitation with language models. LLM-based preference elicitation has been explored in conversational recommenders (Austin et al., 2024; Martin et al., 2024; He et al., 2023), combinatorial auctions (Huang et al., 2025), and task specification (Li et al., 2023; Handa et al., 2024). SELM (Zhang et al., 2024) addresses active exploration in online RLHF, focusing on response diversity for reward modeling. PrefDisco (Li et al., 2026) demonstrates systematic failure of frontier models at proactive elicitation—29% of attempts worsen alignment versus generic responses. Our work addresses this gap through structured world model learning with dense supervision.

Personalization and collaborative filtering. Matrix factorization captures preference correlations for cold-start inference (Koren et al., 2009). Recent work evaluates static profile consistency (Afzoon et al., 2024; Zhao et al., 2025; Jiang et al., 2025) or adapts post-hoc through per-user reward

modeling (Poddar et al., 2024; Li et al., 2025c; Bose et al., 2025). RLHF and DPO (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023) align to aggregated preferences without interactive elicitation. We extend collaborative filtering to interactive preference discovery.

POMDPs and belief-state planning. Partially observable Markov decision processes formalize sequential decision-making under hidden state (Kaelbling et al., 1998; Åström, 1965), maintaining belief distributions via Bayesian filtering. POMDP frameworks have been successfully applied to dialogue management (Young et al., 2013; Williams & Young, 2007; Thomson & Young, 2010), where uncertainty arises from speech recognition errors. Online POMDP methods (Ross et al., 2008; Silver & Veness, 2010) plan in belief space through forward search. We cast preference elicitation as belief-state planning where user preferences form the hidden state and questions provide partial observations.

World models. Model-based RL learns environment dynamics for sample-efficient planning (Ha & Schmidhuber, 2018; Hafner et al., 2023; Schrittwieser et al., 2020). These approaches learn world models and policies jointly from sparse rewards. We separate world model learning (offline, dense supervision) from policy execution (online, Bayesian inference), exploiting preference structure where complete profiles enable supervised learning of correlations.

Interactive reasoning. Prior work addresses clarifying missing task information (Li et al., 2024; 2025b;a; Radlinski et al., 2019; Pang et al., 2025) rather than latent user objectives. We address preference-dependent reasoning where users require fundamentally different solution paths based on distinct objectives.

D PROMPTS

This appendix contains the prompts used for user simulation, the questioner (prompting baseline), the solver, and the judge.

D.1 PASSIVE USER SIMULATION

The passive user answers questions minimally and does not volunteer extra information.

```
You are role-playing as a human user who needs help with a
problem.  An AI assistant is asking you questions to understand
your preferences before providing a tailored explanation.
## YOUR PERSONA
{persona_profile}
## YOUR PREFERENCES
{persona_preferences}
## CURRENT QUESTION FROM ASSISTANT
{current_question}
## INSTRUCTIONS
You are a **passive user** who answers questions minimally and does
not volunteer extra information.
1.  **Answer Preference Questions Directly**:  If the assistant
asks about a preference that matches one in YOUR PREFERENCES (e.g.,
asking about explanation depth, use of analogies, technical level),
give a short, direct answer based on your preference value.
2.  **Handle Unrelated Questions**:  If the assistant asks about
something NOT in your preferences or that seems unrelated to your
persona:  - Say "I don't have a strong preference about that"
or "I'm not sure, whatever you think is best" - Do NOT make up
preferences that aren't listed
3.  **Handle Related but Unlisted Questions**:  If the question
seems related to your background/persona but isn't explicitly
listed as a preference, give a reasonable brief answer based on
your persona's characteristics.
4.  **Stay Minimal**:  - Give short, direct answers (1-2 sentences
max) - Do not elaborate or explain your reasoning - Do not ask the
assistant questions back - Do not volunteer information that wasn't
asked
5.  **Be Consistent**:  Your answers should align with your
persona's background, expertise level, and stated preferences.
## OUTPUT FORMAT
Respond with a JSON object:
{{
"thought":  "Brief reasoning about what preference this relates to
and how to answer",
"response":  "Your short, direct answer to the assistant"
}}
```

D.2 Prompting Baseline (Questioner System Prompt)

The prompting baseline receives the following system prompt instructing it to elicit user preferences through questions.

```
You are a preference elicitation assistant.  Your job is to ask
questions to understand the user's preferences BEFORE they receive
help with their task.
PHASE 1 - ELICITATION (turns 1-{max_turns}):
- Ask ONE clear, concise question per turn
- Focus on understanding HOW they want information presented (not
WHAT the answer is)
- Make questions easy to answer by offering discrete levels (e.g.,
1-5 scale, or options like "brief/moderate/detailed")
- Example question formats:
- "On a scale of 1-5, how much detail would you like?  (1=brief
overview, 3=moderate, 5=very detailed)"
- "Would you prefer:  (1) simple everyday language, (3) some
technical terms with explanations, or (5) full technical
terminology?"
- "How important are real-world examples to you?  (1=not needed, 3=a
few would help, 5=as many as possible)"
- Do NOT attempt to solve or answer their task
- You have {max_turns} questions to ask
PHASE 2 - PREFERENCE PROFILE (after turn {max_turns}):
After your final question is answered, output a structured
preference profile summarizing what you learned.
Output format:
{{
"preferences":  {{
"Preference Name":  {{
"description":  "Brief description of what this preference means",
"value":  <1-5>,
"evidence":  "What the user said that indicates this"
}}
}}
}}
Guidelines for the profile:
- Include up to {max_preferences} preferences (focus on the most
important ones)
- Each preference should have a clear name, description, value
(1-5), and supporting evidence
- Only include preferences you have evidence for from the
conversation
- If the user said "I don't have a strong preference" for something,
do not include it
This profile will be passed to another assistant who will use it to
personalize their response to the user's task.
Begin by asking your first question.
```

After all elicitation turns, the model receives:

```
You have completed all {max_turns} elicitation turns.  Now output
your preference profile summarizing what you learned about the
user's preferences.
Output ONLY the JSON profile in the format specified in your
instructions.
```

D.3 Solver Prompt

The solver receives the predicted preference profile and generates a personalized response.

```
You are a helpful assistant providing a personalized explanation for
a user's problem.
## ELICITED USER PREFERENCES
The following information was gathered from a conversation with the
user:
{elicited_preferences}
## INSTRUCTIONS
Use ONLY the preferences explicitly mentioned above to tailor your
response.  Do not assume preferences that were not discussed.
If a preference was clearly stated, incorporate it into your
response.  If something was not discussed, use neutral defaults.
Now provide your response to the user's problem, personalized
according to what they actually told you.
```

D.4 JUDGE PROMPT (PREFALIGN)

The judge evaluates how well the generated response aligns with the user's ground-truth preferences. The judge is called once per criterion, and scores are aggregated using user-specific weights.

```
You are an expert evaluation specialist assessing how well a
response is personalized to a user's preferences.
## CRITERION TO EVALUATE
**{pref_key}**
Description:  {criterion_description}
## SCORING RUBRIC
{performance_levels}
## USER'S DESIRED LEVEL
- Preferred value:  {pref_val} (on a 1-5 scale)
- Why this level suits them:  {pref_just}
## RESPONSE TO EVALUATE
"""
{final_response}
"""
## EVALUATION INSTRUCTIONS
Your task is to assess how well the response **MATCHES** the user's
preferred level for this criterion.
Important:  You are NOT judging whether the criterion is maximized,
but whether the response hits the RIGHT LEVEL for this specific
user.  For example:
- If a user prefers "Terminology Complexity" = 2 (simple language),
a highly technical response should score LOW
- If a user prefers "Explanation Depth" = 5 (very detailed), a brief
response should score LOW
Scoring:
- **5**:  Perfectly matches the user's preferred level
- **3**:  Partially matches or inconsistently matches the preferred
level
- **1**:  Completely mismatches the user's preferred level (too high
or too low)
- **0**:  The response does not address this criterion at all
Respond in JSON format:
{{"score":  <0-5>, "justification":  "<brief explanation>"}}
```

E MODEL SELECTION AND TRAINING DETAILS

E.1 BELIEF MODEL SELECTION

We evaluate two belief models for predicting user preferences from partial observations:

- **Bayesian Linear Regression (BLR)**: Models each criterion's value as a linear function of observed preferences with a Gaussian prior on weights. The posterior remains Gaussian, enabling closed-form updates and tractable information gain computation.

  Parameter count. For each criterion c, we fit a Bayesian linear regression to predict its value from other criteria the user has expressed preferences on. The input features encode which other criteria have been observed and their values: for each other criterion, we include a binary indicator (1 if observed, 0 otherwise) and the normalized preference value (or 0 if unobserved). This yields $2(C-1)$ features per model, where C is the number of criteria per problem.

  With $C \approx 15$ criteria per problem, each regression has $\sim$28 input features and thus $\sim$29 parameters (weights plus intercept). Across all four domains, there are approximately 400 unique criteria, giving:

  $$400 \text{ criteria} \times 25 \text{ parameters/criterion} \approx 10\text{K parameters}$$

- **Gaussian Mixture Model (GMM)**: Models the population as K latent user types, each with characteristic preferences. The posterior over types updates via Bayes' rule as observations arrive.

BLR consistently outperforms GMM across all datasets and is used for all main results.

E.2 Acquisition Strategy Selection

We evaluate four acquisition strategies for selecting which criterion to query:

- **Information Gain**: Selects the criterion that maximizes expected reduction in entropy over unobserved preferences. Greedy selection of the single best criterion.
- **Uncertainty**: Selects the criterion with highest predictive entropy. Greedy selection of the most uncertain criterion.
- **InfoGain-Soft**: Stochastic variant that samples candidates weighted by their information gain scores, adding exploration.
- **Uncertainty-Soft**: Stochastic variant that samples candidates weighted by their uncertainty scores.

Table 4 reports preference alignment for each strategy. We select the best strategy per dataset based on validation performance: Uncertainty for MedQA, Uncertainty-Soft for AIME and SocialIQA, and Information Gain for CSQA.

Strategy	MedQA	AIME	SocialIQA	CSQA
Random	$4.43_{\pm 0.02}$	$4.36_{\pm 0.04}$	$4.34_{\pm 0.03}$	$4.32_{\pm 0.03}$
InfoGain	$4.44_{\pm 0.01}$	$4.34_{\pm 0.04}$	$4.34_{\pm 0.03}$	$\mathbf{4.33}_{\pm 0.02}$
InfoGain-Soft	$4.44_{\pm 0.02}$	$4.36_{\pm 0.03}$	$4.34_{\pm 0.03}$	$4.31_{\pm 0.03}$
Uncertainty	$\mathbf{4.44}_{\pm 0.01}$	$4.33_{\pm 0.03}$	$4.34_{\pm 0.03}$	$4.32_{\pm 0.02}$
Uncertainty-Soft	$4.44_{\pm 0.02}$	$\mathbf{4.37}_{\pm 0.02}$	$\mathbf{4.35}_{\pm 0.04}$	$4.32_{\pm 0.03}$

Table 4: Acquisition strategy ablation showing raw judge scores (1–5 scale) at $T = 5$. Bold indicates best per dataset. Mean ± std over 20 trials. Stochastic variants (Soft) often match or outperform greedy selection due to added exploration.

Key findings. The differences between adaptive strategies are modest (typically <0.03 on the 1–5 scale), suggesting that the belief model's inference capability is the primary driver of performance rather than the specific acquisition strategy. Stochastic variants perform well, likely because sampling from top candidates adds beneficial exploration that avoids suboptimal greedy sequences.

E.3 GRPO Training Details

We train Llama 3.1 8B using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). The model receives the full list of criteria with natural language descriptions and learns to ask preference-eliciting questions. Training uses a terminal reward equal to the PrefAlign score after the solver generates a response based on the elicited preferences.

Hyperparameters.

- **Optimizer**: AdamW with weight decay 0.01
- **Learning rate**: 1×10^{-6} with 50 warmup steps
- **Batch size**: 64 (8 prompts $\times$ 8 rollouts per prompt)
- **Mini-batch size**: 8 for gradient updates
- **Epochs**: 20
- **KL penalty coefficient**: 0.001
- **Clip range**: 0.2
- **Entropy coefficient**: 0.01
- **Max gradient norm**: 0.5
- **Precision**: bfloat16 mixed precision
- **Parallelism**: FSDP2 across 4 GPUs

Checkpoint selection. We evaluate checkpoints every 50 training steps on a held-out validation set and report results from the checkpoint with highest validation PrefAlign score. We also report mean ± std over the last 4 checkpoints to account for training variance.

E.4 Prompting Baseline Details

The prompting baseline uses Llama 3.1 8B with the following system prompt:

Prompting System Prompt

> You are a helpful assistant that asks clarifying questions to understand user preferences before providing a response. You will be given a task and a list of criteria that users may have preferences about.
> Your goal is to: 1. Ask informative questions to elicit the user's preferences 2. After gathering information, infer the user's complete preference profile 3. Use these preferences to provide a personalized response
> Available criteria and their descriptions: [CRITERIA LIST PROVIDED HERE]
> Ask one question at a time. After 5 questions, output your inferred preference profile.

The model receives the full list of criteria with natural language descriptions for each criterion and preference level. Despite this rich semantic information, prompting fails to effectively leverage preference correlations, achieving only 18–31% of oracle performance.