Efficient Generative Models Personalization via Optimal Experimental Design

Guy Schacht [1] **Mojmír Mutný** [1] **Riccardo De Santi** [1] **Ziyad Sheebaelhamd** [2] **Andreas Krause** [1]

Abstract

Preference learning from human feedback has been widely adopted to align generative models with end-users. However, human feedback is costly and time-consuming to obtain, creating demand for data-efficient query selection methods. This work presents a novel approach that leverages optimal experimental design to ask humans the most informative preference queries, which can efficiently elucidate the latent reward function modeling user preferences. To this end, we formulate the problem of preference query selection as a planning problem aimed to maximize the information that queries provide about the user's underlying latent reward model. We show that this problem has a convex optimization formulation, and introduce ED-PBRL, a statistically and computationally efficient algorithm that is supported by theoretical guarantees. We empirically showcase the proposed framework by personalizing a text-to-image generative model to user-specific styles, showing that it requires substantially fewer preference queries compared to random query selection.

1. Introduction

Generative Models & Reinforcement Learning In recent years, large-scale generative models have demonstrated tremendous success in generating high-fidelity content across various modalities (Brown et al., 2020; Rombach et al., 2022; Brooks et al., 2024). These models are sequential by nature; they append to or refine generated content incrementally. For example, Large Language Models (LLMs) generate text by sequentially adding tokens (Brown et al., 2020; Ouyang et al., 2022), and text-to-image diffusion models refine their generations over a series of steps (Dhariwal & Nichol, 2021). This sequential decision-making process naturally fits the Reinforcement Learning (RL) paradigm, where the generation process is modeled by an RL agent aiming to take the best action at each intermediate step (Ouyang et al., 2022; Deng et al., 2022). This inherent sequential structure makes RL a powerful framework for optimizing and controlling the behavior of these generative agents, a connection that has been successfully exploited for multiple purposes, such as improving generation quality (Lee et al., 2023; Xu et al., 2023), aligning models with safety constraints (Bai et al., 2022; Askell et al., 2021), or with other personal user tastes (Ouyang et al., 2022; Rafailov et al., 2023; Stiennon et al., 2020).

PBRL for Personalization Framing the generative process as an RL problem is particularly powerful for personalization, as it allows for aligning the agent's policy with a user's subjective taste. The key challenge is that this taste is difficult to formalize as a numerical reward function. Reinforcement Learning from Human Feedback (RLHF) is the standard paradigm for this, learning rewards from human-supplied demonstrations or other forms of feedback (Ziebart et al., 2008; Finn et al., 2016; Linder et al., 2022; Casper et al., 2023). Perhaps the most prominent and practical instance of RLHF is Preference-Based Reinforcement Learning (PBRL), where the latent reward model is learned from comparative feedback (e.g., a user choosing between two generated images). This feedback modality is often more intuitive for humans to provide than absolute scores or full demonstrations (Christiano et al., 2017; Sadigh et al., 2017; Biyik et al., 2019; Ouyang et al., 2022; Saha et al., 2023; Azar et al., 2024). After collecting preference feedback from the user, an estimated reward model then serves as the reward signal aligning the RL agent to the human.

PBRL Query Selection via OED The success of PBRL, however, hinges on the accuracy of this learned reward model, which in turn depends on the quality of the preference queries presented for user feedback. Collecting these user preferences is a significant practical bottleneck, as it requires a human to provide numerous labels—a process that is both time-consuming and costly (Ouyang et al., 2022; Lee

[1]ETH Zurich, Switzerland [2]Max Planck Institute for Intelligent Systems, Tübingen, Germany. Correspondence to: Guy Schacht <gshacht@ethz.ch>, Mojmír Mutný <mutny@inf.ethz.ch>, Riccardo De Santi <rdesanti@ethz.ch>, Ziyad Sheebaelhamd <ziyad.sheebaelhamd@uni-tuebingen.de>, Andreas Krause <krausea@ethz.ch>.

Proceedings of the ICML 2025 Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada, 2025. Copyright 2025 by the author(s).

et al., 2023). This data collection bottleneck makes sample efficiency paramount, which requires selecting maximally informative queries. Existing PBRL methods for selecting such queries often face a trade-off: they are either computationally tractable but lack theoretical guarantees, or they are theoretically grounded but computationally expensive (Chen et al., 2022; Wu et al., 2023; Saha et al., 2023; Zhan et al., 2023; Pacchiano et al., 2023). This raises a fundamental question for making personalization practical:

Can we select PBRL queries in a way that is both statistically efficient, computationally tractable, and also theoretically guaranteed?

In this work, we address this question by leveraging the principles of Optimal Experimental Design (OED) (Chaloner & Verdinelli, 1995; Pukelsheim, 2006; Fedorov & Hackl, 1997). We propose a method to select the most informative queries to present to the user, ensuring that the preference model is learned with as few interactions as possible. Specifically, our objective is to determine a set of K distinct exploration policies for the generative agent. These policies are carefully chosen to generate a diverse, informative, and discerning set of outputs. When the user provides feedback on these outputs, we gain maximal information about their latent reward parameters. We do this by reformulating the generally intractable OED problem (Pukelsheim, 2006; Fedorov & Hackl, 1997) to a convex optimization problem over the space of state visitation measures induced by the policies. This allows us to use Convex Reinforcement Learning (Hazan et al., 2019) to efficiently compute the optimal set of policies for the query generation.

Our contributions To sum up, we provide the following:

- A formal problem setting for query selection for generative models, modeled via Markov Decision Processes (Sec. 3).
- ED-PBRL, a method that leverages Optimal Experimental Design (OED) to efficiently solve the problem of learning preferences from a minimal number of queries (Sec. 4 and 5.1).
- Convergence guarantees for ED-PBRL based on convex optimization analysis, ensuring the procedure finds a globally optimal set of query policies (Sec. 5.2).
- An experimental evaluation of the proposed method, showcasing promising performance for the personalization of text-to-image models (Sec. 6).

2. Related Work

Generative Model Guidance Generative models, especially diffusion models (Ho et al., 2020; Sohl-Dickstein et al., 2015; Dhariwal & Nichol, 2021) and Large Language Models (LLMs), have achieved remarkable success but often require guidance to align outputs with user preferences. For diffusion models, guidance techniques steer pre-trained models by incorporating preference information, for example, through gradients from an auxiliary classifier (*classifier guidance* (Dhariwal & Nichol, 2021; Song et al., 2021)) or by leveraging conditional model properties (*classifier-free guidance* (Ho & Salimans, 2022)). Similarly, LLMs are often guided in a post-training phase to better align with user intent; for instance, InstructGPT (Ouyang et al., 2022) uses human feedback to fine-tune models to follow instructions. The effectiveness of these methods often hinges on an accurate underlying preference model. Our work focuses on efficiently learning such preference models to enhance personalized generative model guidance.

Preference-Based Reinforcement Learning A key challenge in realizing effective generative model guidance is the accurate and efficient learning of the underlying user preference models. Preference-Based Reinforcement Learning (PBRL) offers a powerful paradigm for this, learning rewards (and thus preference models) from comparative feedback, which is often more intuitive for humans than providing explicit reward values or detailed demonstrations. While traditional Inverse Reinforcement Learning (IRL) methods also infer reward functions, often from expert demonstrations (Ziebart et al., 2008; Finn et al., 2016), and some IRL approaches actively query for expert actions to improve sample efficiency (Linder et al., 2022), PBRL's focus on preferences aligns well with capturing nuanced user tastes for guidance. Many PBRL advancements focus on statistical efficiency and regret guarantees (Chen et al., 2022; Saha et al., 2023; Zhan et al., 2023; Pacchiano et al., 2023). However, these methods can rely on computationally expensive components, such as oracles for selecting informative queries over pairs of policies from an exponentially large set, or complex algorithmic structures (Wu et al., 2023). Our work differs by focusing on a computationally tractable method for query selection in PBRL. We optimize a set of K exploration policies to generate informative comparative queries using an experiment design (ED) objective, rather than relying on pairwise policy comparison oracles.

Optimal Experiment Design To efficiently learn preference models for guidance, the queries presented to the user must be highly informative. Optimal Experimental Design (OED) (Pukelsheim, 2006; Fedorov & Hackl, 1997) provides principles for selecting experiments to maximize information gain, often by optimizing scalar criteria of the Fisher Information Matrix. Due to the NP-hardness of discrete design, continuous relaxations optimizing over design measures are common. Mutny et al. (2023) applied OED to active exploration in Markov Chains by optimizing over

visitation measures of a single policy. Our work adapts OED to PBRL by designing a *set of K policies* for generating informative comparative queries, a distinct problem setting, making it tractable using Convex Reinforcement Learning (Convex-RL) (Hazan et al., 2019).

3. Preliminaries and Problem Formulation

We frame the task of personalized content generation as an RL problem, where the agent sequentially appends to or refines its output. The reward function is unknown and defines the latent personal user's taste. Our goal is to learn this latent reward model using the fewest preference queries possible to be given user feedback upon.

3.1. MDPs and Latent Reward

We consider a finite-horizon Markov Decision Process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, H)$. Here, $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $P(s'|s,a)$ is the *known* transition matrix, and H is the finite horizon. A policy $\pi(a|s)$ defines a distribution over actions given a state, which induces a distribution over trajectories $\tau = \{(s_1, a_1), \ldots, (s_H, a_H)\}$.

We assume the user's latent reward function is linear in a known feature space. We assume access to a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ that transforms state-action pairs into d-dimensional embeddings. The user's preferences are driven by an unknown reward function $r(s,a) = (\theta^*)^\top \phi(s,a)$, parameterized by a true but unknown vector $\theta^* \in \mathbb{R}^d$. The objective of the learning process is to produce a good estimate $\hat{\theta}$ of θ^*.

Remark 3.1 (State vs. State-Action Rewards)**.** *Our framework is general and applies equally to state-based reward models, $r(s) = (\theta^*)^\top \phi(s)$, where features depend only on the state. For notational clarity in the theoretical sections, we often use state-based notation (e.g., features $\phi(s)$ and visitation measures $d(s)$). This is done without loss of generality, as all theorems can be extended to the state-action case by considering an augmented state space $\mathcal{S}' = \mathcal{S} \times \mathcal{A}$.*

3.2. Learning from Preference Feedback

To learn θ^*, we rely on comparative feedback rather than explicit reward values. Given K options (e.g., trajectories or states), denoted by $\{x_1, \ldots, x_K\}$, a user selects the one they prefer most. We model the probability of this choice using the standard multinomial logit (softmax) model. The probability that a user chooses option x_q is proportional to its latent reward:

$$P(x_q \text{ is best}) = \frac{\exp((\theta^*)^\top \phi(x_q))}{\sum_{k'=1}^{K} \exp((\theta^*)^\top \phi(x_k))} \qquad (1)$$

where $\phi(x_k)$ is the feature vector of option x_k. This model is a generalization of the Bradley-Terry model (which corresponds to $K = 2$) (Bradley & Terry, 1952).

3.3. Interaction Protocol

The learning process follows a fixed experimental design protocol with three phases:

1. **Policy Optimization:** The algorithm determines a set of K exploration policies, $\pi_1, \ldots, \pi_K$, by solving an information-maximization optimization problem (detailed in Section 4).

2. **Data Collection:** The K policies are executed for T episodes, generating T sets of trajectories. Each set is $\{\tau_{t,1}, \ldots, \tau_{t,K}\}$, where $\tau_{t,q} \sim \pi_q$. These sets (or their components, see below) are presented to the user, who provides one preference choice for each set, resulting in a dataset of T feedback points.

3. **Parameter Estimation:** Using the collected feedback and the features of the corresponding trajectories, the algorithm computes the final estimate $\hat{\theta}$ of the true parameter θ^*.

The central challenge, which we address, is how to perform Phase 1 to select policies that make the estimation in Phase 3 as efficient as possible.

3.4. Feedback Models

We consider two plausible models for how feedback is elicited over the generated trajectories.

State-based Preference Feedback At each timestep $h \in [H]$ of an episode, the user compares the states $\{s_{1,h}, \ldots, s_{K,h}\}$ reached by the K trajectories. The probability that the user selects state $s_{q,h}$ is given by the softmax model in Eq. 1, using the state features $\phi(s_{q,h})$. This model is a direct application of the general framework where the reward depends only on the state (see Remark 3.1).

Truncated Trajectory Feedback In many applications, evaluating cumulative progress is more natural. For instance, if trajectories are sequences of words forming a sentence, a user might prefer to compare partial sentences. In the *truncated trajectory feedback* model, we assume the user's preference at each timestep h is formed over the partial sequence of states $\sigma_q[1:h] = \{s_{q,1}, \ldots, s_{q,h}\}$. The probability of the user selecting the q-th sequence is given by Eq. 1 using features of that partial sequence, $\phi(\sigma_q[1:h])$. These features (e.g., a CLIP embedding of a partial sentence) are not necessarily simple sums of their constituent state features.

3.5. Estimation

Given a dataset of $T \times H$ preferences from the user, the algorithm estimates θ^* via regularized maximum likelihood. Let $y_{t,h,q}$ be a one-hot indicator that alternative q was chosen at step h of episode t, and let $p(q|t,h,\theta)$ be the probability of this choice under the relevant feedback model. The estimate $\hat{\theta}$ is:

$$\hat{\theta} = \arg\max_{\theta \in \mathbb{R}^d} \sum_{t=1,h=1,q=1}^{T,H,K} y_{t,h,q} \log(p(q|t,h,\theta)) + \frac{\lambda}{2}||\theta||_2^2$$

g in where $\lambda \geq 0$ is a regularization coefficient.

4. Optimal Experimental Design for Preference Learning

Our main motivation is selecting queries for PBRL in a sample efficient manner. This core challenge can be framed as:

> *Which exploration strategies yield trajectories that maximize information about θ?*

To address this, we use an information-theoretic approach, leveraging the Fisher Information Matrix.

4.1. Fisher Information and Estimation Error

The quality of the estimate $\hat{\theta}$ is fundamentally linked to the queries selected. The Fisher Information Matrix (FIM), $I(\theta)$, quantifies how informative these queries are. For regularized estimators like ours, maximizing the regularized FIM, $I_\lambda(\theta) = I(\theta) + \lambda I_d$, serves to reduce the overall estimation error. This is formalized in the following result.

Theorem 4.1 (Maximizing FIM improves Estimation)**.** *Under regularity conditions and local consistency assumptions detailed in Appendix B.1, the Mean Squared Error (MSE) matrix of the estimator θ_λ is bounded in terms of the inverse regularized FIM at the true parameter θ^*:*

$$\mathbb{E}[(\theta_\lambda - \theta^*)(\theta_\lambda - \theta^*)^T] \preceq C \cdot I_\lambda(\theta^*)^{-1}$$

for a constant $C > 0$ that depends on the local quality of the estimator.

Theorem 4.1 shows that maximizing $I_\lambda(\theta^*)$ in the Loewner sense (which makes its inverse smaller) is a principled way to reduce estimation error. The full proof, which leverages the self-concordance of the log-likelihood, is provided in Appendix B.1.

Our goal is thus to select a set of K policies, $\pi_{1:K}$, to maximize a scalar criterion of the expected regularized FIM they induce, $I_{reg}(\pi_{1:K}, \theta)$. However, this ideal objective (defined formally in Appendix B.3, Eq. 4) presents two major challenges:

- **Dependence on unknown** θ: The FIM depends on the true θ, which is unknown at the design stage.
- **Intractable Optimization**: The objective involves an expectation over an exponentially large trajectory space and optimization over the high-dimensional space of K policies.

4.2. Reformulation to a Tractable Objective

We address these challenges by deriving a tractable objective in three steps. Full details are in Appendix B.4.

Step 1: Reformulation using State Visitation Measures. The expected FIM, initially defined over policies $\pi_{1:K}$, can be equivalently expressed in terms of the state visitation measures $d_{1:K} = \{d^h_{\pi_q}\}_{h,q}$ induced by these policies. This shifts the problem from the space of policies to the space of visitation measures, but does not yet resolve the core challenges.

Step 2: Approximation for θ-Independence. To remove the dependency on the unknown θ at the design stage, we assume a uniform preference distribution over the K options, i.e., $p(q|s_{1..K}) \approx 1/K$. This is a standard approximation for initial designs and yields an approximate FIM, I_{approx}, that is independent of θ.

Step 3: Marginalization for Tractability. The expectation in the approximate FIM can be resolved into a tractable matrix form. As shown in Theorem B.4, the per-timestep contribution $I_{approx,h}(d^h_{1:K})$ can be computed efficiently:

$$I_{approx,h}(d^h_{1:K}) = \Phi^T \left(\frac{1}{K} \sum_{q=1}^{K} \text{diag}(d^h_q) - \bar{d}^h (\bar{d}^h)^T \right) \Phi \tag{2}$$

where Φ is the state feature matrix, d^h_q is the visitation vector for policy q at step h, and $\bar{d}^h = \frac{1}{K}\sum_q d^h_q$.

This yields our final practical experimental design objective: optimizing a scalar criterion $s(\cdot)$ over the state visitation measures $d_{1:K} = \{d^h_q\}_{q\in[K],h\in[H]}$:

$$\arg\max_{d_{1:K}} s\left(T \cdot \sum_{h=1}^{H} I_{approx,h}(d^h_{1:K}) + \lambda I_d \right) \tag{3}$$

This optimization is subject to the constraints that each d^h_q must be a valid visitation measure. The information decomposition of this objective, which highlights its preference for policy diversity, is discussed in Appendix B.4.1.

4.3. Information Equivalence of Feedback Models

A natural question is how the state-based and truncated trajectory feedback models relate in terms of information. Under a simplifying assumption that trajectory features are sums of state features, we can show a formal connection.

Theorem 4.2 (Informal: Information Relationship)**.** *If trajectory features decompose additively, then the information from truncated trajectory feedback is lower-bounded by the information from state-based feedback: $\mathcal{I}_{trunc} \succeq c \cdot \mathcal{I}_{state}$ for some constant $c > 0$.*

This result (formally stated and proven as Theorem B.2 in Appendix B.5) provides confidence that optimizing for the more analytically tractable state-based model is beneficial even when using the more user-friendly truncated trajectory model in practice.

5. The ED-PBRL Algorithm and Guarantees

5.1. Algorithm Overview

Our approach first determines K optimal exploration policies by maximizing the information objective from Eq. 3. This involves optimizing a scalar criterion $s(I_{total})$, where I_{total} is the total approximate expected regularized Fisher Information Matrix (FIM), i.e., the matrix argument of $s(\cdot)$ in Eq. 3. These policies then generate trajectories for user preference collection, which are used to estimate the reward parameters. The conceptual flow is:

Algorithm 1 ED-PBRL (Conceptual Overview)

Input: MDP details (M, Φ), design parameters $(K, T, s(\cdot), \lambda)$
Output: Estimated preference parameter $\hat{\theta}$
Phase 1: Compute Optimal State Visitation Measures
Solve Eq. 3 for optimal state visitation measures $\{d_q^{*h}\}_{h,q}$.
Phase 2: Policy Extraction and Trajectory Sampling
Extract policies $\{\pi_q^*\}_{q=1}^K$ from $\{d_q^{*h}\}_{h,q}$.
Sample $K \times T$ trajectories using $\{\pi_q^*\}$ and collect preference feedback.
Phase 3: Parameter Estimation
Estimate $\hat{\theta}$ using all collected feedback (cf. Section 3).

The algorithm proceeds in three phases. Phase 1 leverages Convex-RL, a Frank-Wolfe based method, to solve the convex optimization problem over visitation measures. Phase 2 derives policies from these measures and samples trajectories. Phase 3 uses the collected feedback for parameter estimation. The detailed algorithmic procedure and further explanation are in Appendix B.8.

5.2. Theoretical Guarantees

The ED-PBRL framework is theoretically well-founded. The core optimization problem (Eq. 3) for finding optimal state visitation measures is convex.

Theorem 5.1. *[Concavity of the Objective Function] Assume the scalar criterion $s : \mathbb{S}_+^d \to \mathbb{R}$ is concave and matrix-monotone non-decreasing. Then the objective function $f(D) = s(I_{total}(D))$, where $I_{total}(D)$ is the total approximate expected regularized FIM (the matrix argument of $s(\cdot)$ in Eq. 3), is concave with respect to the collection of state visitation vectors $D = \{d_q^h\}_{h\in[H],q\in[K]}$.*

Specifically, the objective function $s(I_{total}(D))$ (Theorem 5.1) is concave if the scalar criterion $s(\cdot)$ (e.g., D- or A-optimality) is concave and monotone. This ensures that the Frank-Wolfe based optimization (Algorithm 2) converges to a globally optimal set of policies.

Theorem 5.2 (Simplified Convergence Guarantee)**.** *Algorithm 2, which employs a Frank-Wolfe based method to optimize the objective $f(D) = s(I_{total}(D))$ (defined in Theorem 5.1) over the compact convex domain $\mathcal{D}_{sv}$ of state visitation measures, converges to a global optimum. If N_{iter} iterations are performed (i.e., the loop for n in Algorithm 2 runs N_{iter} times), the suboptimality of the final solution is bounded by $O(1/N_{iter})$.*

Proofs for these theorems, including a detailed version of Theorem 5.2 (as Theorem B.6), are provided in Appendix B.9.

6. Experimental Evaluation

We evaluate our Optimal Experimental Design (OED) approach for personalizing text-to-image generation based on CLIP embeddings (Radford et al., 2021). We conduct two types of experiments: (1) a quantitative evaluation using synthetic ground truth (GT) models to simulate user preferences, and (2) a qualitative study involving a real human-in-the-loop. An overview of the experimental flow is illustrated in Figure 1.

6.1. Experimental Methodology

Both our synthetic and human-feedback experiments are centered around a prompt construction task, modeled as an MDP, and share a common set of core components.

Prompt Construction MDP The environment is a finite-horizon MDP where states correspond to timesteps in the prompt creation process (H total steps). Actions involve selecting textual tokens (e.g., "Man drinking tea", "artistic") from a predefined vocabulary. A trajectory through this MDP forms a sequence of tokens, which are concatenated to create a textual prompt.

Figure 1. Simplified workflow for our experiments. ED-PBRL selects prompts, which are used by Stable Diffusion to generate images. Feedback on these images is collected (either from a synthetic GT model or a real human) and used to estimate the guidance model $\hat{\theta}$. A more detailed diagram is in Appendix (Figure 6).

Table 1. Summary of experimental parameters.

Common Parameters			
Feedback Model	Truncated Trajectory	OED Criterion	V-design (App. A.1)
Horizon (H)	6	Frank-Wolfe Iters (N)	100
Num. Policies (K)	4	FW Step Size	Line Search
CLIP Model	ViT-L/14		
Synthetic Experiment		**Human-Feedback Experiment**	
Feedback Source	GT Model	Feedback Source	Real Human
Num. Episodes (T)	10, 30, ..., 110	Num. Episodes (T)	30
Num. Runs	25	Episode Split	20 train / 10 test
Num. Test Prompts	1000	Vocabulary Split	N/A
Num. Eval Pairs	5000	Regularization (λ)	0.1
Vocabulary Split	75% train / 25% test	Evaluation Metrics	App. A.3
Regularization (λ)	100		
Evaluation Metrics	App. A.2		

Features and Preference Model The features $\phi(\cdot)$ for individual design tokens (used in the OED objective) and for full/partial prompts (used in preference modeling) are their respective CLIP text embeddings. We assume user preferences for prompts can be represented by a linear model $r(\text{prompt}) = \theta^\top \phi(\text{prompt})$. We model these preferences using the Truncated Trajectory Feedback model. As the features of a partial prompt are derived from the sequence of chosen tokens (actions), our setup uses a state-action based instance of this model (see Section 3.4 and Remark 3.1). At each timestep h, the user provides a preference over K partial prompts.

Design Objective and Optimization To efficiently learn θ, our ED-PBRL algorithm selects K exploration policies by optimizing an A-optimality criterion, $s(I_{total,reg}) = -\text{Tr}(V(I_{total,reg})^{-1})$, where V prioritizes minimizing uncertainty in relative preferences between tokens. The optimization is solved using the Convex-RL procedure (Algorithm 2). This design optimization is a one-time, offline cost.

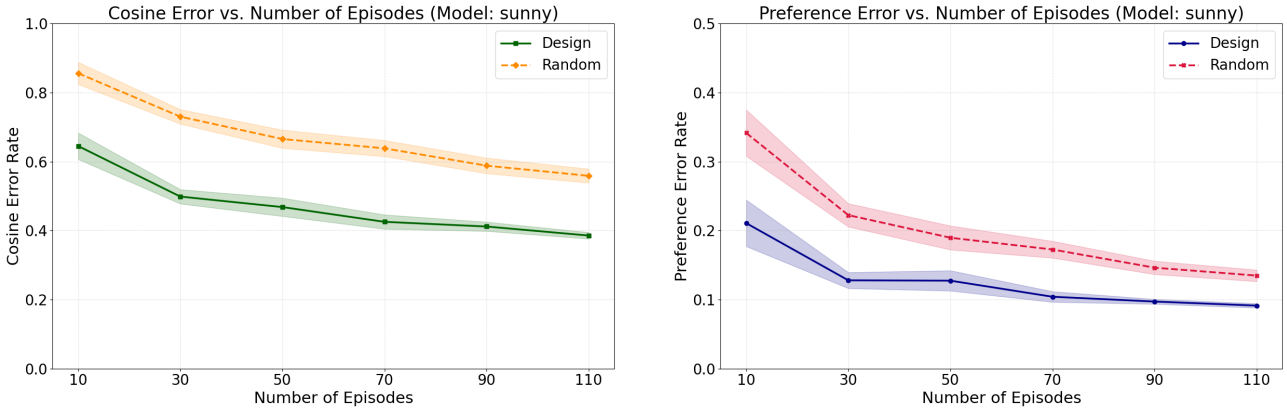
Experimental Parameters Table 1 summarizes the key parameters and settings for both the synthetic and human-feedback experiments.

6.2. Synthetic Ground Truth Model Experiments

This phase focuses on the quantitative evaluation of ED-PBRL against known GT preference models. We simulate a user whose preferences are dictated by a GT linear preference model θ^*. Each GT model is constructed from the normalized CLIP text embedding of a descriptive sentence. For instance, the Sunny GT model, which is the focus of our main results, uses the phrase *"An image with warm colors depicting bright sunshine"*. We also evaluate against Medieval and Technological GT models, with full details for all models provided in Appendix A.2. The goal is to measure how accurately and efficiently our method recovers this known θ^*. Performance is assessed using two main metrics (detailed in Appendix A.2):

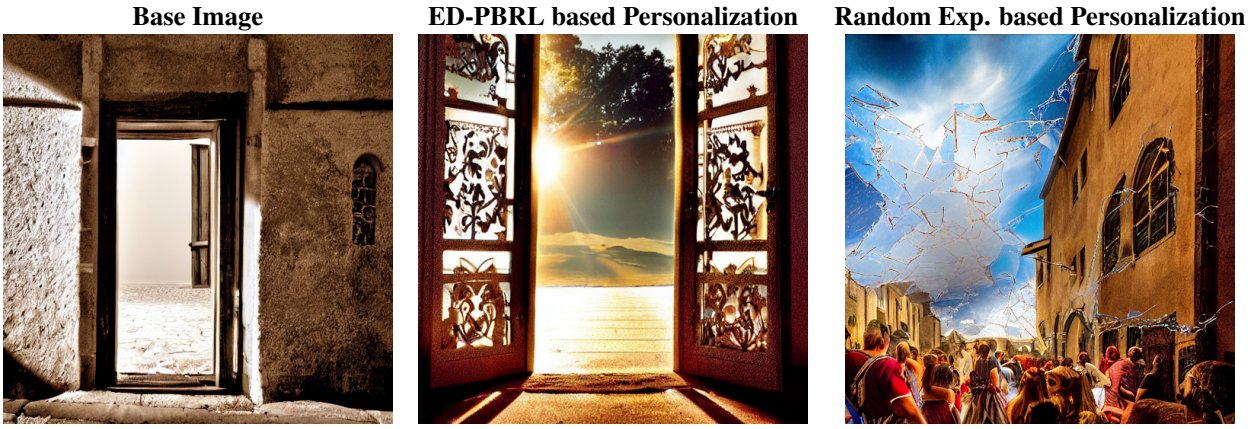
- **Cosine Error:** The cosine distance between the learned preference vector $\hat{\theta}$ and the GT vector θ^*.
- **Preference Prediction Error:** The error rate of $\hat{\theta}$ in predicting the synthetic user's preference on new, unseen pairs of prompts from a held-out test set of tokens.

Figure 2 presents the learning curves for these metrics for the Sunny GT model, averaged over multiple independent runs (see Table 1). The results show that ED-PBRL consistently learns the underlying preference model more effectively than random exploration, as evidenced by lower error rates. Similar trends hold for other GT models (see



(a) Cosine Error (Sunny GT) (b) Preference Error (Sunny GT)

Figure 2. Performance of ED-PBRL on the Sunny synthetic Ground Truth (GT) model. We plot the Cosine Error (left) and Preference Prediction Error (right) against the number of interaction episodes. These results demonstrate the efficiency of our OED approach. Numerical results for all GT models (Sunny, Medieval, and Technological) are presented in Appendix (Figure 7).



Prompt: "A photo of a gate" $\hat{\theta}_{\text{ED-PBRL}} \approx \theta^*_{\text{Sunny}}$ $\hat{\theta}_{\text{Random}} \approx \theta^*_{\text{Sunny}}$

Figure 3. **Synthetic Experiment:** Qualitative comparison demonstrating generalization for personalizing the base prompt "A photo of a gate" towards the "Sunny" GT model aesthetic. Both methods learn a preference model $\hat{\theta}$ to approximate the true preferences θ^*_{Sunny}. The personalized images are generated by using these learned models to select optimal style tokens from the held-out test vocabulary, testing their ability to generalize. The model learned via ED-PBRL successfully captures the target style, while the model from random exploration is less successful. See Appendix Fig. 8 for full details.

Appendix Figure 7). Figure 3 provides a qualitative understanding of these results, illustrating image generation guided by a model estimated by ED-PBRL versus a model learned from random exploration. The ED-PBRL-guided image better reflects the target "sunny" aesthetic.

6.3. Real Human-Feedback Experiment

To validate our approach in a real-world scenario, we conducted an experiment with a human participant. The user's stated goal was to personalize a text-to-image model to generate images with a "vintage photo" aesthetic.

Setup We collected feedback over $T = 30$ episodes. For each episode, the user was presented with a set of $K = 4$ images generated by ED-PBRL and another set of $K = 4$ images from random exploration. The query sets were shuffled to obscure which strategy generated them. The first 20 episodes were used to train two separate preference models, $\hat{\theta}_{\text{ED-PBRL}}$ and $\hat{\theta}_{\text{Random}}$, while the remaining 10 episodes were held out for evaluation.

Evaluation Metric Since there is no ground truth θ^* for a real user, we evaluate the learned model's ability to predict the user's own choices on unseen data. We use **Hold-out Preference Accuracy**: the percentage of times the learned model $\hat{\theta}$ correctly predicts the human's choice on the 10 held-out episodes. With a horizon of $H = 6$, there are $10 \times 6 = 60$ preference decisions in the test set.

Results The model learned by ED-PBRL achieved a Hold-out Preference Accuracy of 51.7% (31/60 correct predictions), significantly outperforming the random exploration baseline which achieved 33.3% (20/60). Both methods surpass the Random Guessing baseline of 25% (15/60). These results, summarized in Figure 4, indicate that our method successfully captured the user's preferences and could generalize to new prompts.

After the main experiment, the user was asked to choose several new, unseen base prompts for a qualitative generalization test. When asked to describe their taste, the user specified a preference for "foresty images with a lot of green, nature and landscapes." As shown in Figure 5, the model learned via ED-PBRL generated images that were more aligned with this specific "foresty" and "green" feel. In contrast, the model from random exploration produced images that, while often featuring landscapes, did not capture the user's nuanced preference as accurately. This pattern was consistent across multiple base prompts (see Appendix A.4), supporting the conclusion that ED-PBRL learned a more accurate preference model within the fixed feedback budget.

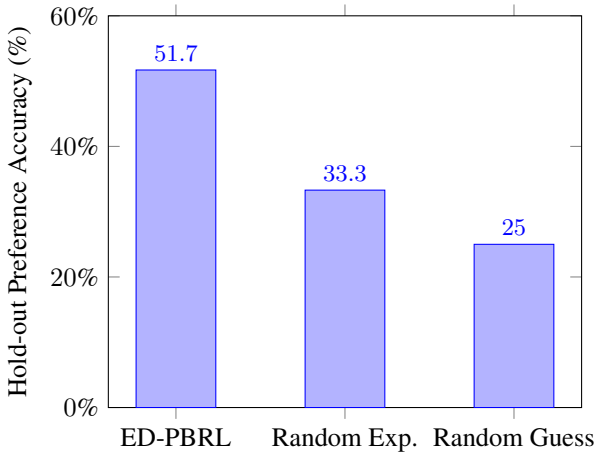


Figure 4. Hold-out preference accuracy for the human-feedback experiment. ED-PBRL correctly predicts the user's preference on held-out data more often than random exploration. Both methods outperform the 25% accuracy expected from Random Guessing. This theoretical baseline corresponds to the expected performance of choosing one of the $K = 4$ options at random for each of the 60 preference decisions.

7. Conclusion

We introduced ED-PBRL, a novel framework for efficiently personalizing generative models by learning user preferences from a minimal number of comparative queries. Our work demonstrates that the principles of Optimal Experimental Design (OED) can be practically and effectively applied to Preference-Based Reinforcement Learning (PBRL), leading to the following conclusions:

1. **Efficient Personalization through OED:** We established a formal connection between OED and PBRL for personalizing generative models. By framing query selection as an information-maximization problem, ED-PBRL significantly accelerates the learning of a user's latent reward function. This directly addresses the critical bottleneck of user feedback, enabling effective personalization with fewer interactions compared to standard random query selection.

2. **A Tractable Formulation for Information Maximization:** A core contribution is rendering the complex problem of optimal query selection tractable. We showed that the generally NP-hard problem of selecting a set of informative policies can be reformulated as a convex optimization problem over the space of state visitation measures. This reformulation, combined with principled approximations, makes our information-theoretic approach computationally feasible.

3. **Theoretically-Grounded and Scalable Algorithm:**

Figure 5. **Human-Feedback Experiment:** Qualitative results from the human-feedback experiment for the base prompt "Reflecting last year". The user's revealed preference was for "foresty images with a lot of green, nature and landscapes". The image generated using the model from ED-PBRL (middle) aligns well with this preference, while the image from the random exploration model (right) does not. Full results for all base prompts are in Appendix A.4.

The resulting algorithm, ED-PBRL, is not only practical but also theoretically sound. By leveraging Convex-RL, we provide guarantees that our method converges to a globally optimal set of query-generating policies. This provides a robust and scalable foundation for active preference learning in sequential decision-making settings.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Kernion, J., Ndousse, K., Olsson, C., Ringer, S., Kaplan, J., Brown, T., Johnston, S., McCandlish, S., Olah, C., and Amodei, D. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2021.

Azar, M. G., Guo, Z. D., Piot, B., Munos, R., Rowland, M., Valko, M., and Calandriello, D. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukosuite, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T., Hume, T., Bowman, S. R., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T., and Kaplan, J. Constitutional ai: Harmlessness from ai feedback, 2022.

Biyik, E., Malayandi, D. J., and Sadigh, D. Asking for demonstrations or preferences. In *Conference on Robot Learning*, pp. 1133–1144. PMLR, 2019.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., et al. Video generation models as world simulators. Technical report, OpenAI, 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. In *Advances in neural information processing systems*, volume 33, pp. 1877–1901, 2020.

Casper, S., Davies, X., Shi, C., Krendl, T., Pfister, J., Hadfield-Menell, D., Pautler, D., and Schulman, J. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023.

Chaloner, K. and Verdinelli, I. Bayesian experimental design: A review. *Statistical Science*, pp. 273–304, 1995.

Chen, L., Li, G., Li, A., Li, J., and Zhao, Y. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, pp. 3439–3467. PMLR, 2022.

Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in neural information processing systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

Deng, M., Wang, J., Zhang, C.-P., Zhang, H., Chen, Y., Li, B., Liu, J., Wang, Z., Wang, L., Chen, Y., et al. RLPrompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 9580–9591, 2022.

Dhariwal, P. and Nichol, A. Q. Diffusion models beat gans on image synthesis. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794, 2021.

Fedorov, V. V. and Hackl, P. *Model-Oriented Design of Experiments*. Lecture Notes in Statistics. Springer-Verlag New York, 1997.

Finn, C., Levine, S., and Abbeel, P. Guided cost learning: Deep inverse reinforcement learning via policy optimization. In *International conference on machine learning*, pp. 49–58. PMLR, 2016.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2): 95–110, 1956.

Hazan, E., Kakade, S., Singh, K., and Van Der Schaar, M. Provably efficient maximum entropy exploration. In *International Conference on Machine Learning*, pp. 2681–2691. PMLR, 2019.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In Dasgupta, S. and McAllester, D. (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *PMLR*, pp. 427–435, 2013.

Lee, H., Phatale, S., Mansoor, H., Mesnard, T., Ferret, J., Lu, K., Bishop, C., Dyer, E. H., and Sohl-Dickstein, J. RLAIF: Scaling reinforcement learning from human feedback with AI feedback, 2023.

Linder, J., Groot, D. A. d., Mutnỳ, M., Adeshina, T., Lyu, M., and Krause, A. Aceirl: Active exploration for inverse reinforcement learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3207–3220. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/26d01e5e700931b3bf389389789d8010-Abstract-Confere
html.

Mutnỳ, M. *Modern Adaptive Experiment Design: Machine Learning Perspective*. PhD thesis, ETH Zurich, 2024.

Mutny, M., Janik, T., and Krause, A. Active exploration via experiment design in markov chains. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7349–7374. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/mutny23a.html.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Pacchiano, A., Saha, A., and Lee, J. Dueling rl: Reinforcement learning with trajectory preferences, 2023. URL https://arxiv.org/abs/2111.04850.

Pukelsheim, F. *Optimal Design of Experiments*. SIAM, 2006.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. Learning transferable visual models from natural language supervision, 2021. URL https://arxiv.org/abs/2103.00020.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741, 2023.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Sadigh, D., Dragan, A. D., Sastry, S., and Seshia, S. A. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017. URL http://www.roboticsproceedings.org/rss13/p42.pdf.

Saha, A., Ramaswamy, A., Krishnamurthy, A., and Agarwal, A. Dueling RL: Reinforcement learning with trajectory preferences. In Ruiz, F., Dy, J., and van de Meent, J.-W. (eds.), *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 609–631. PMLR, 25–27 Apr 2023. URL https://proceedings.mlr.press/v206/saha23a.html.

Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pp. 2256–2265. PMLR, 2015.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Finn, C., Christiano, P., and Schulman, J. Learning to summarize from human feedback, 2020.

Wu, Y., Lykouris, T., Slivkins, A., and Xu, H. Making reinforcement learning with preference-based feedback efficient via randomization. In *Conference on Learning Theory*, pp. 4406–4453. PMLR, 2023.

Xu, J., Liu, X., Wu, Y., Wang, Y., Cao, Y., Dai, J., Wu, R., Wei, Y., Li, Z., Li, W., et al. ImageReward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Zhan, W., Zheng, M., Liu, C., Zhao, P., Wang, M., and Xie, T. Provable offline reinforcement learning with human feedback. In *The Eleventh International Conference on Learning Representations*, 2023.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., and Dey, A. K. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438, 2008.

A. Appendix: Detailed Experimental Setup

This section provides a comprehensive description of the experimental environment, parameters, and models used in our evaluation, intended for reproducibility and completeness. The overall workflow is depicted in Figure 6. A summary of key parameters is available in the main paper in Table 1.

A.1. Common Experimental Components

Environment: Prompt Construction MDP The environment is modeled as a finite-horizon Markov Decision Process (MDP) designed to simulate the construction of textual prompts.

- **States ($\mathcal{S}$):** States $s \in \{0, 1, \ldots, H-1\}$ directly correspond to the current timestep or depth in the prompt construction process.
- **Horizon (H):** The horizon corresponds to the number of vocabulary files used for sequential token selection.
- **Actions ($\mathcal{A}$):** Actions are indices corresponding to unique "design tokens" extracted from the vocabulary files. These tokens represent semantic concepts (e.g., "Man sitting", "artistic", "happy").
- **Vocabulary:** The vocabulary is sourced from H files: ‘bases.txt‘, ‘ambient.txt‘, ‘style.txt‘, ‘composition.txt‘, ‘lighting.txt‘, ‘detail.txt‘. The selection of tokens is structured by timestep. At $s = 0$, only "base" concepts are allowed. For $s > 0$, tokens from other categories are used.
- **Transitions (P):** Deterministic. Selecting a token at state s transitions to state $s + 1$.
- **Feature Representation for OED ($\phi(a)$):** The features for design tokens are their 768-dimensional, normalized CLIP text embeddings (‘ViT-L/14‘).

Preference Model and Estimation

- **Feedback Model:** We use the Truncated Trajectory Feedback model (Section 3.4) for both experiments. At each timestep h, a preference is given over K partial prompts $\{\tau_1[0:h], \ldots, \tau_K[0:h]\}$.
- **Features for Estimation (ϕ(partial prompt)):** The feature vector for a partial prompt is its normalized CLIP text embedding (‘ViT-L/14‘).

Experimental Design (OED) The experimental design objective is to select policies that maximize information about θ.

- **Scalarization Criterion $s(\cdot)$:** We use an A-optimality variant, $s(I_{total,reg}) = -\text{Tr}\left(V(I_{total,reg})^{-1}\right)$, where $I_{total,reg}$ is the regularized total approximate FIM from Eq. 3.
- **V Matrix Construction:** The matrix $V = C^T C$ is constructed from differences between feature embeddings of tokens from the same thematic category (excluding ’bases.txt’), i.e., $c_k^T = (\phi(a_i) - \phi(a_j))^T$. This c-optimality criterion directly targets the precision of estimated preference differences, which is essential for learning an effective ranking model. The full construction is detailed in the original appendix text.

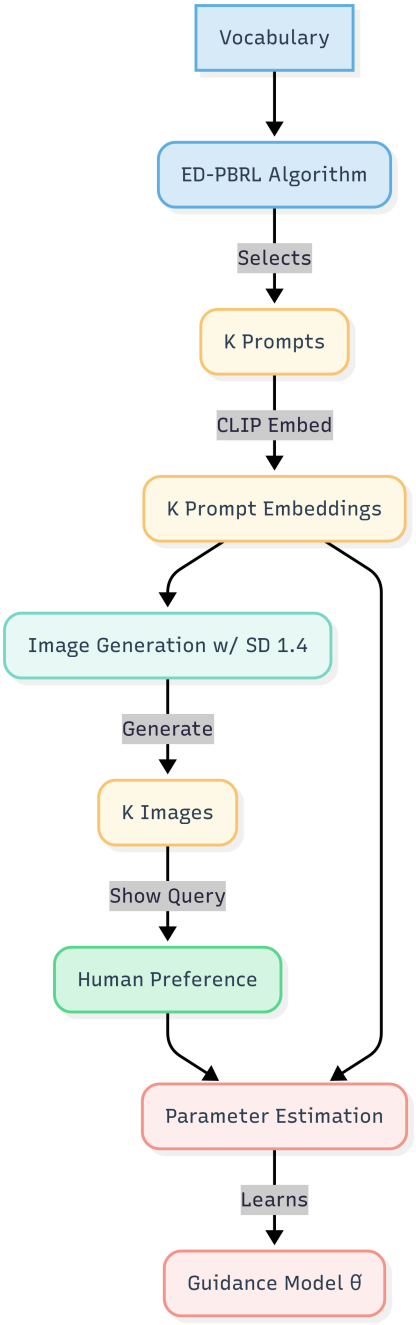


Figure 6. Detailed workflow simulated in the synthetic user experiment. This diagram illustrates the generation of prompts by ED-PBRL, image creation via Stable Diffusion, feedback collection (simulated by a GT model), and subsequent guidance model learning. It expands on the simplified flow shown in Figure 1.

- **Optimization:** The Convex-RL procedure (Algorithm 2) is used to solve the design problem.
- **Computational Cost:** The one-time design optimization for a vocabulary of approximately 5000 tokens takes around 10 minutes on a single NVIDIA A100 GPU.

A.2. Synthetic Ground Truth Model Experiments: Setup and Metrics

Ground Truth Scorer Models For the synthetic experiments, we simulate user preferences using three distinct ground truth (GT) scorer models. Each is represented by a weight vector $\theta^* \in \mathbb{R}^d$ constructed by taking the normalized CLIP text embedding of a descriptive sentence:

- **Sunny GT Model (θ^*_{sunny}):** From `CLIP("An image with warm colors depicting bright sunshine")`.
- **Medieval GT Model ($\theta^*_{medieval}$):** From `CLIP("An image with ancient kingdom depicting medieval times")`.
- **Technological GT Model ($\theta^*_{technological}$):** From `CLIP("An image with advanced technologies depicting futuristic style")`.

The GT vector θ^* is used to simulate user choices and serves as the ground truth for evaluation.

Evaluation Metrics

- **Cosine Error:** $1 - \text{cosine\_similarity}(\hat{\theta}, \theta^*)$. Measures the angular deviation between the estimated preference vector and the ground truth θ^*.
- **Preference Prediction Error:** The fraction of pairs where $\hat{\theta}$'s prediction mismatches the GT's preference on prompts generated exclusively from the held-out testing vocabulary.

A.3. Human-Feedback Experiment: Setup and Metrics

Setup This experiment involved one human participant aiming to personalize the model to a "vintage photo" aesthetic. We collected feedback for $T = 30$ episodes. The first 20 episodes were used for training the preference model $\hat{\theta}$, and the final 10 episodes were held out for testing.

Evaluation Metric

- **Hold-out Preference Accuracy:** Since no ground truth θ^* exists, we measure how well the learned model predicts the user's own choices on unseen data. This is the percentage of times that the preference predicted by $\hat{\theta}$ (i.e., $\arg\max_q \hat{\theta}^\top \phi(\tau_q[1:h])$) matches the actual choice made by the human user on the 10 held-out test episodes. With a horizon of $H = 6$, this evaluation is performed over a total of $10 \times 6 = 60$ preference decisions.

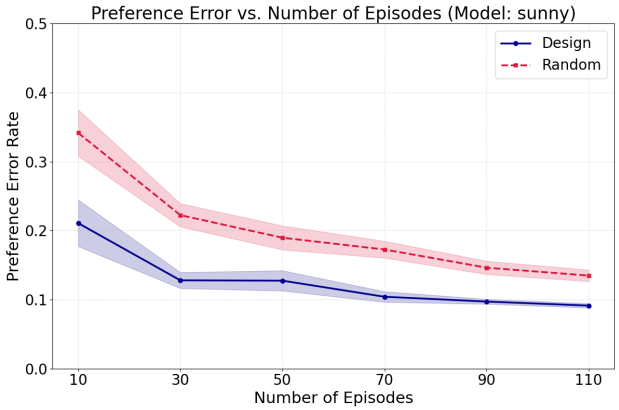
A.4. Full Numerical and Qualitative Results

This section provides the full set of results for all experiments.

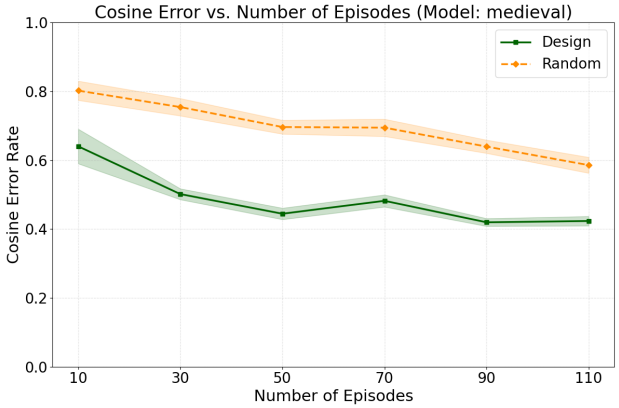
Synthetic Experiment Results

Qualitative Results Summary (Synthetic) For each Ground Truth (GT) model (Sunny, Medieval, Technological), Figures 8, 9, and 10 show a visual comparison of the prompts generated by ED-PBRL (Design) and Random exploration. The figures correspond to the median cosine error run (out of 25 seeds) after $T = 110$ feedback episodes with $K = 4$ policies. To test generalization, the personalized prompts are constructed by adding style tokens selected from the held-out test vocabulary to a base prompt.

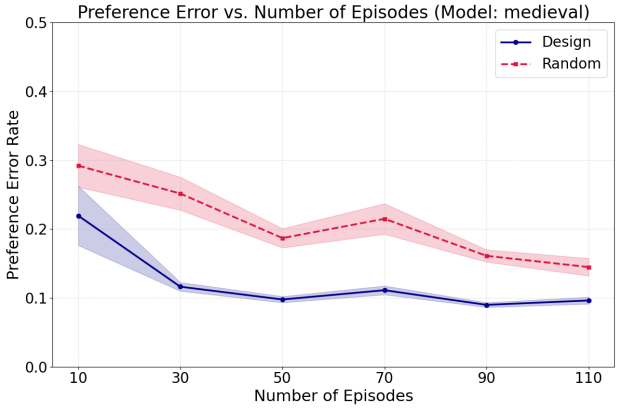
(a) Cosine Error (Sunny GT)



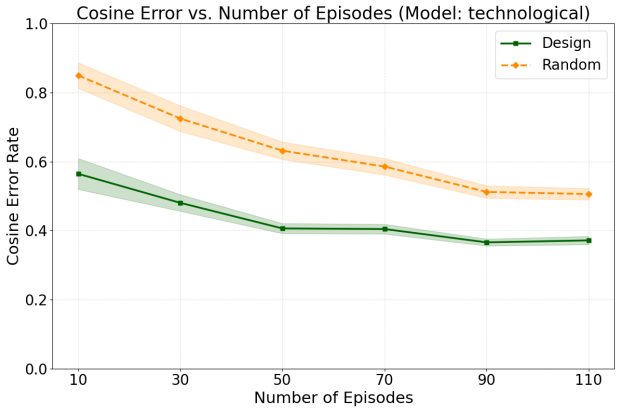
(b) Preference Error (Sunny GT)



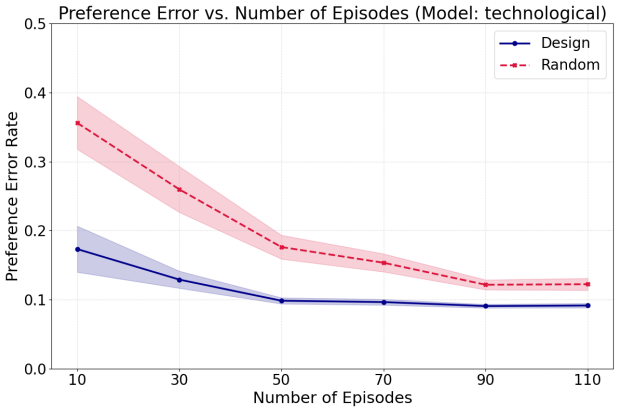
(c) Cosine Error (Medieval GT)



(d) Preference Error (Medieval GT)

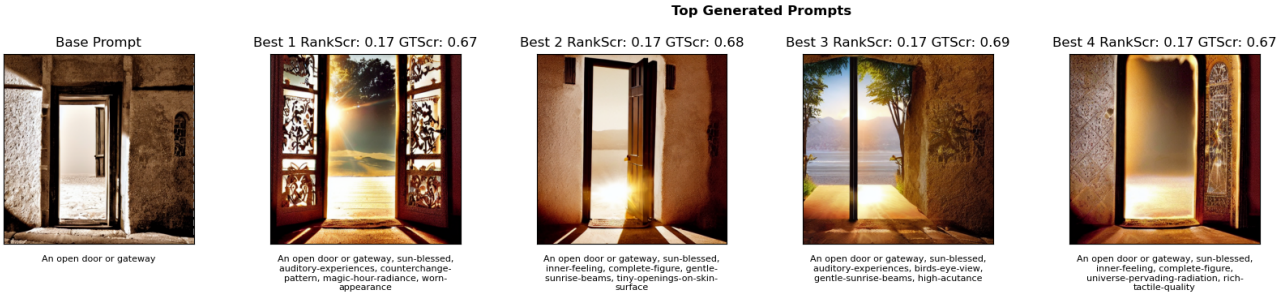


(e) Cosine Error (Technological GT)

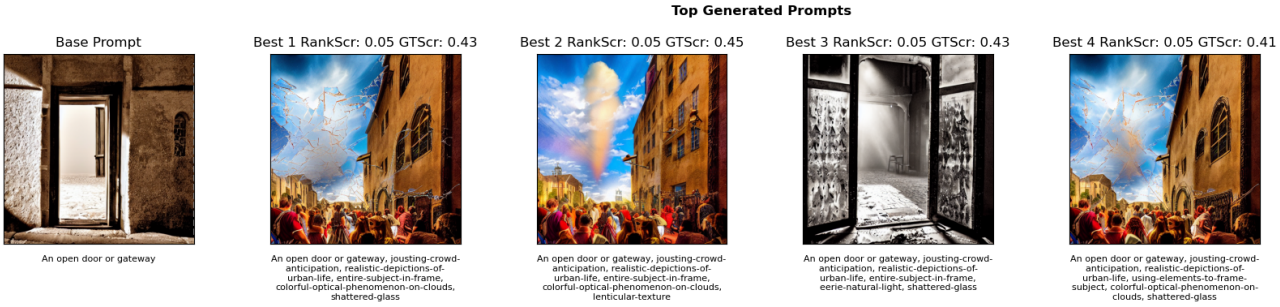


(f) Preference Error (Technological GT)

Figure 7. Performance of ED-PBRL on Sunny, Medieval, and Technological synthetic Ground Truth (GT) models. For each GT model, we plot the Cosine Error (left column) and Preference Prediction Error (right column) against the number of interaction episodes. Results are averaged over N=25 independent runs, and the shaded regions represent the standard error of the mean. The Sunny GT model results are also shown in the main paper (Figure 2).

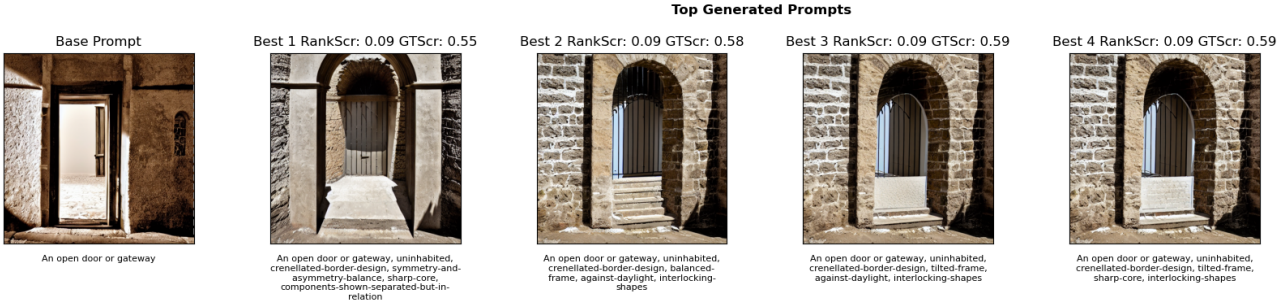


(a) ED-PBRL (Design) - Top Prompts for Sunny GT

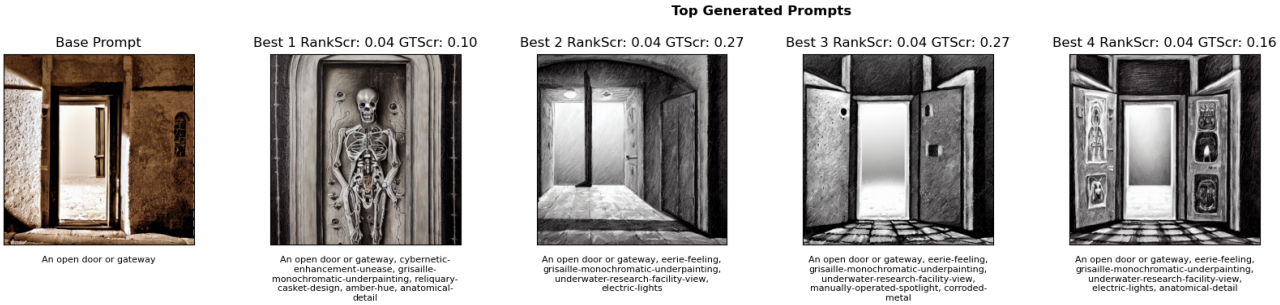


(b) Random Exploration - Top Prompts for Sunny GT

Figure 8. Full summary of top generated prompts for the Sunny GT Model. The images compare prompts generated via ED-PBRL (Design) and Random exploration. Each personalized image is annotated with its estimated score from the learned model (RankScore) and its true score from the ground truth model (GTScore), where a higher GTScore indicates better alignment with the target 'Sunny' aesthetic. Note that ED-PBRL consistently finds prompts that yield higher GT Scores, demonstrating its superior personalization capability.

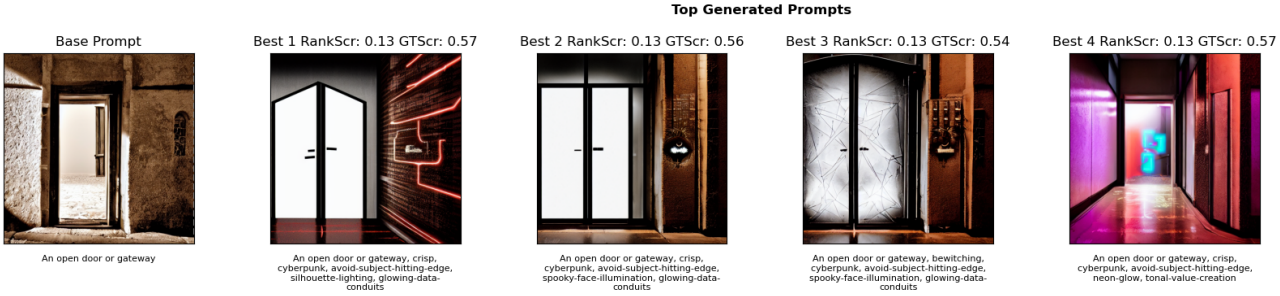


(a) ED-PBRL (Design) - Top Prompts for Medieval GT

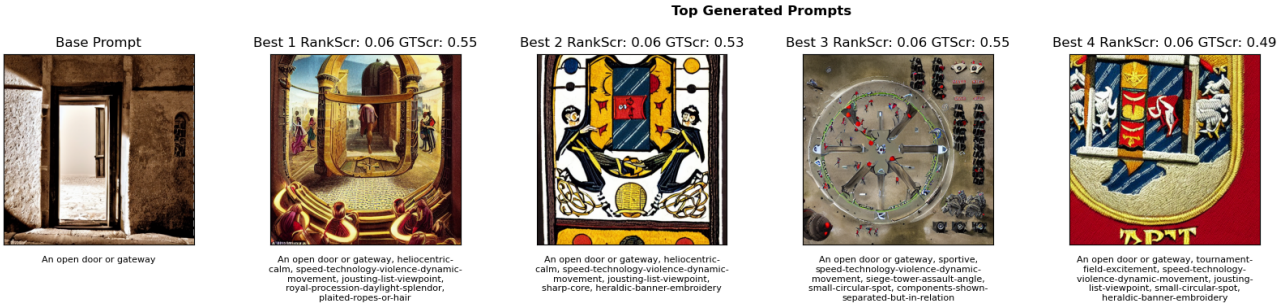


(b) Random Exploration - Top Prompts for Medieval GT

Figure 9. Full summary of top generated prompts for the Medieval GT Model. The images compare prompts generated via ED-PBRL (Design) and Random exploration. Each personalized image is annotated with its estimated score from the learned model (RankScore) and its true score from the ground truth model (GTScore), where a higher GTScore indicates better alignment with the target 'Medieval' aesthetic. Note that ED-PBRL consistently finds prompts that yield higher GT Scores, demonstrating its superior personalization capability.



(a) ED-PBRL (Design) - Top Prompts for Technological GT

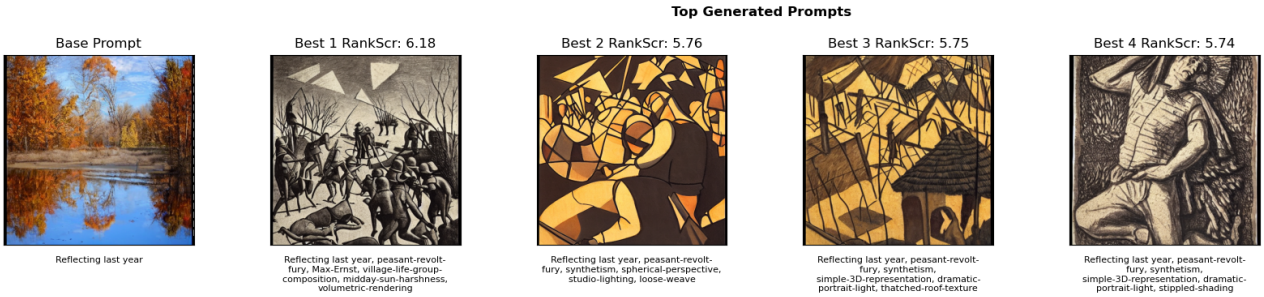


(b) Random Exploration - Top Prompts for Technological GT

Figure 10. Full summary of top generated prompts for the Technological GT Model. The images compare prompts generated via ED-PBRL (Design) and Random exploration. Each personalized image is annotated with its estimated score from the learned model (RankScore) and its true score from the ground truth model (GTScore), where a higher GTScore indicates better alignment with the target 'Technological' aesthetic. Note that ED-PBRL consistently finds prompts that yield higher GT Scores, demonstrating its superior personalization capability.

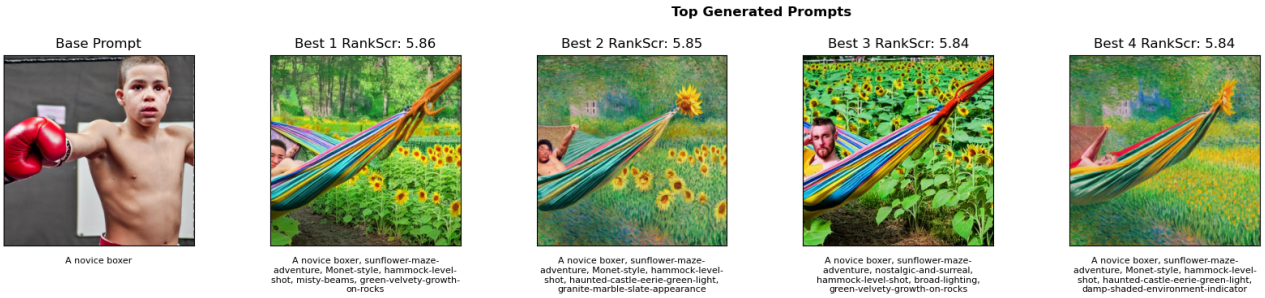
Qualitative Results Summary (Human Feedback) This section presents the full qualitative results for the human-feedback experiment. After the main feedback collection phase, the user chose four new base prompts to test the generalization of the learned preference models ($\hat{\theta}_{\text{ED-PBRL}}$ and $\hat{\theta}_{\text{Random}}$). The following figures show the top-ranked personalized images generated by each model for these base prompts. The user's revealed preference was for "foresty images with a lot of green, nature and landscapes."

(a) ED-PBRL (Design) - Top Prompts for "Reflecting last year"

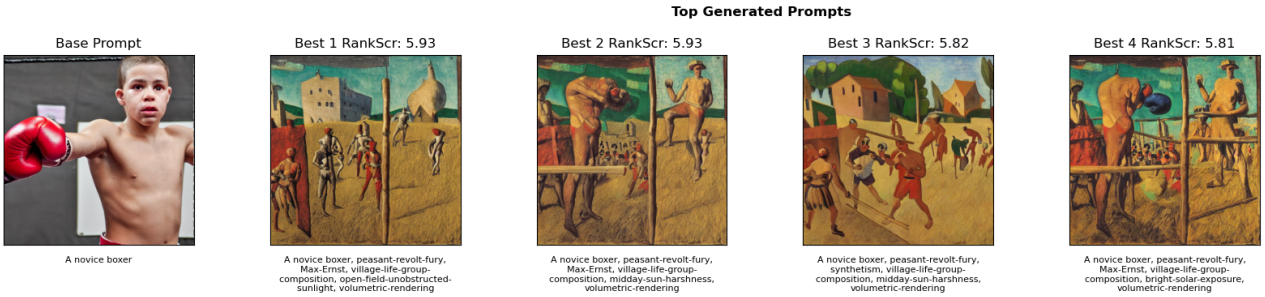


(b) Random Exploration - Top Prompts for "Reflecting last year"

Figure 11. Full summary of top generated prompts for the base prompt "Reflecting last year" from the human-feedback experiment. The images are ranked according to the score from the respective learned models (RankScore).

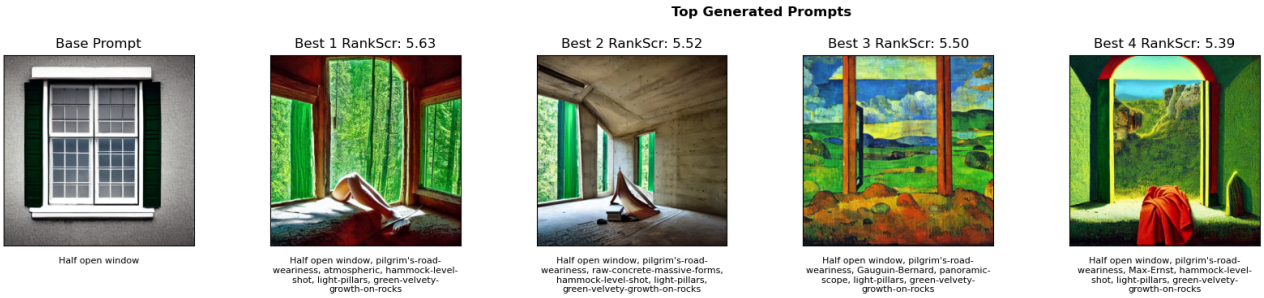


(a) ED-PBRL (Design) - Top Prompts for "A novice boxer"



(b) Random Exploration - Top Prompts for "A novice boxer"

Figure 12. Full summary of top generated prompts for the base prompt "A novice boxer" from the human-feedback experiment. The images are ranked according to the score from the respective learned models (RankScore).



(a) ED-PBRL (Design) - Top Prompts for "Half open window"

(b) Random Exploration - Top Prompts for "Half open window"

Figure 13. Full summary of top generated prompts for the base prompt "Half open window" from the human-feedback experiment. The images are ranked according to the score from the respective learned models (RankScore).

(a) ED-PBRL (Design) - Top Prompts for "A family vibing"

(b) Random Exploration - Top Prompts for "A family vibing"

Figure 14. Full summary of top generated prompts for the base prompt "A family vibing" from the human-feedback experiment. The images are ranked according to the score from the respective learned models (RankScore).

B. Appendix: Proofs, Derivations and Further Results

B.1. Relationship between Estimation Error and Fisher Information

To design experiments that yield accurate parameter estimates, we need a measure of the information provided by the data. The Fisher Information Matrix (FIM), $I(\theta)$, is central here. For unbiased estimators, the Cramér-Rao Lower Bound states that the estimator's covariance is lower-bounded by $I(\theta^*)^{-1}$. Our estimator $\hat{\theta}$ (denoted θ_λ) is obtained via regularized maximum likelihood and is generally biased. For such estimators, it is more direct to establish an upper bound on the Mean Squared Error (MSE) matrix, $\mathbb{E}[(\theta_\lambda - \theta^*)(\theta_\lambda - \theta^*)^T]$, in terms of the inverse regularized FIM at the true parameter, $I_\lambda(\theta^*)^{-1} = (I(\theta^*) + \lambda I_d)^{-1}$. This motivates maximizing a scalar function of $I_\lambda(\theta^*)$ to reduce the overall estimation error.

The following theorem formalizes this relationship. The proof leverages the self-concordance property of the logistic regression log-likelihood function. This property is crucial as it ensures the Fisher Information Matrix (the Hessian of the negative log-likelihood) does not change drastically in a local neighborhood of the true parameter θ^*. This allows us to control the Hessian at an intermediate point from a Taylor expansion by relating it to the Hessian at θ^*.

B.1.1. Assumptions for the MSE Bound

The derivation of the bound relies on two key assumptions. We state them formally here before proceeding with the proof.

Assumption B.1 (Uniform Local Consistency)**.** *For a given experimental design, total number of samples T, and regularization parameter λ, the resulting regularized maximum likelihood estimator θ_λ is guaranteed to lie within a local norm ball of radius $r < 1$ around the true parameter θ^*, for any realization of the data. Specifically, we assume there exists a constant $r \in [0, 1)$ such that:*

$$\|\theta_\lambda - \theta^*\|_{I(\theta^*)} \equiv \sqrt{(\theta_\lambda - \theta^*)^T I(\theta^*)(\theta_\lambda - \theta^*)} \leq r$$

This assumption is necessary for our finite-sample analysis. It requires the condition to hold deterministically for all data realizations, which ensures that the radius r is a constant. This allows us to define a non-random constant $C = (1 - r)^{-4}$ that can be moved outside the expectation in the proof, simplifying the analysis by avoiding more complex concentration arguments. Standard large-sample theory for MLE suggests that θ_λ converges to θ^* (or a neighborhood for biased estimators), so for a sufficiently large number of samples, this condition is expected to hold with high probability.

Assumption B.2 (Bounded True Parameter)**.** *The squared ℓ_2-norm of the true parameter vector θ^* is bounded relative to the regularization strength λ:*

$$\|\theta^*\|_2^2 \leq \frac{1}{\lambda}$$

This assumption, which is standard in the analysis of ridge regression and regularized estimators (Mutnỳ, 2024), constrains the magnitude of the true parameter relative to the regularization strength. It ensures that the bias introduced by the ℓ_2 penalty does not overwhelm the information-related terms in the analysis. In matrix terms, this implies $\lambda^2 \theta^* (\theta^*)^T \preceq \lambda I_d$.

With these conditions explicitly stated, we can now present the formal theorem and its proof.

Theorem B.1 (Upper Bound on MSE)**.** *Let θ_λ be the regularized maximum likelihood estimator and θ^* be the true parameter. Under Assumptions B.1 and B.2, the Mean Squared Error (MSE) matrix of the estimator is bounded by:*

$$\mathbb{E}[(\theta_\lambda - \theta^*)(\theta_\lambda - \theta^*)^T] \preceq C \cdot I_\lambda(\theta^*)^{-1}$$

where $I_\lambda(\theta^*) = I(\theta^*) + \lambda I_d$ is the regularized Fisher Information Matrix at the true parameter, and the constant $C = (1 - r)^{-4}$ depends on the radius r from Assumption B.1.

Proof. The proof proceeds in three main steps. First, we establish an exact expression for the estimation error using a Taylor expansion. Second, we use the self-concordance property of the negative log-likelihood, combined with Assumption B.1, to bound the random Hessian that appears in the error expression. Finally, we combine these results and use Assumption B.2 to derive the upper bound on the MSE.

Step 1: Taylor Expansion of the Score Function. The estimator θ_λ is the solution to the regularized maximum likelihood problem, defined by the first-order optimality condition $s_\lambda(\theta_\lambda) = 0$. The regularized score function $s_\lambda(\theta)$ is the gradient of the regularized log-likelihood:

$$s_\lambda(\theta) = \nabla L_{reg}(\theta) = \sum_{t=1}^{T} \sum_{q=1}^{K} (y_{t,q} - p_{t,q}(\theta)) x_{t,q} - \lambda\theta$$

The Hessian of the negative regularized log-likelihood is the regularized Fisher Information Matrix, $I_\lambda(\theta) = -\nabla^2 L_{reg}(\theta) = I(\theta) + \lambda I_d$. Note that $\nabla s_\lambda(\theta) = -I_\lambda(\theta)$.

By the vector-valued Mean Value Theorem (a form of Taylor's theorem), we can expand the function $s_\lambda(\theta)$ around the true parameter θ^*:

$$0 = s_\lambda(\theta_\lambda) = s_\lambda(\theta^*) + \nabla s_\lambda(\theta_\tau)(\theta_\lambda - \theta^*)$$

for some θ_τ on the line segment between θ^* and θ_λ. Substituting $\nabla s_\lambda(\theta_\tau) = -I_\lambda(\theta_\tau)$, we get:

$$0 = s_\lambda(\theta^*) - I_\lambda(\theta_\tau)(\theta_\lambda - \theta^*)$$

Rearranging gives the exact expression for the estimation error:

$$\theta_\lambda - \theta^* = I_\lambda(\theta_\tau)^{-1} s_\lambda(\theta^*)$$

The MSE matrix is therefore given by the expectation:

$$\mathbb{E}[(\theta_\lambda - \theta^*)(\theta_\lambda - \theta^*)^T] = \mathbb{E}[I_\lambda(\theta_\tau)^{-1} s_\lambda(\theta^*) s_\lambda(\theta^*)^\top I_\lambda(\theta_\tau)^{-1}]$$

Step 2: Bounding the Hessian via Self-Concordance. The main difficulty is relating the terms $I_\lambda(\theta_\tau)$ and $s_\lambda(\theta^*)$ in the error expression, as they are evaluated at different points. We resolve this by bounding the Hessian term $I_\lambda(\theta_\tau)$ using the self-concordance property of the unregularized negative log-likelihood function, $L(\theta) = -\log P(\text{data}|\theta)$.

The negative log-likelihood for multinomial logistic regression is a sum of log-sum-exp functions, which is a standard example of a self-concordant function. Its Hessian is the Fisher Information Matrix, $I(\theta) = \nabla^2 L(\theta)$. For a self-concordant function f, the Hessians at two points x, y are related by $(1 - \|y - x\|_x)^2 \nabla^2 f(x) \preceq \nabla^2 f(y)$ provided that the local norm $\|y - x\|_x = \sqrt{(y-x)^T \nabla^2 f(x)(y-x)}$ is less than 1.

We now invoke **Assumption B.1**, which states that $\|\theta_\lambda - \theta^*\|_{I(\theta^*)} \leq r < 1$ for all data realizations. Since θ_τ lies on the line segment between θ^* and θ_λ, it is necessarily closer to θ^* in any norm, including the local norm defined by $I(\theta^*)$. Thus, $\|\theta_\tau - \theta^*\|_{I(\theta^*)} \leq \|\theta_\lambda - \theta^*\|_{I(\theta^*)} \leq r$.

Applying the self-concordance property with $f(\theta) = L(\theta)$, $x = \theta^*$, and $y = \theta_\tau$, we get a lower bound on the unregularized FIM:

$$I(\theta_\tau) \succeq (1 - \|\theta_\tau - \theta^*\|_{I(\theta^*)})^2 I(\theta^*) \succeq (1 - r)^2 I(\theta^*)$$

This inequality holds deterministically for any realization of the data due to our assumption. We use this to bound the regularized FIM:

$$\begin{aligned} I_\lambda(\theta_\tau) &= I(\theta_\tau) + \lambda I_d \\ &\succeq (1-r)^2 I(\theta^*) + \lambda I_d \\ &\succeq (1-r)^2 I(\theta^*) + (1-r)^2 \lambda I_d \quad \text{(since } 0 < (1-r)^2 \leq 1 \text{ and } \lambda I_d \text{ is pos. semidef.)} \\ &= (1-r)^2 (I(\theta^*) + \lambda I_d) = (1-r)^2 I_\lambda(\theta^*) \end{aligned}$$

Inverting this matrix inequality (using the property that if $A \succeq B \succ 0$, then $B^{-1} \succeq A^{-1} \succ 0$) yields an upper bound on the inverse Hessian:

$$I_\lambda(\theta_\tau)^{-1} \preceq (1-r)^{-2} I_\lambda(\theta^*)^{-1}$$

Step 3: Deriving the Final MSE Bound. We substitute the bound on the inverse Hessian back into the MSE expression. Since the bound holds deterministically for a constant r, the term $(1-r)^{-2}$ is a constant and can be manipulated outside the expectation.

$$\begin{aligned} \mathbb{E}[(\theta_\lambda - \theta^*)(\theta_\lambda - \theta^*)^T] &= \mathbb{E}[I_\lambda(\theta_\tau)^{-1} s_\lambda(\theta^*) s_\lambda(\theta^*)^\top I_\lambda(\theta_\tau)^{-1}] \\ &\preceq \mathbb{E}\left[\left((1-r)^{-2} I_\lambda(\theta^*)^{-1} \right) s_\lambda(\theta^*) s_\lambda(\theta^*)^\top \left((1-r)^{-2} I_\lambda(\theta^*)^{-1} \right) \right] \\ &= (1-r)^{-4} I_\lambda(\theta^*)^{-1} \, \mathbb{E}[s_\lambda(\theta^*) s_\lambda(\theta^*)^\top] I_\lambda(\theta^*)^{-1} \end{aligned}$$

Next, we analyze the expectation of the outer product of the regularized score at the true parameter, $\mathbb{E}[s_\lambda(\theta^*)s_\lambda(\theta^*)^\top]$. Let $s(\theta^*)$ be the unregularized score. We know that $\mathbb{E}[s(\theta^*)] = 0$ and $\mathbb{E}[s(\theta^*)s(\theta^*)^\top] = I(\theta^*)$ (by the Information Matrix Equality).

$$\begin{aligned}\mathbb{E}[s_\lambda(\theta^*)s_\lambda(\theta^*)^\top] &= \mathbb{E}[(s(\theta^*) - \lambda\theta^*)(s(\theta^*) - \lambda\theta^*)^\top] \\ &= \mathbb{E}[s(\theta^*)s(\theta^*)^\top] - \lambda\,\mathbb{E}[s(\theta^*)](\theta^*)^\top - \lambda\theta^*\,\mathbb{E}[s(\theta^*)^\top] + \lambda^2\theta^*(\theta^*)^\top \\ &= I(\theta^*) - 0 - 0 + \lambda^2\theta^*(\theta^*)^\top \\ &= I(\theta^*) + \lambda^2\theta^*(\theta^*)^\top\end{aligned}$$

Now, we invoke **Assumption B.2**, which states $\|\theta^*\|_2^2 \leq 1/\lambda$. This implies that $\lambda^2\theta^*(\theta^*)^T \preceq \lambda I_d$. Using this, we can bound the expected score term:

$$\mathbb{E}[s_\lambda(\theta^*)s_\lambda(\theta^*)^\top] = I(\theta^*) + \lambda^2\theta^*(\theta^*)^\top \preceq I(\theta^*) + \lambda I_d = I_\lambda(\theta^*)$$

Finally, substituting this back into the MSE expression gives the result:

$$\begin{aligned}\mathbb{E}[(\theta_\lambda - \theta^*)(\theta_\lambda - \theta^*)^T] &\preceq (1-r)^{-4} I_\lambda(\theta^*)^{-1} \left(\mathbb{E}[s_\lambda(\theta^*)s_\lambda(\theta^*)^\top]\right) I_\lambda(\theta^*)^{-1} \\ &\preceq (1-r)^{-4} I_\lambda(\theta^*)^{-1} I_\lambda(\theta^*) I_\lambda(\theta^*)^{-1} \\ &= \frac{1}{(1-r)^4} I_\lambda(\theta^*)^{-1}\end{aligned}$$

This establishes the bound with constant $C = (1-r)^{-4}$, concluding the proof. □

B.2. Derivation of the Fisher Information Matrix for Multinomial Logistic Regression

The Fisher Information Matrix (FIM) quantifies the amount of information that an observable random variable carries about an unknown parameter θ upon which the probability of the random variable depends. Here, we derive the FIM for a single preference observation within a multinomial logistic regression model.

Consider a single observation where an expert chooses one item from a set of K items. Let $\mathbf{x}_q \in \mathbb{R}^d$ be the feature vector associated with item $q \in \{1, \ldots, K\}$. The probability of the expert choosing item q, given the parameter vector $\theta \in \mathbb{R}^d$, is modeled by the softmax function:

$$p_q(\theta) = P(\text{item } q \text{ is chosen} | \mathbf{x}_1, \ldots, \mathbf{x}_K, \theta) = \frac{\exp(\theta^\top \mathbf{x}_q)}{\sum_{q'=1}^K \exp(\theta^\top \mathbf{x}_{q'})}$$

Let y_q be an indicator variable such that $y_q = 1$ if item q is chosen, and $y_q = 0$ otherwise. Note that $\sum_{q=1}^K y_q = 1$. The log-likelihood for this single observation is:

$$\mathcal{L}(\theta) = \sum_{q=1}^K y_q \log p_q(\theta)$$

The score vector, which is the gradient of the log-likelihood with respect to θ, is:

$$S(\theta) = \nabla_\theta \mathcal{L}(\theta) = \sum_{q=1}^K y_q \frac{1}{p_q(\theta)} \nabla_\theta p_q(\theta)$$

The gradient of $p_q(\theta)$ is $\nabla_\theta p_q(\theta) = p_q(\theta)(\mathbf{x}_q - \bar{\mathbf{x}}(\theta))$, where $\bar{\mathbf{x}}(\theta) = \sum_{q'=1}^K p_{q'}(\theta)\mathbf{x}_{q'}$ is the expected feature vector under the current model. Substituting this into the score function:

$$S(\theta) = \sum_{q=1}^K y_q(\mathbf{x}_q - \bar{\mathbf{x}}(\theta)) = \left(\sum_{q=1}^K y_q \mathbf{x}_q\right) - \bar{\mathbf{x}}(\theta)$$

This can also be written as $S(\theta) = \sum_{q=1}^K (y_q - p_q(\theta))\mathbf{x}_q$.

The Hessian matrix $H(\theta)$ is the matrix of second derivatives of the log-likelihood: $H(\theta) = \nabla_\theta S(\theta)^\top$.

$$H(\theta) = \nabla_\theta \left(\sum_{q=1}^{K} y_q \mathbf{x}_q - \sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \right)^\top = -\nabla_\theta \left(\sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \right)^\top$$

Calculating the derivative:

$$\begin{aligned}
\nabla_\theta \left(\sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \right)^\top &= \sum_{q'=1}^{K} \left((\nabla_\theta p_{q'}(\theta)) \mathbf{x}_{q'}^\top + p_{q'}(\theta) \nabla_\theta \mathbf{x}_{q'}^\top \right) \\
&= \sum_{q'=1}^{K} p_{q'}(\theta)(\mathbf{x}_{q'} - \bar{\mathbf{x}}(\theta)) \mathbf{x}_{q'}^\top \quad (\text{since } \nabla_\theta \mathbf{x}_{q'}^\top = 0) \\
&= \sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \mathbf{x}_{q'}^\top - \left(\sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \right) \left(\sum_{q''=1}^{K} p_{q''}(\theta) \mathbf{x}_{q''} \right)^\top \\
&= \sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \mathbf{x}_{q'}^\top - \bar{\mathbf{x}}(\theta) \bar{\mathbf{x}}(\theta)^\top
\end{aligned}$$

So, the Hessian is:

$$H(\theta) = -\left(\sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \mathbf{x}_{q'}^\top - \bar{\mathbf{x}}(\theta) \bar{\mathbf{x}}(\theta)^\top \right)$$

The Fisher Information Matrix $I(\theta)$ for this single observation is defined as the negative expectation of the Hessian: $I(\theta) = -\mathbb{E}[H(\theta)]$. Since the Hessian $H(\theta)$ as derived here does not depend on the random outcome variables y_q (after simplification using properties of $p_q(\theta)$), the expectation does not change it. Thus:

$$I(\theta) = \sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \mathbf{x}_{q'}^\top - \bar{\mathbf{x}}(\theta) \bar{\mathbf{x}}(\theta)^\top$$

Expanding $\bar{\mathbf{x}}(\theta) = \sum_{q=1}^{K} p_q(\theta) \mathbf{x}_q$, the term $\bar{\mathbf{x}}(\theta) \bar{\mathbf{x}}(\theta)^\top$ becomes $\left(\sum_{q=1}^{K} p_q(\theta) \mathbf{x}_q \right) \left(\sum_{q'=1}^{K} p_{q'}(\theta) \mathbf{x}_{q'} \right)^\top = \sum_{q,q'} p_q(\theta) p_{q'}(\theta) \mathbf{x}_q \mathbf{x}_{q'}^\top$. Thus, we get the form:

$$I(\theta) = \sum_{q=1}^{K} p_q(\theta) \mathbf{x}_q \mathbf{x}_q^\top - \sum_{q,q'} p_q(\theta) p_{q'}(\theta) \mathbf{x}_q \mathbf{x}_{q'}^\top$$

This expression represents the FIM for one multinomial preference choice. If there are N independent such choices, the total FIM is the sum of the FIMs from each choice. This derivation provides the basis for the FIM expressions used in the subsequent experimental design.

B.3. Expected Fisher Information Objective for PBRL

Our goal is to select K policies, $\pi_{1:K} = (\pi_1, \ldots, \pi_K)$, to maximize information about the unknown parameter θ. A classical challenge in Optimal Experimental Design (OED) is that directly optimizing a discrete set of experiments (trajectories in our case) is often intractable (Pukelsheim, 2006; Fedorov & Hackl, 1997). A standard approach in OED is to instead optimize a design measure, which in our policy-based setting corresponds to optimizing over policies and considering the *expected* Fisher Information Matrix (FIM) they induce.

The total expected regularized FIM, $I_{reg}(\pi_{1:K}, \theta)$, for K policies $\pi_{1:K}$ generating T episodes of H steps each is:

$$I_{reg}(\pi_{1:K}, \theta) = T \sum_{h=1}^{H} I_h(\pi_{1:K}, \theta) + \lambda I_d$$

Here, $I_h(\pi_{1:K}, \theta)$ is the expected FIM contribution from timestep h of a single episode, averaged over the trajectory distributions η_{π_q} induced by each policy π_q. Let s_h^q be the state of trajectory $\tau_q \sim \eta_{\pi_q}$ at step h, and $p(q|h; \tau_{1..K})$ be the softmax probability of preferring state s_h^q from the set of K states $\{s_h^1, \dots, s_h^K\}$ presented at that step. Then $I_h(\pi_{1:K}, \theta)$ is: $I_h(\pi_{1:K}, \theta) = \mathbb{E}_{\substack{\tau_q \sim \eta_{\pi_q} \\ q \in [K]}} \left[\sum_{q=1}^K p(q|h; \tau_{1..K}) \phi(s_h^q) \phi(s_h^q)^\top - \sum_{q,q'} p(q|h; \tau_{1..K}) p(q'|h; \tau_{1..K}) \phi(s_h^q) \phi(s_h^{q'})^\top \right]$

The detailed FIM derivation for a single multinomial choice is in Appendix B.2.

The ideal experimental design objective is to choose policies $\pi_{1:K}$ to optimize a scalar criterion $s(\cdot)$ of this expected FIM (e.g., D- or A-optimality):

$$\arg\max_{\pi_{1:K}} s\left(I_{reg}(\pi_{1:K}, \theta)\right) \tag{4}$$

The challenges associated with this ideal objective are discussed in Section 4, and are addressed by the reformulation and approximation techniques detailed in the main text (Section 4.2) and expanded upon in Section B.4 below.

B.4. Reformulation to a Tractable Objective

This section provides the full derivation of the tractable experimental design objective discussed in Section 4.2.

Step 1: Reformulation using State Visitation Measures. The evaluation of $I_h(\pi_{1:K}, \theta)$ (defined in Section 4 based on an expectation over trajectories $\tau_q \sim \eta_{\pi_q}$) can be simplified. The term inside this expectation, which we denote $f_h(s_1, \dots, s_K; \theta)$, is given by:

$$f_h(s_1, \dots, s_K; \theta) = \left[\sum_{q=1}^K p(q|s_{1..K}) \phi(s_q) \phi(s_q)^\top - \sum_{q,q'=1}^K p(q|s_{1..K}) p(q'|s_{1..K}) \phi(s_q) \phi(s_{q'})^\top \right],$$

where $p(q|s_{1..K}) = \frac{\exp(\theta^\top \phi(s_q))}{\sum_{k=1}^K \exp(\theta^\top \phi(s_k))}$. This term f_h depends only on the states $(s_1, \dots, s_K)$ presented at timestep h. Invoking Lemma B.3, $I_h(\pi_{1:K}, \theta)$ can be directly rewritten using state visitation measures $d_{\pi_q}^h(s)$ (the probability policy π_q visits state s at step h). Let $d_{1:K}^h = (d_{\pi_1}^h, \dots, d_{\pi_K}^h)$. Then:

$$I_h(\pi_{1:K}, \theta) = \mathop{\mathbb{E}}_{\substack{s_q \sim d_{\pi_q}^h \\ q \in [K]}} \left[f_h(s_1, \dots, s_K; \theta) \right] \equiv I_h(d_{1:K}^h, \theta).$$

This equality $I_h(\pi_{1:K}, \theta) = I_h(d_{1:K}^h, \theta)$ signifies that the expected FIM contribution at step h, originally defined over policies, can be equivalently expressed in terms of the state visitation measures $d_{1:K}^h$ induced by those policies. This shifts the expectation from trajectory distributions to state visitation measures. While this simplifies the dependency, the expectation is still over $|\mathcal{S}|^K$ state tuples and depends on the unknown θ via f_h.

Step 2: Approximation for θ-Independence. When no prior information about user preferences is available, the only unbiased assumption is that each of the K presented options is equally likely to be chosen. This leads to the uniform probability assignment:

$$p(q|s_{1..K}) = \frac{1}{K}.$$

This assignment is standard for fixed design and serves as a natural starting point for adaptive strategies before any information about θ is gathered.

Substituting this into the expression for $I_h(d_{1:K}^h, \theta)$ (specifically, into $f_h(s_1, \dots, s_K; \theta)$ within the expectation) yields the approximate expected FIM contribution at timestep h, denoted $I_{approx,h}(d_{1:K}^h)$:

$$I_{approx,h}(d_{1:K}^h) = \mathop{\mathbb{E}}_{\substack{s_q \sim d_{\pi_q}^h \\ q \in [K]}} \left[\frac{1}{K} \sum_{q=1}^K \phi(s_q) \phi(s_q)^\top - \frac{1}{K^2} \sum_{q,q'=1}^K \phi(s_q) \phi(s_{q'})^\top \right] \tag{5}$$

Crucially, this approximate FIM, $I_{approx,h}(d_{1:K}^h)$, is now independent of θ. This step implies $I_h(d_{1:K}^h, \theta) \approx I_{approx,h}(d_{1:K}^h)$. However, computing this expectation still involves a sum over $|\mathcal{S}|^K$ terms if done naively.

Step 3: Marginalization for Tractability. The expression for $I_{approx,h}(d^h_{1:K})$ in Eq. 5 can be made computationally tractable. As established in Theorem B.4 (and its proof, which shows how the expectation is resolved), this expectation can be computed efficiently using the state visitation measures. Let $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ be the feature matrix (rows $\phi(s_j)^T$) and $d^h_q \in \mathbb{R}^{|\mathcal{S}|}$ be the state visitation vector for policy π_q at step h. Theorem B.4 provides the following tractable matrix form:

$$I_{approx,h}(d^h_{1:K}) = \Phi^T \left(\frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d^h_q) - \bar{d}^h (\bar{d}^h)^T \right) \Phi \tag{6}$$

where $\bar{d}^h = \frac{1}{K} \sum_{q=1}^{K} d^h_q$ is the average state visitation vector at step h.

This final expression $I_{approx,h}(d^h_{1:K})$ depends only on the state visitation vectors d^h_q and the known feature matrix Φ. It is independent of θ and computationally tractable.

Therefore, our practical experimental design objective becomes optimizing a scalar criterion $s(\cdot)$ applied to the total approximate expected regularized FIM, which is now expressed entirely in terms of the state visitation measures $d_{1:K} = \{d^h_q\}_{q \in [K], h \in [H]}$:

$$\arg\max_{d_{1:K}} s \left(T \cdot \sum_{h=1}^{H} I_{approx,h}(d^h_{1:K}) + \lambda I_d \right) \tag{7}$$

The optimization is subject to the constraints that these visitation measures are valid in the given MDP.

B.4.1. Information Decomposition and Policy Diversity

The tractable objective derived from Theorem B.4 provides valuable insight into what constitutes an informative experiment in the context of preference-based RL. Let's examine the core matrix term within the approximate FIM at timestep h:

$$M_h(d^h_{1:K}) = \frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d^h_q) - \bar{d}^h (\bar{d}^h)^T$$

This expression can be interpreted in terms of the statistics of the state visitation distributions. The first term, $\frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d^h_q)$, represents the average of the per-policy state variances (since $\mathrm{diag}(d^h_q)$ captures the variance if states were one-hot encoded). The second term, $\bar{d}^h (\bar{d}^h)^T$, represents the outer product of the *average* state visitation vector. The structure resembles the definition of a covariance matrix: $\mathbb{E}[xx^T] - \mathbb{E}[x]\,\mathbb{E}[x]^T$. Maximizing a scalar function of $I_{approx,h}(d^h_{1:K}) = \Phi^T M_h(d^h_{1:K}) \Phi$ intuitively encourages policies whose average state visitation behavior exhibits high variance or spread in the feature space, after accounting for the variance of the average distribution.

This suggests that the objective implicitly favors diversity among the chosen policies $\pi_1, \ldots, \pi_K$. If all policies induce very similar state visitation distributions ($d^h_q \approx \bar{d}^h$ for all q), the term $M_h(d^h_{1:K})$ might be small. Conversely, if the policies explore distinct regions of the state space, leading to diverse d^h_q vectors, the resulting $M_h(d^h_{1:K})$ is likely to be larger (in a matrix sense, e.g., larger eigenvalues), contributing more to the information gain.

This intuition is made precise by Lemma B.5, which provides an alternative decomposition of $I_{approx,h}(d^h_{1:K})$. Invoking this lemma, we can rewrite the approximate FIM contribution as:

$$\Phi^T \Bigg[\underbrace{\frac{1}{K} \sum_{q=1}^{K} \left(\mathrm{diag}(d^h_q) - d^h_q (d^h_q)^T \right)}_{\substack{\text{Average Per-Policy} \\ \text{State Covariance}}} + \underbrace{\frac{1}{K^2} \sum_{1 \le i < j \le K} (d^h_i - d^h_j)(d^h_i - d^h_j)^T}_{\substack{\text{Average Pairwise Difference} \\ \text{(Diversity Term)}}} \Bigg] \Phi$$

This decomposition elegantly separates the information contribution into two components.

Average Per-Policy State Covariance The first term, **Average Per-Policy State Covariance**, represents the average of the covariance matrices associated with each individual policy's state visitation distribution d^h_q. It captures the inherent uncertainty or spread within each policy's behavior at timestep h; maximizing this term encourages policies that individually explore diverse states within their own trajectories.

Average Pairwise Difference (Diversity Term) The second component, the **Average Pairwise Difference (Diversity Term)**, directly quantifies the diversity between the policies. It is a sum of outer products of the differences between the state visitation vectors of all unique pairs of policies (i, j). This term is explicitly maximized when the state visitation distributions d_i^h and d_j^h are significantly different from each other, thereby encouraging the selection of policies that explore distinct parts of the state space relative to one another.

Therefore, optimizing the approximate FIM objective naturally balances exploring broadly within each policy and ensuring that the set of policies collectively covers different aspects of the state space, maximizing the potential for informative comparisons.

B.5. Formal: Relationship between the State-based Feedback model and the Truncated Feedback model

We now formally analyze the relationship between the information content of the State-based feedback model and the Truncated Trajectory feedback model. This analysis is performed under the *perfect decomposition condition*, where the features of a truncated trajectory are assumed to be the sum of the features of its constituent states. Additionally, we utilize the uniform preference approximation ($p(q|s_{1..K}) \approx 1/K$) introduced in Step 2 of Section 4.2 (Eq. 5), which yields the following θ-independent structure for the approximated Fisher Information matrix component, $I_{approx,h}$, derived from comparing K feature vectors $\{\mathbf{x}_q\}_{q=1}^K$ at a given step h:

$$I_{approx,h}(\mathbf{x}_1, \dots, \mathbf{x}_K) = \frac{1}{K}\sum_{q=1}^{K} \mathbf{x}_q \mathbf{x}_q^\top - \frac{1}{K^2}\sum_{q,q'=1}^{K} \mathbf{x}_q \mathbf{x}_{q'}^\top$$

The following theorem compares the approximated FIMs of the two models under these conditions.

Theorem B.2 (Comparison of Approximated FIMs under Perfect Decomposition)**.** *Let $\mathcal{T}^K = \{(\tau_t^1, \dots, \tau_t^K)\}_{t=1}^T$ be a set of $T \times K$ trajectories. For the standard (state-based) feedback model, let $\phi(s_{t,h}^q)$ be the feature vector for the state $s_{t,h}^q$. The approximated Fisher Information Matrix is*

$$I_{approx}(\mathcal{T}^K) = \sum_{t=1}^{T}\sum_{h=1}^{H}\left(\frac{1}{K}\sum_{q=1}^{K}\phi(s_{t,h}^q)\phi(s_{t,h}^q)^\top - \frac{1}{K^2}\sum_{q,q'=1}^{K}\phi(s_{t,h}^q)\phi(s_{t,h}^{q'})^\top\right).$$

For the truncated trajectory feedback model, assume the perfect decomposition condition holds, such that the feature representation of the q-th trajectory in episode t truncated at timestep h is $\psi_{t,h}^q = \sum_{h'=1}^{h}\phi(s_{t,h'}^q)$. The corresponding approximated Fisher Information Matrix is

$$I_{approx}^{trunc}(\mathcal{T}^K) = \sum_{t=1}^{T}\sum_{h=1}^{H}\left(\frac{1}{K}\sum_{q=1}^{K}\psi_{t,h}^q(\psi_{t,h}^q)^\top - \frac{1}{K^2}\sum_{q,q'=1}^{K}\psi_{t,h}^q(\psi_{t,h}^{q'})^\top\right).$$

Then, under the perfect decomposition condition,

$$I_{approx}^{trunc}(\mathcal{T}^K) \succeq \frac{1}{4}\cdot I_{approx}(\mathcal{T}^K).$$

Proof of Theorem B.2. Let $\mathcal{T}^K = \{(\tau_t^1, \dots, \tau_t^K)\}_{t=1}^T$ be the set of trajectories. We define two approximated Fisher Information Matrices based on the uniform preference assumption ($p \approx 1/K$).

First, the FIM for the state-based feedback model, denoted $I_{\text{approx}}(\mathcal{T}^K)$, uses features $\phi(s_{t,h}^q)$ from individual states:

$$I_{\text{approx}}(\mathcal{T}^K) = \sum_{t=1}^{T}\sum_{h=1}^{H}\left(\frac{1}{K}\sum_{q=1}^{K}\phi(s_{t,h}^q)\phi(s_{t,h}^q)^\top - \frac{1}{K^2}\sum_{q,q'=1}^{K}\phi(s_{t,h}^q)\phi(s_{t,h}^{q'})^\top\right).$$

Next, the FIM for the truncated trajectory feedback model, $I_{\text{approx}}^{\text{trunc}}(\mathcal{T}^K)$, under the perfect decomposition condition, uses the sum-decomposed features $\psi_{t,h}^q = \sum_{h'=1}^{h}\phi(s_{t,h'}^q)$:

$$I_{\text{approx}}^{\text{trunc}}(\mathcal{T}^K) = \sum_{t=1}^{T}\sum_{h=1}^{H}\left(\frac{1}{K}\sum_{q=1}^{K}\psi_{t,h}^q(\psi_{t,h}^q)^\top - \frac{1}{K^2}\sum_{q,q'=1}^{K}\psi_{t,h}^q(\psi_{t,h}^{q'})^\top\right).$$

Our goal is to show that $I^{\text{trunc}}_{\text{approx}}(\mathcal{T}^K) \succeq C_H \cdot I_{\text{approx}}(\mathcal{T}^K)$ for some constant $C_H > 0$. Let $\mathbf{\Phi}_{t,q} \in \mathbb{R}^{H\times d}$ be the matrix whose h-th row is $\phi(s^q_{t,h})^\top$. That is,

$$\mathbf{\Phi}_{t,q} = \begin{pmatrix} \phi(s^q_{t,1})^\top \\ \phi(s^q_{t,2})^\top \\ \vdots \\ \phi(s^q_{t,H})^\top \end{pmatrix}.$$

The sum of outer products over the horizon H for the state-based model is $\sum_{h=1}^H \phi(s^q_{t,h})\phi(s^q_{t,h})^\top = \mathbf{\Phi}^\top_{t,q} I_H \mathbf{\Phi}_{t,q}$, where I_H is the $H \times H$ identity matrix. Similarly, the sum of cross-products is $\sum_{h=1}^H \phi(s^q_{t,h})\phi(s^{q'}_{t,h})^\top = \mathbf{\Phi}^\top_{t,q} I_H \mathbf{\Phi}_{t,q'}$. Thus, $I_{\text{approx}}(\mathcal{T}^K)$ can be rewritten as:

$$I_{\text{approx}}(\mathcal{T}^K) = \sum_{t=1}^T \left(\frac{1}{K} \sum_{q=1}^K \mathbf{\Phi}^\top_{t,q} I_H \mathbf{\Phi}_{t,q} - \frac{1}{K^2} \sum_{q,q'=1}^K \mathbf{\Phi}^\top_{t,q} I_H \mathbf{\Phi}_{t,q'} \right).$$

For the truncated trajectory model (under perfect decomposition), let $\mathbf{\Psi}_{t,q} \in \mathbb{R}^{H\times d}$ be the matrix whose h-th row is $(\psi^q_{t,h})^\top = \left(\sum_{h'=1}^h \phi(s^q_{t,h'})\right)^\top$. Let $S \in \mathbb{R}^{H\times H}$ be the lower triangular matrix of ones, i.e., $S_{ij} = 1$ if $i \geq j$ and $S_{ij} = 0$ if $i < j$. For example, if $H = 3$:

$$S = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}.$$

The cumulative sum structure means $\mathbf{\Psi}_{t,q} = S\mathbf{\Phi}_{t,q}$. The sum of outer products over the horizon H for the truncated model is $\sum_{h=1}^H \psi^q_{t,h}(\psi^q_{t,h})^\top = \mathbf{\Psi}^\top_{t,q}\mathbf{\Psi}_{t,q} = (S\mathbf{\Phi}_{t,q})^\top(S\mathbf{\Phi}_{t,q}) = \mathbf{\Phi}^\top_{t,q} S^\top S \mathbf{\Phi}_{t,q}$. Similarly, the sum of cross-products is $\sum_{h=1}^H \psi^q_{t,h}(\psi^{q'}_{t,h})^\top = \mathbf{\Psi}^\top_{t,q}\mathbf{\Psi}_{t,q'} = \mathbf{\Phi}^\top_{t,q} S^\top S \mathbf{\Phi}_{t,q'}$. Let $M = S^\top S$. This is an $H \times H$ symmetric positive definite matrix. For $H = 3$, $M = S^\top S = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 2 & 1 \\ 2 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$. Thus, $I^{\text{trunc}}_{\text{approx}}(\mathcal{T}^K)$ can be expressed in terms of $\mathbf{\Phi}_{t,q}$ and M:

$$I^{\text{trunc}}_{\text{approx}}(\mathcal{T}^K) = \sum_{t=1}^T \left(\frac{1}{K} \sum_{q=1}^K \mathbf{\Phi}^\top_{t,q} M \mathbf{\Phi}_{t,q} - \frac{1}{K^2} \sum_{q,q'=1}^K \mathbf{\Phi}^\top_{t,q} M \mathbf{\Phi}_{t,q'} \right).$$

Let $X_t = \begin{pmatrix} \mathbf{\Phi}^\top_{t,1} & \dots & \mathbf{\Phi}^\top_{t,K} \end{pmatrix}^\top \in \mathbb{R}^{(KH)\times d}$. Let $J_K = \frac{1}{K}\mathbf{1}_K\mathbf{1}_K^\top$ be the $K \times K$ matrix of all $1/K$. Let I_K be the $K \times K$ identity matrix. The FIM expressions can be written compactly using Kronecker products $\otimes$:

$$I_{\text{approx}}(\mathcal{T}^K) = \sum_{t=1}^T X_t^\top \left(\frac{1}{K}(I_K - J_K) \otimes I_H \right) X_t$$

$$I^{\text{trunc}}_{\text{approx}}(\mathcal{T}^K) = \sum_{t=1}^T X_t^\top \left(\frac{1}{K}(I_K - J_K) \otimes M \right) X_t$$

Since $M = S^\top S$ is positive definite (as S is invertible), its eigenvalues are positive. Let $\lambda_{min}(M)$ be the smallest eigenvalue of M. Then $M \succeq \lambda_{min}(M) I_H$. The matrix $I_K - J_K$ is positive semidefinite (it's proportional to a projection matrix). Therefore, using properties of Kronecker products and Loewner order:

$$(I_K - J_K) \otimes M \succeq (I_K - J_K) \otimes (\lambda_{min}(M) I_H) = \lambda_{min}(M)(I_K - J_K) \otimes I_H$$

Multiplying by $1/K$ and summing over t after pre- and post-multiplying by $X_t^\top$ and X_t:

$$\sum_{t=1}^T X_t^\top \left(\frac{1}{K}(I_K - J_K) \otimes M \right) X_t \succeq \lambda_{min}(M) \sum_{t=1}^T X_t^\top \left(\frac{1}{K}(I_K - J_K) \otimes I_H \right) X_t$$

This shows $I^{\text{trunc}}_{\text{approx}}(\mathcal{T}^K) \succeq \lambda_{min}(M) \cdot I_{\text{approx}}(\mathcal{T}^K)$. Let $C_H = \lambda_{min}(M)$. Since $M = S^\top S$ and S is invertible, M is positive definite, so its eigenvalues are positive, and $C_H > 0$. C_H depends only on H. The eigenvalues of M are $\lambda_k(M) = \frac{1}{4\sin^2\left(\frac{(2k-1)\pi}{2(2H+1)}\right)}$ for $k = 1, \ldots, H$. The minimum eigenvalue occurs at $k = H$:

$$C_H = \lambda_{min}(M) = \frac{1}{4\sin^2\left(\frac{(2H-1)\pi}{2(2H+1)}\right)}$$

Since $\sin^2(x) \leq 1$ for any x, we have a simple lower bound:

$$C_H = \frac{1}{4\sin^2\left(\frac{(2H-1)\pi}{2(2H+1)}\right)} \geq \frac{1}{4 \cdot 1} = \frac{1}{4}$$

Therefore, we can state the result using the constant lower bound $1/4$:

$$I^{\text{trunc}}_{\text{approx}}(\mathcal{T}^K) \succeq \frac{1}{4} \cdot I_{\text{approx}}(\mathcal{T}^K)$$

This completes the proof. □

B.6. Derivation of the Tractable Experimental Design Objective

We consider T episodes where in each episode we generate K parallel trajectories of horizon H. Let $\tau_t^q = (s_{t,1}^q, a_{t,1}^q, \ldots, s_{t,H}^q, a_{t,H}^q)$ be the trajectory for policy q in episode t. States are mapped to features via $\phi(s)$. We use $[K]$ to denote the set $\{1, 2, \ldots, K\}$.

The probability of the state $s_{t,h}^q$ from trajectory q being preferred at timestep h in episode t is given by:

$$p(q|t,h) = \frac{\exp(\theta^\top \phi(s_{t,h}^q))}{\sum_{q'=1}^K \exp(\theta^\top \phi(s_{t,h}^{q'}))}$$

The regularized log-likelihood function is:

$$L_{reg}(\theta) = \left(\sum_{t=1}^T \sum_{h=1}^H \sum_{q=1}^K y_{t,h,q} \log(p(q|t,h))\right) - \frac{\lambda}{2}\theta^\top\theta$$

where $y_{t,h,q} = 1$ if trajectory q was chosen at step (t,h) and 0 otherwise.

The regularized Fisher Information Matrix (FIM) for a specific set of $T \times K$ trajectories $\mathcal{T}^K = \{(\tau_t^1, \ldots, \tau_t^K)\}_{t=1}^T$ is:

$$I_{reg}(\mathcal{T}^K, \theta) = \left(\sum_{t=1}^T \sum_{h=1}^H \left[\sum_q p(q|t,h)\phi(s_{t,h}^q)\phi(s_{t,h}^q)^\top - \sum_{q,q'} p(q|t,h)p(q'|t,h)\phi(s_{t,h}^q)\phi(s_{t,h}^{q'})^\top\right]\right) + \lambda I_d$$

We relax the problem by choosing K policies $\pi_1, \ldots, \pi_K$. We assume that in each episode t, the K trajectories $(\tau_t^1, \ldots, \tau_t^K)$ are generated independently, with $\tau_t^q \sim \eta_{\pi_q}$, where $\eta_{\pi_q}(\tau)$ is the probability distribution over trajectories $\mathcal{T}$ induced by policy π_q. The joint probability of generating a specific tuple of K trajectories $(\tau_1, \ldots, \tau_K)$ in an episode is $\eta_{\pi_{1:K}}(\tau_1, \ldots, \tau_K) = \prod_{q=1}^K \eta_{\pi_q}(\tau_q)$. Episodes are independent.

We are interested in the expected Regularized FIM over the distribution of trajectories induced by these policies:

$$I_{reg}(\pi_{1:K}, \theta) = \mathop{\mathbb{E}}_{\mathcal{T}^K \sim \eta_{\pi_{1:K}}}[I_{reg}(\mathcal{T}^K, \theta)]$$

Since episodes are i.i.d., the expectation of the sum over t is T times the expectation for a single episode. Let $I_h(\pi_{1:K}, \theta)$ denote the expected FIM contribution at timestep h of a single episode. Let $(\tau_1, \ldots, \tau_K)$ denote the trajectories in a single

episode, where $\tau_q \sim \eta_{\pi_q}$. Let s_h^q be the state at timestep h in trajectory τ_q. Let $p(q|h;\tau_1..\tau_K) = \frac{\exp(\theta^\top \phi(s_h^q))}{\sum_{k=1}^K \exp(\theta^\top \phi(s_h^k))}$. Then

$$I_h(\pi_{1:K},\theta) = \mathop{\mathbb{E}}_{\substack{\tau_q\sim\eta_{\pi_q}\\ q\in[K]}}\left[\sum_q p(q|h;\tau_1..\tau_K)\phi(s_h^q)\phi(s_h^q)^\top - \sum_{q,q'} p(q|h;\tau_1..\tau_K)p(q'|h;\tau_1..\tau_K)\phi(s_h^q)\phi(s_h^{q'})^\top\right]$$

The total expected regularized FIM is then:

$$I_{reg}(\pi_{1:K},\theta) = T\cdot\sum_{h=1}^H I_h(\pi_{1:K},\theta) + \lambda I_d$$

Now, we want to express this expectation in terms of state visitation measures $d^h_{\pi_q}(s) = \sum_{\tau\in\mathcal{T}}\eta_{\pi_q}(\tau)\mathbb{I}_{\{s_h^q=s\}}$. We use a lemma to formalize the transition from trajectory expectations to state-visitation expectations for a single timestep.

Lemma B.3. *Let $\pi_1,\dots,\pi_K$ be policies with corresponding trajectory distributions $\eta_{\pi_1},\dots,\eta_{\pi_K}$ and state visitation measures $d^h_{\pi_1},\dots,d^h_{\pi_K}$. Let $f(s_1,\dots,s_K)$ be any function of a tuple of K states. Assume trajectories $\tau_1,\dots,\tau_K$ are drawn independently, $\tau_q\sim\eta_{\pi_q}$. Let s_h^q denote the state at timestep h of trajectory τ_q. Then for any fixed h:*

$$\mathop{\mathbb{E}}_{\substack{\tau_q\sim\eta_{\pi_q}\\ q\in[K]}}\left[f(s_h^1,\dots,s_h^K)\right] = \mathop{\mathbb{E}}_{\substack{s_q\sim d^h_{\pi_q}\\ q\in[K]}}\left[f(s_1,\dots,s_K)\right]$$

where the expectation on the right is taken with respect to states $s_1,\dots,s_K$ drawn independently from the respective state visitation measures at timestep h. The notation $s_q\sim d^h_{\pi_q}$ for $q\in[K]$ implies the joint draw $(s_1,\dots,s_K)$ is from the product distribution $\prod_{q=1}^K d^h_{\pi_q}$.

Proof of Lemma B.3. For a fixed h, we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{\tau_q\sim\eta_{\pi_q}\\ q\in[K]}}\left[f(s_h^1,\dots,s_h^K)\right] &= \sum_{\tau_1,\dots,\tau_K\in\mathcal{T}}\left(\prod_{q=1}^K\eta_{\pi_q}(\tau_q)\right)f(s_h^1,\dots,s_h^K)\\
&= \sum_{\tau_1,\dots,\tau_K\in\mathcal{T}}\left(\prod_{q=1}^K\eta_{\pi_q}(\tau_q)\right)\sum_{s_1,\dots,s_K\in\mathcal{S}}f(s_1,\dots,s_K)\prod_{q=1}^K\mathbb{I}_{\{s_q=s_h^q\}} \quad \text{(Introduce indicators)}\\
&= \sum_{s_1,\dots,s_K\in\mathcal{S}}f(s_1,\dots,s_K)\sum_{\tau_1,\dots,\tau_K\in\mathcal{T}}\left(\prod_{q=1}^K\eta_{\pi_q}(\tau_q)\mathbb{I}_{\{s_q=s_h^q\}}\right) \quad \text{(Rearrange sums)}\\
&= \sum_{s_1,\dots,s_K\in\mathcal{S}}f(s_1,\dots,s_K)\left(\prod_{q=1}^K\sum_{\tau_q\in\mathcal{T}}\eta_{\pi_q}(\tau_q)\mathbb{I}_{\{s_q=s_h^q\}}\right) \quad \text{(Factorize sum over } \tau)\\
&= \sum_{s_1,\dots,s_K\in\mathcal{S}}f(s_1,\dots,s_K)\left(\prod_{q=1}^K d^h_{\pi_q}(s_q)\right) \quad \text{(Definition of } d^h_{\pi_q})\\
&= \mathop{\mathbb{E}}_{\substack{s_q\sim d^h_{\pi_q}\\ q\in[K]}}\left[f(s_1,\dots,s_K)\right] \quad \text{(Definition of expectation w.r.t. product measure)}
\end{aligned}$$

This completes the proof. □

Using Lemma B.3, we can rewrite the per-timestep expected FIM $I_h(\pi_{1:K},\theta)$. Let $f_h(s_1,\dots,s_K;\theta)$ be the term inside the expectation defining $I_h(\pi_{1:K},\theta)$:

$$f_h(s_1,\dots,s_K;\theta) = \left[\sum_q p(q|s_{1..K})\phi(s_q)\phi(s_q)^\top - \sum_{q,q'}p(q|s_{1..K})p(q'|s_{1..K})\phi(s_q)\phi(s_{q'})^\top\right]$$

where $p(q|s_{1..K}) = \frac{\exp(\theta^\top \phi(s_q))}{\sum_{k=1}^K \exp(\theta^\top \phi(s_k))}$. Let $s_{1:K} = (s_1, \ldots, s_K)$. Applying the lemma:

$$\begin{aligned} I_h(\pi_{1:K}, \theta) &= \mathop{\mathbb{E}}_{\substack{s_q \sim d^h_{\pi_q} \\ q \in [K]}} [f_h(s_1, \ldots, s_K; \theta)] \\ &= \sum_{s_1, \ldots, s_K \in \mathcal{S}} \left(\prod_{i=1}^K d^h_{\pi_i}(s_i) \right) f_h(s_1, \ldots, s_K; \theta) \end{aligned}$$

Theorem B.4. *Let d^h_q (representing $d^h_{\pi_q}$) be the state visitation measure for policy π_q at step h, for $q \in [K]$. Assume the preference probabilities are approximated as uniform, $p(q|s_{1..K}) \approx 1/K$. Let $I_{approx,h}(d^h_{1:K})$ be the approximate expected FIM contribution at timestep h for the set of state visitation measures $d^h_{1:K} = (d^h_1, \ldots, d^h_K)$, given by*

$$I_{approx,h}(d^h_{1:K}) = \mathop{\mathbb{E}}_{\substack{s_q \sim d^h_q \\ q \in [K]}} \left[\frac{1}{K} \sum_{q=1}^K \phi(s_q)\phi(s_q)^\top - \frac{1}{K^2} \sum_{q,q'=1}^K \phi(s_q)\phi(s_{q'})^\top \right]$$

Then $I_{approx,h}(d^h_{1:K})$ can be rewritten using these state visitation measures as:

$$I_{approx,h}(d^h_{1:K}) = \frac{1}{K} \sum_{q=1}^K \sum_{s \in \mathcal{S}} d^h_q(s)\phi(s)\phi(s)^\top - \frac{1}{K^2} \sum_{q,q'=1}^K \left(\sum_{s \in \mathcal{S}} d^h_q(s)\phi(s) \right) \left(\sum_{s' \in \mathcal{S}} d^h_{q'}(s')\phi(s')^\top \right) \tag{8}$$

Furthermore, let the state space $\mathcal{S}$ be ordered $\{s_1, \ldots, s_{|\mathcal{S}|}\}$. Let $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ be the feature matrix where the j-th row is $\phi(s_j)^T$. Let $d^h_q \in \mathbb{R}^{|\mathcal{S}|}$ be the state visitation vector (with $(d^h_q)_j = d^h_q(s_j)$). Let $\bar{d}^h = \frac{1}{K} \sum_{q=1}^K d^h_q$ be the average state visitation vector at step h. Then,

$$I_{approx,h}(d^h_{1:K}) = \Phi^T \left(\frac{1}{K} \sum_{q=1}^K \mathrm{diag}(d^h_q) - \bar{d}^h (\bar{d}^h)^T \right) \Phi \tag{9}$$

where $\mathrm{diag}(v)$ is the diagonal matrix with vector v on the diagonal.

Proof of Theorem B.4. We start with the definition of $I_{approx,h}(d^h_{1:K})$ under the uniform approximation:

$$I_{approx,h}(d^h_{1:K}) = \sum_{s_1, \ldots, s_K \in \mathcal{S}} \left(\prod_{i=1}^K d^h_i(s_i) \right) \left[\frac{1}{K} \sum_{q=1}^K \phi(s_q)\phi(s_q)^\top - \frac{1}{K^2} \sum_{q,q'=1}^K \phi(s_q)\phi(s_{q'})^\top \right]$$

Let $A_h = \sum_{s_1, \ldots, s_K \in \mathcal{S}} \left(\prod_{i=1}^K d^h_i(s_i) \right) \left[\frac{1}{K} \sum_{q=1}^K \phi(s_q)\phi(s_q)^\top \right]$ and $B_h = \sum_{s_1, \ldots, s_K \in \mathcal{S}} \left(\prod_{i=1}^K d^h_i(s_i) \right) \left[\frac{1}{K^2} \sum_{q,q'=1}^K \phi(s_q)\phi(s_{q'})^\top \right]$. Then $I_{approx,h}(d^h_{1:K}) = A_h - B_h$.

Compute A_h:

$$\begin{aligned} A_h &= \frac{1}{K} \sum_{q=1}^K \sum_{s_1, \ldots, s_K \in \mathcal{S}} \left(\prod_{i=1}^K d^h_{\pi_i}(s_i) \right) \phi(s_q)\phi(s_q)^\top \\ &= \frac{1}{K} \sum_{q=1}^K \left(\sum_{s_q \in \mathcal{S}} d^h_{\pi_q}(s_q)\phi(s_q)\phi(s_q)^\top \prod_{i \neq q} \sum_{s_i \in \mathcal{S}} d^h_{\pi_i}(s_i) \right) \quad \text{(Marginalizing)} \\ &= \frac{1}{K} \sum_{q=1}^K \sum_{s \in \mathcal{S}} d^h_{\pi_q}(s)\phi(s)\phi(s)^\top \quad \text{(since } \sum_{s_i} d^h_{\pi_i}(s_i) = 1\text{)} \end{aligned}$$

Compute B_h:

$$\begin{aligned}
B_h &= \frac{1}{K^2}\sum_{q,q'=1}^{K}\sum_{s_1,\ldots,s_K\in\mathcal{S}}\left(\prod_{i=1}^{K} d^h_{\pi_i}(s_i)\right)\phi(s_q)\phi(s_{q'})^\top \\
&= \frac{1}{K^2}\sum_{q,q'=1}^{K}\left(\sum_{s_q,s_{q'}\in\mathcal{S}} d^h_{\pi_q}(s_q)d^h_{\pi_{q'}}(s_{q'})\phi(s_q)\phi(s_{q'})^\top \prod_{i\neq q,q'}\sum_{s_i\in\mathcal{S}} d^h_{\pi_i}(s_i)\right) \quad \text{(Marginalizing)} \\
&= \frac{1}{K^2}\sum_{q,q'=1}^{K}\left(\sum_{s_q\in\mathcal{S}} d^h_{\pi_q}(s_q)\phi(s_q)\right)\left(\sum_{s_{q'}\in\mathcal{S}} d^h_{\pi_{q'}}(s_{q'})\phi(s_{q'})^\top\right) \\
&= \frac{1}{K^2}\sum_{q,q'=1}^{K}\left(\sum_{s\in\mathcal{S}} d^h_{\pi_q}(s)\phi(s)\right)\left(\sum_{s'\in\mathcal{S}} d^h_{\pi_{q'}}(s')\phi(s')^\top\right)
\end{aligned}$$

Combining these yields the first result (8):

$$I_{approx,h}(d^h_{1:K}) = A_h - B_h = \frac{1}{K}\sum_{q=1}^{K}\sum_{s\in\mathcal{S}} d^h_q(s)\phi(s)\phi(s)^\top - \frac{1}{K^2}\sum_{q,q'=1}^{K}\left(\sum_s d^h_q(s)\phi(s)\right)\left(\sum_{s'} d^h_{q'}(s')\phi(s')^\top\right)$$

Now, we express this in matrix form. Let $\Phi \in \mathbb{R}^{|\mathcal{S}|\times d}$ have rows $\phi(s_j)^T$, and let $d^h_q \in \mathbb{R}^{|\mathcal{S}|}$ (with entries $(d^h_q)_j = d^h_q(s_j)$). The first term A_h can be written as:

$$\begin{aligned}
A_h &= \frac{1}{K}\sum_{q=1}^{K}\sum_{j=1}^{|\mathcal{S}|}(d^h_q)_j\phi(s_j)\phi(s_j)^T \\
&= \frac{1}{K}\sum_{q=1}^{K}\Phi^T\mathrm{diag}(d^h_q)\Phi \\
&= \Phi^T\left(\frac{1}{K}\sum_{q=1}^{K}\mathrm{diag}(d^h_q)\right)\Phi
\end{aligned}$$

The second term B_h can be written as:

$$\begin{aligned}
B_h &= \frac{1}{K^2}\sum_{q,q'=1}^{K}\left(\sum_{j=1}^{|\mathcal{S}|}(d^h_q)_j\phi(s_j)\right)\left(\sum_{k=1}^{|\mathcal{S}|}(d^h_{q'})_k\phi(s_k)^\top\right) \\
&= \frac{1}{K^2}\sum_{q,q'=1}^{K}(\Phi^T d^h_q)(\Phi^T d^h_{q'})^T \\
&= \frac{1}{K^2}\sum_{q,q'=1}^{K}(\Phi^T d^h_q)(d^h_{q'})^T\Phi \\
&= \Phi^T\left(\frac{1}{K^2}\sum_{q=1}^{K} d^h_q\sum_{q'=1}^{K}(d^h_{q'})^T\right)\Phi \\
&= \Phi^T\left(\left(\frac{1}{K}\sum_{q=1}^{K} d^h_q\right)\left(\frac{1}{K}\sum_{q'=1}^{K} d^h_{q'}\right)^T\right)\Phi \\
&= \Phi^T\bar{d}^h(\bar{d}^h)^T\Phi
\end{aligned}$$

where $\bar{d}^h = \frac{1}{K}\sum_{q=1}^{K} d_q^h$. Substituting the matrix forms for A_h and B_h into $I_{approx,h}(d_{1:K}^h) = A_h - B_h$ yields the second result (9):

$$I_{approx,h}(d_{1:K}^h) = \Phi^T \left(\frac{1}{K}\sum_{q=1}^{K} \mathrm{diag}(d_q^h) \right) \Phi - \Phi^T \bar{d}^h (\bar{d}^h)^T \Phi = \Phi^T \left(\frac{1}{K}\sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \bar{d}^h (\bar{d}^h)^T \right) \Phi$$

This completes the proof. □

The final optimization objective, using this approximate expected FIM, becomes optimizing over the state visitation measures $d_{1:K} = \{d_q^h\}_{q\in[K],h\in[H]}$:

$$\arg\max_{d_{1:K}} s \left(T \cdot \sum_{h=1}^{H} I_{approx,h}(d_{1:K}^h) + \lambda I_d \right)$$

where $I_{approx,h}(d_{1:K}^h)$ is given by the expression(s) in Theorem B.4, and the optimization is subject to d_q^h being valid state visitation measures.

B.7. Information Decomposition and Policy Diversity

The tractable objective derived from Theorem B.4 provides valuable insight into what constitutes an informative experiment in the context of preference-based RL. Let's examine the core matrix term within the approximate FIM at timestep h:

$$M_h(d_{1:K}^h) = \frac{1}{K}\sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \bar{d}^h (\bar{d}^h)^T$$

This expression can be interpreted in terms of the statistics of the state visitation distributions. The first term, $\frac{1}{K}\sum_{q=1}^{K} \mathrm{diag}(d_q^h)$, represents the average of the per-policy state variances (since $\mathrm{diag}(d_q^h)$ captures the variance if states were one-hot encoded). The second term, $\bar{d}^h(\bar{d}^h)^T$, represents the outer product of the *average* state visitation vector. The structure resembles the definition of a covariance matrix: $\mathbb{E}[xx^T] - \mathbb{E}[x]\,\mathbb{E}[x]^T$. Maximizing a scalar function of $I_{approx,h}(d_{1:K}^h) = \Phi^T M_h(d_{1:K}^h)\Phi$ intuitively encourages policies whose average state visitation behavior exhibits high variance or spread in the feature space, after accounting for the variance of the average distribution.

This suggests that the objective implicitly favors diversity among the chosen policies $\pi_1, \ldots, \pi_K$. If all policies induce very similar state visitation distributions ($d_q^h \approx \bar{d}^h$ for all q), the term $M_h(d_{1:K}^h)$ might be small. Conversely, if the policies explore distinct regions of the state space, leading to diverse d_q^h vectors, the resulting $M_h(d_{1:K}^h)$ is likely to be larger (in a matrix sense, e.g., larger eigenvalues), contributing more to the information gain.

This intuition is made precise by Lemma B.5, which provides an alternative decomposition of $I_{approx,h}(d_{1:K}^h)$. Invoking this lemma, we can rewrite the approximate FIM contribution as:

$$\Phi^T \Bigg[\underbrace{\frac{1}{K}\sum_{q=1}^{K} \left(\mathrm{diag}(d_q^h) - d_q^h (d_q^h)^T \right)}_{\text{Average Per-Policy State Covariance}} + \underbrace{\frac{1}{K^2} \sum_{1\le i<j\le K} (d_i^h - d_j^h)(d_i^h - d_j^h)^T}_{\text{Average Pairwise Difference (Diversity Term)}} \Bigg] \Phi$$

This decomposition elegantly separates the information contribution into two components.

Average Per-Policy State Covariance The first term, **Average Per-Policy State Covariance**, represents the average of the covariance matrices associated with each individual policy's state visitation distribution d_q^h. It captures the inherent uncertainty or spread within each policy's behavior at timestep h; maximizing this term encourages policies that individually explore diverse states within their own trajectories.

Average Pairwise Difference (Diversity Term) The second component, the **Average Pairwise Difference (Diversity Term)**, directly quantifies the diversity between the policies. It is a sum of outer products of the differences between the state visitation vectors of all unique pairs of policies (i, j). This term is explicitly maximized when the state visitation

distributions d_i^h and d_j^h are significantly different from each other, thereby encouraging the selection of policies that explore distinct parts of the state space relative to one another.

Therefore, optimizing the approximate FIM objective naturally balances exploring broadly within each policy and ensuring that the set of policies collectively covers different aspects of the state space, maximizing the potential for informative comparisons.

Lemma B.5. *Let $I_{approx,h}(d_{1:K}^h)$ be the approximate expected Fisher Information Matrix contribution at timestep h under the uniform preference assumption, as given in Theorem B.4:*

$$I_{approx,h}(d_{1:K}^h) = \Phi^T \left(\frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \bar{d}^h (\bar{d}^h)^T \right) \Phi$$

where $d_q^h \in \mathbb{R}^{|\mathcal{S}|}$ is the state visitation vector for policy π_q at step h, $\Phi \in \mathbb{R}^{|\mathcal{S}| \times d}$ is the feature matrix, and $\bar{d}^h = \frac{1}{K} \sum_{q=1}^{K} d_q^h$. This can be rewritten in terms of pairwise differences between state visitation vectors as:

$$I_{approx,h}(d_{1:K}^h) = \Phi^T \left[\frac{1}{K} \sum_{q=1}^{K} \left(\mathrm{diag}(d_q^h) - d_q^h (d_q^h)^T \right) + \frac{1}{K^2} \sum_{1 \le i < j \le K} (d_i^h - d_j^h)(d_i^h - d_j^h)^T \right] \Phi$$

Proof. We begin with the definition from Theorem B.4. Let $M_h(d_{1:K}^h)$ denote the matrix expression within $\Phi^T(\dots)\Phi$:

$$M_h(d_{1:K}^h) = \frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \bar{d}^h (\bar{d}^h)^T$$

Expand the outer product of the average state visitation vector:

$$\bar{d}^h (\bar{d}^h)^T = \left(\frac{1}{K} \sum_{i=1}^{K} d_i^h \right) \left(\frac{1}{K} \sum_{j=1}^{K} d_j^h \right)^T = \frac{1}{K^2} \sum_{i=1}^{K} \sum_{j=1}^{K} d_i^h (d_j^h)^T$$

Substitute this into the expression for $M_h(d_{1:K}^h)$:

$$M_h(d_{1:K}^h) = \frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \frac{1}{K^2} \sum_{i=1}^{K} \sum_{j=1}^{K} d_i^h (d_j^h)^T$$

We split the double summation based on whether the indices are equal ($i = j$) or distinct ($i \neq j$):

$$\sum_{i=1}^{K} \sum_{j=1}^{K} d_i^h (d_j^h)^T = \sum_{q=1}^{K} d_q^h (d_q^h)^T + \sum_{i \neq j} d_i^h (d_j^h)^T$$

Substituting this yields:

$$M_h(d_{1:K}^h) = \frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \frac{1}{K^2} \sum_{q=1}^{K} d_q^h (d_q^h)^T - \frac{1}{K^2} \sum_{i \neq j} d_i^h (d_j^h)^T$$

By adding and subtracting the term $(K-1)\frac{1}{K^2} \sum_{q=1}^{K} d_q^h (d_q^h)^T$:

$$\begin{aligned} M_h(d_{1:K}^h) = & \frac{1}{K} \sum_{q=1}^{K} \mathrm{diag}(d_q^h) - \frac{1}{K^2} \sum_{q=1}^{K} d_q^h (d_q^h)^T - (K-1) \frac{1}{K^2} \sum_{q=1}^{K} d_q^h (d_q^h)^T \\ & + (K-1) \frac{1}{K^2} \sum_{q=1}^{K} d_q^h (d_q^h)^T - \frac{1}{K^2} \sum_{i \neq j} d_i^h (d_j^h)^T \end{aligned}$$

Combine the terms containing $\sum_{q=1}^{K} d_q^h (d_q^h)^T$:

$$\begin{aligned}
M_h(d_{1:K}^h) &= \frac{1}{K}\sum_{q=1}^{K}\operatorname{diag}(d_q^h) - (1+K-1)\frac{1}{K^2}\sum_{q=1}^{K} d_q^h (d_q^h)^T \\
&\quad + \frac{K-1}{K^2}\sum_{q=1}^{K} d_q^h (d_q^h)^T - \frac{1}{K^2}\sum_{i\neq j} d_i^h (d_j^h)^T \\
&= \frac{1}{K}\sum_{q=1}^{K}\operatorname{diag}(d_q^h) - \frac{K}{K^2}\sum_{q=1}^{K} d_q^h (d_q^h)^T \\
&\quad + \frac{K-1}{K^2}\sum_{q=1}^{K} d_q^h (d_q^h)^T - \frac{1}{K^2}\sum_{i\neq j} d_i^h (d_j^h)^T \\
&= \left(\frac{1}{K}\sum_{q=1}^{K}\operatorname{diag}(d_q^h) - \frac{1}{K}\sum_{q=1}^{K} d_q^h (d_q^h)^T\right) \\
&\quad + \left(\frac{K-1}{K^2}\sum_{q=1}^{K} d_q^h (d_q^h)^T - \frac{1}{K^2}\sum_{i\neq j} d_i^h (d_j^h)^T\right)
\end{aligned}$$

Consider the sum of outer products of pairwise differences over unique pairs $\{i, j\}$ such that $1 \le i < j \le K$:

$$\begin{aligned}
\sum_{1\le i<j\le K}(d_i^h - d_j^h)(d_i^h - d_j^h)^T &= \sum_{i<j}\left(d_i^h (d_i^h)^T - d_i^h (d_j^h)^T - d_j^h (d_i^h)^T + d_j^h (d_j^h)^T\right) \\
&= (K-1)\sum_{q=1}^{K} d_q^h (d_q^h)^T - \sum_{i<j}\left(d_i^h (d_j^h)^T + d_j^h (d_i^h)^T\right)
\end{aligned}$$

The second term $\sum_{i<j}(d_i^h (d_j^h)^T + d_j^h (d_i^h)^T)$ sums over all distinct pairs $\{i, j\}$, equivalent to the summation $\sum_{i\neq j} d_i^h (d_j^h)^T$. Thus,

$$\sum_{1\le i<j\le K}(d_i^h - d_j^h)(d_i^h - d_j^h)^T = (K-1)\sum_{q=1}^{K} d_q^h (d_q^h)^T - \sum_{i\neq j} d_i^h (d_j^h)^T$$

Dividing by K^2 yields:

$$\frac{1}{K^2}\sum_{1\le i<j\le K}(d_i^h - d_j^h)(d_i^h - d_j^h)^T = \frac{K-1}{K^2}\sum_{q=1}^{K} d_q^h (d_q^h)^T - \frac{1}{K^2}\sum_{i\neq j} d_i^h (d_j^h)^T$$

This exactly matches the second grouped term derived for $M_h(d_{1:K}^h)$. Substituting this structure back gives:

$$\begin{aligned}
M_h(d_{1:K}^h) &= \left(\frac{1}{K}\sum_{q=1}^{K}\operatorname{diag}(d_q^h) - \frac{1}{K}\sum_{q=1}^{K} d_q^h (d_q^h)^T\right) + \frac{1}{K^2}\sum_{1\le i<j\le K}(d_i^h - d_j^h)(d_i^h - d_j^h)^T \\
&= \frac{1}{K}\sum_{q=1}^{K}\left(\operatorname{diag}(d_q^h) - d_q^h (d_q^h)^T\right) + \frac{1}{K^2}\sum_{1\le i<j\le K}(d_i^h - d_j^h)(d_i^h - d_j^h)^T
\end{aligned}$$

Finally, reintroducing the outer feature matrix multiplication provides the desired result:

$$I_{approx,h}(d_{1:K}^h) = \Phi^T M_h(d_{1:K}^h)\Phi = \Phi^T\left[\frac{1}{K}\sum_{q=1}^{K}\left(\operatorname{diag}(d_q^h) - d_q^h (d_q^h)^T\right) + \frac{1}{K^2}\sum_{1\le i<j\le K}(d_i^h - d_j^h)(d_i^h - d_j^h)^T\right]\Phi$$

□

B.8. Detailed Algorithm Description

Our Experimental Design for Preference-Based Reinforcement Learning (ED-PBRL) algorithm, detailed in Algorithm 2, consists of two main phases. The first phase optimizes a set of K policies using Convex-RL according to our objective derived in Section 4.2. The second phase plays these optimized policies to collect K sets of trajectories for obtaining preferences.

Algorithm 2 ED-PBRL using Convex-RL (Detailed Version of Algorithm 1)

Input: Known MDP components $M = (\mathcal{S}, \mathcal{A}, P, H, \rho)$, number of policies K, number of episodes T, feature map Φ, scalar criterion $s(\cdot)$, number of optimization iterations N, regularization constant λ ($\lambda > 0$)
Output: Estimated preference parameter $\hat{\theta}$

Phase 1: Compute Optimal State Visitation Measures {Solve Eq. 3}

Initialize ${}^{(1)}d_{\text{mix},q}^{\{1,\dots,H\}} \leftarrow \mathbf{0}$ for $q = 1, \dots, K$ {Initialize visitation measures}
for $n = 1$ **to** $N - 1$ **do**
  Let $I_{total}^{(n)} = T \sum_{h=1}^{H} I_{approx,h}({}^{(n)}d_{\text{mix},1:K}^{h}) + \lambda I_d$ {Objective using ${}^{(n)}d_{\text{mix}}$}
  for $q = 1$ **to** K **do**
    Compute gradient reward: $r_{\text{grad}_q}(h, s, a) \leftarrow \nabla_{d_{\pi_q}^h(s,a)} s(I_{total}^{(n)})$
    Find policy maximizing linear objective: $\pi_{\text{grad}_q}^{(n)} \leftarrow \texttt{value\_iteration}(M, r_{\text{grad}_q})$
    Compute corresponding visitation vector $d_{\text{grad}_q}^{(n),\{1,\dots,H\}}$ from $\pi_{\text{grad}_q}^{(n)}$
  end for
  Determine step size α_n via line search: For $q = 1, \dots, K$, let $d_{\text{cand},q}^{h}(\alpha') = (1 - \alpha')^{(n)}d_{\text{mix},q}^{h} + \alpha' d_{\text{grad},q}^{(n),h}$. Find $\alpha_n \leftarrow \arg\max_{\alpha' \in [0,1]} s\left(T \sum_{h=1}^{H} I_{approx,h}(d_{\text{cand},1:K}^{h}(\alpha')) + \lambda I_d\right)$ (see Eq. 6 for $I_{approx,h}$)
  for $q = 1$ **to** K **do**
    ${}^{(n+1)}d_{\text{mix},q}^{\{1,\dots,H\}} \leftarrow (1 - \alpha_n) \cdot {}^{(n)}d_{\text{mix},q}^{\{1,\dots,H\}} + \alpha_n \cdot d_{\text{grad}_q}^{(n),\{1,\dots,H\}}$
  end for
end for
Let $\{d_{\text{mix},q}^{*h}\}_{h,q} \leftarrow \{{}^{(N)}d_{\text{mix},q}^{\{1,\dots,H\}}\}_{h,q}$ be the final optimal visitation measures.

Phase 2: Policy Extraction and Trajectory Sampling

for $q = 1$ **to** K **do**
  Extract policy π_q^* from final visitation measure $d_{\text{mix},q}^{*h}$
  $\mathcal{T}_q \leftarrow \emptyset$ {Initialize trajectory set for policy π_q^*}
end for
for $t = 1$ **to** T **do**
  for $q = 1$ **to** K **do**
    Sample trajectory $\tau_t^q \sim \pi_q^*$
    $\mathcal{T}_q \leftarrow \mathcal{T}_q \cup \{\tau_t^q\}$
  end for
end for
Let $\mathcal{D}_{feedback} = \{\mathcal{T}_q\}_{q=1}^{K}$ be the collected trajectories.

Phase 3: Parameter Estimation

Collect preference feedback for trajectories in $\mathcal{D}_{feedback}$.
Estimate $\hat{\theta}$ using all collected feedback (cf. Section 3 for estimation equation).
return $\hat{\theta}$

Phase 1: Compute Optimal State Visitation Measures This phase adapts the Frank-Wolfe algorithm (Frank & Wolfe, 1956) to maximize the objective $s(I_{total}(\pi_{1:K}))$. Here, $I_{total}(\pi_{1:K})$ represents the total approximate expected regularized FIM (the matrix argument of $s(\cdot)$ in Eq. 3), expressed in terms of policy-induced visitation measures. This is achieved by iteratively building state-action visitation measures $\{{}^{(n)}d_{\text{mix},q}^{h}\}$ corresponding to conceptual mixture policies. The process starts with ${}^{(1)}d_{\text{mix},q} = \mathbf{0}$.

Each iteration n of this phase involves these main steps:

1. **Gradient Computation**: The gradient of $s(I_{total})$ (using the current ${}^{(n)}d_{\text{mix},q}$) defines a reward function r_{grad_q} for each policy q.
2. **Policy Search Oracle**: For each q, a new base policy $\pi^{(n)}_{\text{grad}_q}$ is found by maximizing the expected cumulative reward r_{grad_q} (e.g., via value iteration). Its visitation measure $d^{(n)}_{\text{grad}_q}$ is computed.
3. **Line Search for Step Size**: The optimal step size α_n is determined by maximizing $s(\cdot)$ for the candidate mixture $(1-\alpha_n){}^{(n)}d_{\text{mix},q} + \alpha_n d^{(n)}_{\text{grad}_q}$.
4. **Mixture Update**: The next mixture's visitation measure is constructed: ${}^{(n+1)}d_{\text{mix},q} \leftarrow (1-\alpha_n){}^{(n)}d_{\text{mix},q} + \alpha_n d^{(n)}_{\text{grad}_q}$. This efficiently computes the visitation measure of the new conceptual mixture policy $\pi^{(n)}_{\text{mix},q}$.

This iterative process converges to the globally optimal visitation measures $\{d^{*h}_{\text{mix},q}\}$ due to the concavity of $s(\cdot)$ and the convexity of the feasible set of visitation measures.

Phase 2: Policy Extraction and Trajectory Sampling Upon convergence of Phase 1 after $N-1$ iterations, the final policies $\{\pi^*_q\}_{q=1}^K$ are extracted from the resulting state-action visitation measures $\{d^{*h}_{\text{mix},q}\}_{q=1}^K$. These policies are then executed to generate the $K \times T$ trajectories, which form the dataset $\mathcal{D}_{feedback}$ for collecting user preference feedback.

Phase 3: Parameter Estimation After the trajectories are generated and collected into $\mathcal{D}_{feedback}$ in Phase 2, preference feedback is obtained from the user for these trajectories. This accumulated feedback is then used to compute the final estimate $\hat{\theta}$ of the true reward parameter θ, as detailed in Section 3.

B.9. Detailed Optimization Guarantees

The Convex-RL optimization phase (Algorithm 2, lines 13-24) employs the Frank-Wolfe algorithm (also known as the conditional gradient method) over the convex polytope of valid state-action visitation measures (Puterman, 2014; Frank & Wolfe, 1956; Jaggi, 2013). The inclusion of an exact line search for the step size α_n is a standard variant of the Frank-Wolfe algorithm.

The key to guaranteeing global optimality for this procedure is the concavity of the objective function. Let $D = \{d^h_q\}_{h\in[H],q\in[K]}$ represent the collection of all state visitation vectors, where each $d^h_q \in \Delta^{|\mathcal{S}|-1}$ (the probability simplex over states). The domain of D is a convex set. The objective function is $f(D) = s(I_{total}(D))$, where $I_{total}(D)$ is precisely the matrix argument of $s(\cdot)$ in Eq. 3:

$$I_{total}(D) = T\sum_{h=1}^{H}\left[\Phi^T\left(\frac{1}{K}\sum_{q=1}^{K}\text{diag}(d^h_q) - \left(\frac{1}{K}\sum_{q=1}^{K}d^h_q\right)\left(\frac{1}{K}\sum_{q=1}^{K}d^h_q\right)^T\right)\Phi\right] + \lambda I_d.$$

With the concavity of the objective function established (Theorem 5.1), we can state the convergence guarantee for Algorithm 2 (Theorem B.6), which implements the Frank-Wolfe method.

B.9.1. Proof of Objective Function Concavity (Theorem 5.1)

Theorem 5.1. *[Concavity of the Objective Function] Assume the scalar criterion $s : \mathbb{S}^d_+ \to \mathbb{R}$ is concave and matrix-monotone non-decreasing. Then the objective function $f(D) = s(I_{total}(D))$, where $I_{total}(D)$ is the total approximate expected regularized FIM (the matrix argument of $s(\cdot)$ in Eq. 3), is concave with respect to the collection of state visitation vectors $D = \{d^h_q\}_{h\in[H],q\in[K]}$.*

Proof. Let $D = \{d^h_q\}_{h\in[H],q\in[K]}$ be the collection of state visitation vectors, where each $d^h_q \in \Delta^{|\mathcal{S}|-1}$ (the probability simplex in $\mathbb{R}^{|\mathcal{S}|}$). The domain of D, denoted $\mathcal{D}_{sv}$, is a Cartesian product of simplices, which is a convex set. The objective

function is $f(D) = s(I_{total}(D))$, where

$$I_{total}(D) = T\sum_{h=1}^{H} I_{approx,h}(D^h) + \lambda I_d,$$

and $D^h = (d_1^h, \ldots, d_K^h)$ are the visitation vectors for timestep h. The term $I_{approx,h}(D^h)$ is given by:

$$I_{approx,h}(D^h) = \Phi^T M_h(D^h)\Phi, \quad \text{with} \quad M_h(D^h) = \frac{1}{K}\sum_{q=1}^{K}\text{diag}(d_q^h) - \left(\frac{1}{K}\sum_{q=1}^{K} d_q^h\right)\left(\frac{1}{K}\sum_{q=1}^{K} d_q^h\right)^T.$$

We will prove the concavity of $f(D)$ by showing that $I_{total}(D)$ is a concave matrix-valued function of D, and then using the properties of $s(\cdot)$.

1. **Concavity of** $M_h(D^h)$: Let $L_h(D^h) = \frac{1}{K}\sum_{q=1}^{K}\text{diag}(d_q^h)$. The function $\text{diag}(v)$ is a linear mapping from a vector v to a diagonal matrix. Thus, $L_h(D^h)$ is a linear function of the collection of vectors $D^h = (d_1^h, \ldots, d_K^h)$. Linear functions are both concave and convex.

Let $\bar{d}^h(D^h) = \frac{1}{K}\sum_{q=1}^{K} d_q^h$. This is also a linear function of D^h. Consider the function $Q(v) = -vv^T$. The function $v \mapsto vv^T$ is convex. To see this, for v_1, v_2 and $\alpha \in [0,1]$:

$$\begin{aligned}
&\alpha v_1 v_1^T + (1-\alpha)v_2 v_2^T - (\alpha v_1 + (1-\alpha)v_2)(\alpha v_1 + (1-\alpha)v_2)^T \\
&= \alpha v_1 v_1^T + (1-\alpha)v_2 v_2^T - (\alpha^2 v_1 v_1^T + \alpha(1-\alpha)(v_1 v_2^T + v_2 v_1^T) + (1-\alpha)^2 v_2 v_2^T) \\
&= (\alpha - \alpha^2)v_1 v_1^T + ((1-\alpha) - (1-\alpha)^2)v_2 v_2^T - \alpha(1-\alpha)(v_1 v_2^T + v_2 v_1^T) \\
&= \alpha(1-\alpha)v_1 v_1^T + \alpha(1-\alpha)v_2 v_2^T - \alpha(1-\alpha)(v_1 v_2^T + v_2 v_1^T) \\
&= \alpha(1-\alpha)(v_1 - v_2)(v_1 - v_2)^T.
\end{aligned}$$

Since $\alpha(1-\alpha) \geq 0$ and $(v_1 - v_2)(v_1 - v_2)^T \succeq 0$ (it's an outer product, hence positive semidefinite), the expression is $\succeq 0$. Thus, $v \mapsto vv^T$ is convex. Therefore, $Q(v) = -vv^T$ is concave. The composition of a concave function with a linear function is concave. Since $Q(v)$ is concave and $\bar{d}^h(D^h)$ is linear, the function $D^h \mapsto Q(\bar{d}^h(D^h)) = -\bar{d}^h(D^h)(\bar{d}^h(D^h))^T$ is concave.

$M_h(D^h) = L_h(D^h) + Q(\bar{d}^h(D^h))$ is the sum of a linear function (which is concave) and a concave function. Thus, $M_h(D^h)$ is a concave matrix-valued function of D^h.

2. **Concavity of** $I_{approx,h}(D^h)$: The function $I_{approx,h}(D^h) = \Phi^T M_h(D^h)\Phi$ is a congruence transformation of $M_h(D^h)$. Since $M_h(D^h)$ is concave in D^h, and congruence transformations preserve concavity (i.e., if $A(x)$ is concave, then $C^T A(x) C$ is concave for any constant matrix C), $I_{approx,h}(D^h)$ is concave in D^h.

3. **Concavity of** $I_{total}(D)$: The total approximate FIM before regularization is $\sum_{h=1}^{H} I_{approx,h}(D^h)$. Since each $I_{approx,h}(D^h)$ is concave with respect to its arguments D^h (and thus with respect to the full D, as it doesn't depend on $D^{h'}$ for $h' \neq h$), their sum is concave with respect to D. Multiplying by a non-negative scalar T preserves concavity. So, $T\sum_{h=1}^{H} I_{approx,h}(D^h)$ is concave in D. Adding a constant matrix λI_d also preserves concavity. Therefore, $I_{total}(D) = T\sum_{h=1}^{H} I_{approx,h}(D^h) + \lambda I_d$ is a concave matrix-valued function of D.

4. **Concavity of** $s(I_{total}(D))$: We are given that the scalar criterion $s : \mathbb{S}_+^d \to \mathbb{R}$ is concave and matrix-monotone non-decreasing. If $g(x)$ is a matrix-valued concave function and $s(A)$ is a scalar-valued concave and non-decreasing function of matrix A (in the Loewner order), then the composition $s(g(x))$ is concave (see Boyd & Vandenberghe, Convex Optimization, Section 3.2.4). In our case, $g(D) = I_{total}(D)$ is concave in D. Thus, $f(D) = s(I_{total}(D))$ is concave with respect to $D = \{d_q^h\}_{h\in[H],q\in[K]}$ over the convex domain $\mathcal{D}_{sv}$. □

B.9.2. Proof of Convergence Guarantee (Theorem B.6)

Theorem B.6. *[Convergence Guarantee of Algorithm 2 (Detailed)] Let $D^{(n)}$ be the sequence of collections of state visitation measures generated by Algorithm 2, where $D^{(1)}$ is the initialization and $D^{(n+1)}$ is the iterate after n Frank-Wolfe*

steps. Let $f(D) = s(I_{total}(D))$ be the objective function defined in Theorem 5.1, and let $D^* \in \mathcal{D}_{sv}$ be an optimal solution, $D^* = \arg\max_{D \in \mathcal{D}_{sv}} f(D)$. The domain $\mathcal{D}_{sv}$ of valid collections of state visitation measures is compact and convex. If Algorithm 2 performs N_{iter} iterations of the Frank-Wolfe update (i.e., the loop from $n = 1$ to N_{iter} in the algorithm's notation, resulting in the final iterate $D^{(N_{iter}+1)}$), using exact line search for α_n at each iteration, then the suboptimality of the final iterate $D^{(N_{iter}+1)}$ is bounded by:

$$f(D^*) - f(D^{(N_{iter}+1)}) \leq \frac{2C_f}{N_{iter} + 2}$$

where C_f is the curvature constant of f over $\mathcal{D}_{sv}$.

Theorem B.6. *[Convergence Guarantee of Algorithm 2 (Detailed)] Let $D^{(n)}$ be the sequence of collections of state visitation measures generated by Algorithm 2, where $D^{(1)}$ is the initialization and $D^{(n+1)}$ is the iterate after n Frank-Wolfe steps. Let $f(D) = s(I_{total}(D))$ be the objective function defined in Theorem 5.1, and let $D^* \in \mathcal{D}_{sv}$ be an optimal solution, $D^* = \arg\max_{D \in \mathcal{D}_{sv}} f(D)$. The domain $\mathcal{D}_{sv}$ of valid collections of state visitation measures is compact and convex. If Algorithm 2 performs N_{iter} iterations of the Frank-Wolfe update (i.e., the loop from $n = 1$ to N_{iter} in the algorithm's notation, resulting in the final iterate $D^{(N_{iter}+1)}$), using exact line search for α_n at each iteration, then the suboptimality of the final iterate $D^{(N_{iter}+1)}$ is bounded by:*

$$f(D^*) - f(D^{(N_{iter}+1)}) \leq \frac{2C_f}{N_{iter} + 2}$$

where C_f is the curvature constant of f over $\mathcal{D}_{sv}$.

Proof. The convergence of Algorithm 2 relies on standard results for the Frank-Wolfe algorithm when maximizing a concave function over a compact convex set. We verify the conditions required for these guarantees.

1. Objective Function and Domain:

- **Concavity:** The objective function $f(D) = s(I_{total}(D))$ is concave with respect to the collection of state visitation vectors $D = \{d_q^h\}_{h,q}$, as proven in Theorem 5.1.
- **Domain $\mathcal{D}_{sv}$:** The domain $\mathcal{D}_{sv}$ is the set of all valid collections of state visitation measures $\{d_q^h\}_{h,q}$. Each d_q^h is a probability distribution over the finite state space $\mathcal{S}$, so it belongs to the probability simplex $\Delta^{|\mathcal{S}|-1}$. The full domain $\mathcal{D}_{sv}$ is a Cartesian product of $K \times H$ such simplices. Each simplex is compact and convex, and thus their Cartesian product $\mathcal{D}_{sv}$ is also compact and convex.

2. Frank-Wolfe Algorithm Steps: Algorithm 2 implements the Frank-Wolfe algorithm:

- **Initialization (Line 13):** ${}^{(1)}d_{\text{mix},q} \leftarrow \mathbf{0}$. This initializes the iterate $D^{(1)}$ within $\mathcal{D}_{sv}$ (the zero vector is on the boundary of the simplex if non-negativity is the only constraint, or can be seen as a valid (degenerate) visitation measure).
- **Gradient Computation (Line 16):** The algorithm computes the gradient $\nabla f(D^{(n)})$ (implicitly, by computing r_{grad_q} which is derived from this gradient).
- **Linear Maximization Oracle (LMO) (Lines 17-18):** For each q, the step $\pi^{(n)}_{\text{grad}_q} \leftarrow \texttt{value\_iteration}(M, r_{\text{grad}_q})$ finds a policy that maximizes the linear objective $\sum_{s,a} d^h_\pi(s,a) r_{\text{grad}_q}(h,s,a)$ over all policies π. This is equivalent to finding a vertex $S_q^{(n)}$ of the polytope of visitation measures for policy q that maximizes $\langle \nabla_{d_q^h} f(D^{(n)}), S_q^{(n)} \rangle$. The collection of these $S_q^{(n)}$ for all q forms the $S^{(n)}$ in the standard Frank-Wolfe update $S^{(n)} = \arg\max_{S \in \mathcal{D}_{sv}} \langle \nabla f(D^{(n)}), S \rangle$. The computation of $d^{(n)}_{\text{grad}_q}$ from $\pi^{(n)}_{\text{grad}_q}$ yields this $S^{(n)}$.
- **Step Size (Line 20):** α_n is determined by exact line search: $\alpha_n \leftarrow \arg\max_{\alpha' \in [0,1]} f((1-\alpha')D^{(n)} + \alpha' S^{(n)})$.
- **Update (Line 22):** $D^{(n+1)} \leftarrow (1-\alpha_n)D^{(n)} + \alpha_n S^{(n)}$.

The algorithm performs $N_{iter} = N - 1$ such iterations, producing iterates $D^{(2)}, \ldots, D^{(N)}$. (Note: $D^{(1)}$ is the initialization).

3. Convergence Rate: For a concave function f maximized over a compact convex set $\mathcal{D}$ using the Frank-Wolfe algorithm with exact line search for the step size, the suboptimality gap $h_k = f(D^*) - f(D^{(k+1)})$ after k iterations (where $D^{(1)}$ is the initial point and $D^{(k+1)}$ is the iterate after k Frank-Wolfe steps) is bounded by (Jaggi, 2013, Theorem 1 and discussion for line search):

$$f(D^*) - f(D^{(k+1)}) \leq \frac{2C_f}{k+2}$$

where C_f is the curvature constant of f over $\mathcal{D}$, defined as

$$C_f = \sup_{\substack{X,S \in \mathcal{D}, \gamma \in (0,1] \\ Y = (1-\gamma)X + \gamma S}} \frac{2}{\gamma^2}(f(X) + \gamma \langle \nabla f(X), S - X \rangle - f(Y)).$$

In our case, Algorithm 2 initializes with $D^{(1)}$ and performs N_{iter} iterations of the Frank-Wolfe update (corresponding to the loop variable n from 1 to N_{iter} in the algorithm's notation as per Algorithm 2 where the loop runs $N - 1$ times; here we use N_{iter} to denote this count of iterations). The final iterate is $D^{(N_{iter}+1)}$. The standard bound $2C_f/(k+2)$ applies after k iterations. Here, $k = N_{iter}$. So, the suboptimality of the final iterate $D^{(N_{iter}+1)}$ is bounded by:

$$f(D^*) - f(D^{(N_{iter}+1)}) \leq \frac{2C_f}{N_{iter}+2}$$

This holds for $N_{iter} \geq 1$. The constant C_f depends on the objective function f and the domain $\mathcal{D}_{sv}$. Since $\mathcal{D}_{sv}$ is compact, C_f is well-defined and finite, provided f is continuously differentiable (which it is, assuming $s(\cdot)$ is, and $I_{total}(D)$ is differentiable).

Thus, Algorithm 2 converges to the global optimum with a rate of $O(1/N_{iter})$. □