Inference-Time Personalized Alignment with a Few User Preference Queries

Victor-Alexandru Pădurean
MPI-SWS
vpadurea@mpi-sws.org

Parameswaran Kamalaruban
Featurespace Innovation Lab, Visa
kaparame@visa.com

Nachiket Kotalwar
Carnegie Mellon University
nkotalwa@cs.cmu.edu

Alkis Gotovos
MPI-SWS
agkotovo@mpi-sws.org

Adish Singla
MPI-SWS
adishs@mpi-sws.org

Abstract

We study the problem of aligning a generative model's response with a user's preferences. Recent works have proposed several different formulations for personalized alignment; however, they either require a large amount of user preference queries or require that the preference be explicitly specified as a text input. In this paper, we propose a novel inference-time personalized alignment method, USERALIGN, that elicits the user's preferences with a few queries as pairwise response comparisons. In particular, USERALIGN builds on the theoretical framework of best-arm identification in logistic bandits and selects a personalized response from a fixed pool of the model's generated responses. The key idea is to consider the user's feedback consistent and noise-free, and incorporate it into the theoretical framework to identify the best response quickly. Experimental results across several tasks, involving personalized text and image generation, showcase the effectiveness of USERALIGN in achieving personalized alignment.

1 Introduction

Generative models have demonstrated remarkable capabilities across language and vision tasks, yet aligning their outputs with human preferences remains a central challenge [1, 2]. While population-level alignment methods such as Reinforcement Learning from Human Feedback (RLHF) [3, 4, 5] and Direct Preference Optimization (DPO) [6, 7, 8] have made significant strides, practical applications often demand personalization. Users exhibit highly individual tastes and requirements, from stylistic writing preferences to visual aesthetics to lifestyle preferences, which generic alignment cannot fully capture. Consequently, our key research question is: *How can we align a generative model's response to a specific user on the fly, where the user's preferences need to be elicited with limited interaction?*

Recent efforts toward personalized alignment have explored both training-time and inference-time strategies, each with drawbacks when query budgets are small. Training-time personalization approaches fine-tune models on user-specific data but typically rely on extensive preference annotations [9, 10]. Inference-time methods offer more flexibility by adapting model outputs at deployment; however, many require users to articulate their preferences as explicit text prompts [11, 12], which can be cognitively demanding and imprecise for complex tastes. Even theoretically grounded active learning and bandit-based methods for modeling latent reward functions demand a lot of pairwise comparisons to converge to a reliable preference estimate [13, 14], making them impractical for real-world, low-interaction settings.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Figure 1: An illustrative example showcasing inference-time personalized alignment methodology. Starting with a user question (Stage 1), the system generates a pool of responses (Stage 2). Then, USERALIGN iteratively collects user preferences (Stage 3) via pairwise comparisons to determine the most suitable response—user's preferred responses are highlighted. At the end, the final response is selected (Stage 4). The user in this example is simulated by the GPT-4o-mini model conditioned on the persona description "*NostalgicExplorer: A 36-year-old who grew up with classic platformers and adventure games. Loves timeless heroes with a sense of wonder and a hint of retro charm*".

In this paper, we introduce USERALIGN, a novel inference-time personalized alignment method that efficiently elicits user preferences through only a few pairwise comparisons among a fixed pool of candidate responses. Building on the best-arm identification framework in logistic bandits [15, 16, 17, 18, 19], USERALIGN maintains a loss-based confidence region over the user's latent preference model and aggressively shrinks this region by treating each comparison as consistent and noise-free. By leveraging version-space elimination via intersecting halfspaces defined by the observed duels [20, 21], our method rapidly isolates the best response without extensive querying. Figure 1 provides an illustrative example showcasing USERALIGN's interaction with the user on an image generation task. Our main results and contributions are summarized below:

- We formulate the problem of inference-time personalized alignment with a particular focus on practical settings where the user's preferences need to be elicited with limited interaction. (Section 3)
- We develop a novel method, USERALIGN, theoretically grounded in the logistic bandits framework, and achieve fast alignment by modeling the user's feedback as consistent and noise-free. (Section 4)
- We demonstrate that USERALIGN can achieve fast personalized alignment with a few preference queries in personalized text and image generation, evaluated on both simulated and real users. We release our implementation and datasets to support further research.[1] (Sections 5 and 6)

[1] Github repo: https://github.com/machine-teaching-group/neurips2025-useralign.

Table 1: Related work on preference alignment of generative models; see Section 2 for details.

(a) **Preference alignment methods.** General preference alignment methods use aggregated population-level preference data, while personalized methods rely on user-level data (with user identity).

		Training-time	Inference-time
General Preferences	**Offline**	RLHF [3, 4, 5] DPO [6, 7, 8]	BoN [22, 23] Reward guided decoding [24, 25, 26]
	Online	Active RLHF [27, 28] Active DPO [28, 30, 31]	Active BoN [29, 30] Active Bayesian PM [14, 32, 33]
Personal Preferences	**Offline**	Personalized RLHF [9] Personalized DPO [10] VPL [37]	URIAL [11], DeAL [12] PAD [34], OPAD [35], Amulet [36] Personalized Soups [38], MOD [39] LoRE [40], PAL [41]
	Online		PBO [42], APL [43] Active BoN [29, 30] Active Bayesian pref. model [14, 32, 33] USERALIGN

(b) **Inference-time personalized preference alignment methods.** User preference input can take the form of explicit text (e.g., preference specifications or prompt-response examples), pairwise comparisons of base model outputs, or weights over predefined objectives. Offline methods impose less user load than online methods, as they do not require active user interaction. Warmup options include training a personalized preference model with multi-user pairwise data, or training an ensemble of generative models for different objectives. Inference-time operations include in-context learning, active preference learning from user comparisons, logit adjustment using on-the-fly reward functions with text-based preference input, logit adjustment via learned user preference weights with a pre-trained preference model, or logit adjustment using an ensemble of pre-trained generative models and user-defined weights. Online methods typically account for uncertainty in user preference modeling. Aligned responses can be generated via guided decoding or selection from a pool of pre-generated outputs.

	Preference Input	**User Load**	**Warmup Operation**	**Test-Time Operation**	**Uncertainty Quantification**	**Response Generation**
URIAL [11]	Text	Low	None	In-context learn	No	Guiding
DeAL [12]	Text	Low	None	In-context learn	No	Guiding
PAD [34]	Text	Low	Train pref. model	On-the-fly pref.	No	Guiding
OPAD [35]	Text	Low	None	On-the-fly pref.	No	Guiding
Amulet [36]	Text	Low	None	On-the-fly pref.	No	Guiding
Personalized Soups [38]	Weight	Low	Train gen. models	Ensemble	No	Guiding
MOD [39]	Weight	Low	Train gen. models	Ensemble	No	Guiding
LoRE [40]	Comparisons	Low	Train pref. model	Learn pref. weight	No	Guiding
PAL [41]	Comparisons	Low	Train pref. model	Learn pref. weight	No	Guiding
PBO [42]	Comparisons	High	None	Active pref. model	Yes	Selection
APL [43]	Comparisons	High	None	Active pref. model	Yes	Selection
Active BoN [29, 30]	Comparisons	High	None	Active pref. model	Yes	Selection
Active Bayesian pref. model [14]	Comparisons	High	None	Active pref. model	Yes	Selection
USERALIGN	Comparisons	Low	None	Active pref. model	Yes	Selection

2 Related Work

A broad range of general preference alignment methods has emerged (see Table 1a). Offline approaches like Reinforcement Learning from Human Feedback (RLHF) [3, 4, 5] and Direct Preference Optimization (DPO) [6, 7, 8] fine-tune models using aggregated preference data. At inference, methods like Best-of-N sampling (BoN) [22, 23] and reward-guided decoding [24, 25, 26] steer outputs without retraining. To lower annotation cost, online variants such as Active RLHF [27, 28] and Active DPO [28, 30, 31] adapt losses on-the-fly, while methods like Active BoN [29, 30] and Active Bayesian preference modeling [14, 32, 33] refine decoding via sequential pairwise feedback. These methods capture broad community norms but struggle to personalize outputs for individual users.

In contrast, personalized preference alignment methods leverage user-specific data to generate individualized outputs (see Table 1b). Offline methods [9, 10, 37] fine-tune models on per-user preference data. At inference, text-based logit adjustment methods like URIAL [11] and DeAL [12] embed few-shot examples, while PAD [34], OPAD [35], and Amulet [36] learn lightweight reward functions on-the-fly. Personalization is also supported by weight-based ensembling (Personalized Soups [38], MOD [39]) and comparison-driven logit adjustment (LoRE [40], PAL [41]). Online personalization algorithms, such as PBO [42] and APL [43], actively query users for pairwise comparisons to refine

Algorithm 1 System-User Interaction

1: User u provides an input prompt $x \in \mathcal{X}$ to the system.
2: System uses the generative model π to generate a pool of responses $\mathcal{Y}_{\text{cand}}$ for x.
3: **while** *stopping criteria not met* **do**
    System selects a pair of responses $(y, y') \in \mathcal{Y}_{\text{cand}}^2$.
    System asks the user to provide a preference over the pair (y, y').
    System sets $r = 1$ if the user prefers y over y', and $r = 0$ otherwise.
    System updates the preference dataset $\mathcal{D}$ with the preference tuple (x, y, y', r).
4: System selects a final response $\widehat{y} \in \mathcal{Y}_{\text{cand}}$ for the user.

decoders in real time; Active BoN and Active Bayesian preference modeling naturally extend to this user-specific setting. Recent surveys provide broader overviews of personalized alignment [44, 45].

Another line of related work is Bayesian active learning that treats a user's latent reward as a random variable and selects queries that most reduce posterior uncertainty [14, 46], enabling sample-efficient recovery of preference weights under mild assumptions. Similarly, the dueling bandit framework models preference learning as an online decision problem with pairwise feedback [13, 47], where algorithms using upper-confidence bounds or Thompson sampling achieve sublinear regret and convergence guarantees [48, 49]. However, both approaches often require many user queries in practice.

3 Problem Formulation

System-user interaction. The envisioned system is powered by a generative model and interacts with a user u. The generative model is a stochastic mapping $\pi : \mathcal{X} \to \Delta(\mathcal{Y})$, where $\mathcal{X}$ is the input space and $\mathcal{Y}$ is the output space ($\Delta(\mathcal{Y})$ denotes the probability simplex over $\mathcal{Y}$). The user's preferences are captured by an unknown latent preference model: for any input $x \in \mathcal{X}$ and a pair of responses $(y, y') \in \mathcal{Y}^2$, the probability that the user prefers y over y' (denoted $y \succ y'$) is given by the preference model $p_u[y \succ y' \mid x]$. Given an input $x \in \mathcal{X}$ from the user, the system first generates a candidate pool of responses $\mathcal{Y}_{\text{cand}}$ using the generative model π, and then seeks to output a preference-aligned response $\widehat{y}$ from this pool. To this end, the system must learn or infer the user's preference model for x through interaction. This interaction consists of querying the user to express a preference over a pair $(y, y') \in \mathcal{Y}_{\text{cand}}^2$ from the pool. Throughout, the system maintains and updates a dataset of preferences $\mathcal{D} = \{(x, y, y', r)\}$, where $r = 1$ if the user prefers y over y' for x, and $r = 0$ otherwise. The complete interaction process is described in Algorithm 1.

Objective. Let $\widetilde{y} \in \mathcal{Y}$ be a baseline response for x generated by the model π with zero sampling temperature. This response is produced prior to any user interaction. For input x, we define the *win-rate* of any response $y \in \mathcal{Y}$ against the baseline $\widetilde{y}$ as $p_u[y \succ \widetilde{y} \mid x]$. The system aims to output a response that approximately maximizes this win-rate while using as few preference queries as possible. Specifically, given the candidate pool $\mathcal{Y}_{\text{cand}}$ for x, the goal is to find the response $y^\star = \arg\max_{y \in \mathcal{Y}_{\text{cand}}} p_u[y \succ \widetilde{y} \mid x]$ with minimal interaction.

4 Methodology

In this section, we present our algorithm, UserAlign, for generating user preference-aligned responses using best-arm identification methods from logistic bandits literature. The full procedure is outlined in Algorithm 2, with its subroutine Solve detailed in Algorithm 3.

Preliminaries. For any input $x \in \mathcal{X}$ and a pair of responses $(y, y') \in \mathcal{Y}^2$, the Bradley-Terry-Luce (BTL) preference model is defined as $p_{\text{BTL}}\left[y \succ y' \,\middle|\, x, \theta\right] := \mu(\langle \theta, \phi(x, y) - \phi(x, y') \rangle)$, where $\theta \in \Theta \subset \mathbb{R}^d$ is a weight vector, $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$ is a feature mapping, and $\mu(z) = 1/(1 + e^{-z})$ denotes the logistic function. To develop our algorithm, we consider a user preference model p_u following the BTL model with some unknown $\theta^\star \in \Theta$. We also adopt the following standard assumption [50]: $\|\phi(x, y) - \phi(x, y')\|_2 \leq 1, \forall x \in \mathcal{X}, y, y' \in \mathcal{Y}$ and $\|\theta^\star\|_2 \leq S$ with known $S > 0$. We define $\kappa^\star_{\mathcal{X},\mathcal{Y}} := \max_{x \in \mathcal{X}} \max_{y,y' \in \mathcal{Y}} \frac{1}{\dot{\mu}(\langle \theta^\star, \phi(x,y) - \phi(x,y') \rangle)}$ and $\Delta_{\mathcal{Y}_{\text{cand}}} := \min_{y,y' \in \mathcal{Y}_{\text{cand}}; y \neq y'} \langle \theta^\star, \phi(x, y) - \phi(x, y') \rangle$. Under this setup, the response maximizing the win-rate can be equivalently written as:

$$y^\star \;=\; \arg\max_{y \in \mathcal{Y}_{\text{cand}}} \langle \theta^\star, \phi(x, y) \rangle. \tag{1}$$

Algorithm 2 USERALIGN: Eliciting User Preferences

Input: generative model $\pi : \mathcal{X} \to \mathcal{Y}$, feature mapping $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$, stopping threshold ϵ, confidence level δ, problem dimension d, and norm bound S
User u provides an input prompt $x \in \mathcal{X}$ to the system.
Generate a diverse set of responses $\mathcal{Y}_{\text{cand}}$ for x by sampling from π.
Initialize the preference dataset $\mathcal{D}_0 \leftarrow \{\}$.
for $t = 0, 1, 2, \ldots$ **do**
▷ Obtain a representative preference model parameter and confidence set.
Obtain $(\widehat{\theta}_t, \Theta_t) \leftarrow \textsc{Solve}(\mathcal{D}_t, d, S, t, \delta)$
▷ Obtain responses for comparison.
Select the first response $y_t^{(1)} \leftarrow \arg\max_{y \in \mathcal{Y}_{\text{cand}}} \langle \widehat{\theta}_t, \phi(x, y) \rangle$.
Select the second response $(y_t^{(2)}, \widetilde{\theta}_t) \leftarrow \arg\max_{(y', \theta) \in \mathcal{Y}_{\text{cand}} \times \Theta_t} \langle \theta, \phi(x, y') - \phi(x, y_t^{(1)}) \rangle$.
▷ Check the stopping condition.
Compute stopping criteria $B(t) = \langle \widetilde{\theta}_t, \phi(x, y_t^{(2)}) - \phi(x, y_t^{(1)}) \rangle$.
if $B(t) \leq \epsilon$ **then**
Output: Response $y_t^{(1)}$ to the user u.
▷ Obtain user feedback and update the preference dataset.
Ask the user u to provide preference over responses $y_t^{(1)}$ and $y_t^{(2)}$.
Observe $r_t \sim p_u[y_t^{(1)} \succ y_t^{(2)} \mid x]$ and update $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x, y_t^{(1)}, y_t^{(2)}, r_t)\}$.

Algorithm 3 USERALIGN: SOLVE Subroutine

Input: preference dataset $\mathcal{D}_t$, dimension d, norm bound S, step t, and confidence level δ
Compute the MLE $\widehat{\theta}_t$ by solving the optimization problem in Eq. (2).
Construct the loss-based confidence set Θ_t as in Eq. (3).
▷ Obtain the practical confidence set (see Section 4.2).
Update $\Theta_t \leftarrow \Theta_t \cap \mathcal{H}_t$ w.r.t. the consistency set in Eq. (4);
when updated Θ_t is empty, set $\Theta_t = \{\widehat{\theta}_t\}$.
Output: preference parameter $\widehat{\theta}_t$ and confidence set Θ_t

4.1 Theoretical Framework for USERALIGN

Here, we describe USERALIGN without line 4 in the SOLVE subroutine, referring to this variant as $\text{USERALIGN}_{\text{LOSS}}$. Starting with an empty preference dataset $\mathcal{D}_0 = \{\}$, we actively populate it with preference tuples $(x, y^{(1)}, y^{(2)}, r)$, where $r = 1$ if user u prefers $y^{(1)}$ over $y^{(2)}$, and $r = 0$ otherwise. Let $\mathcal{D}_t = \{(x_\tau, y_\tau^{(1)}, y_\tau^{(2)}, r_\tau)\}_{\tau=0}^{t-1}$ be the dataset at step t. We select a pair of responses $(y_t^{(1)}, y_t^{(2)})$ for which we request user feedback and then add the resulting tuple $(x, y_t^{(1)}, y_t^{(2)}, r_t)$ to $\mathcal{D}_t$.

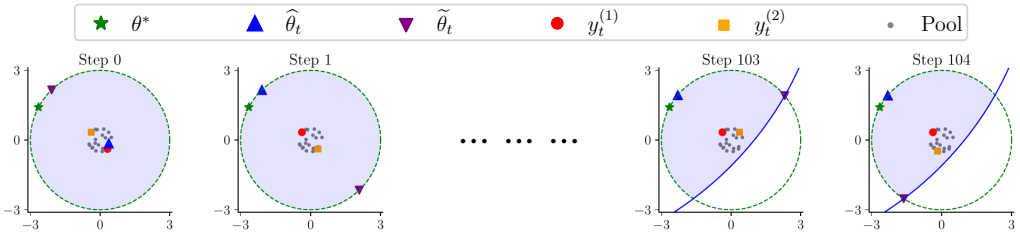
SOLVE. For the preference dataset $\mathcal{D}_t$, we define the negative log-likelihood function as follows: $\mathcal{L}_t(\theta) := \sum_{\tau=0}^{t-1} \ell(\theta; (x_\tau, y_\tau^{(1)}, y_\tau^{(2)}, r_\tau))$, where $\ell(\theta; (x, y^{(1)}, y^{(2)}, r)) := -r \cdot \log \mu(\langle \theta, z \rangle) - (1 - r) \cdot \log(1 - \mu(\langle \theta, z \rangle))$ is the logistic loss function, with $z = \phi(x, y^{(1)}) - \phi(x, y^{(2)})$. Finally, we obtain the norm-constrained, unregularized maximum likelihood estimator (MLE) of the unknown parameter $\theta^\star$ via solving the following optimization problem:

$$\widehat{\theta}_t \;:=\; \underset{\|\theta\|_2 \leq S}{\arg\min}\, \mathcal{L}_t(\theta). \tag{2}$$

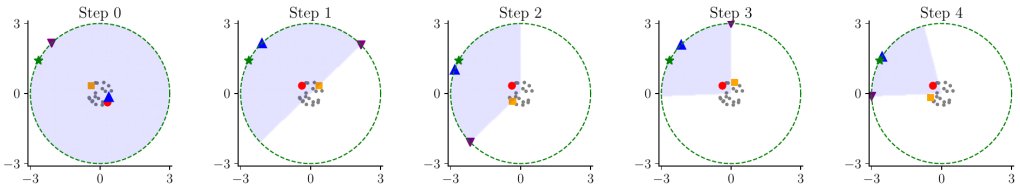
We construct the loss-based confidence set as follows:

$$\Theta_t \;:=\; \left\{ \theta \in \mathbb{R}^d : \|\theta\|_2 \leq S \text{ and } \mathcal{L}_t(\theta) \leq \mathcal{L}_t(\widehat{\theta}_t) + \beta_t \right\}, \tag{3}$$

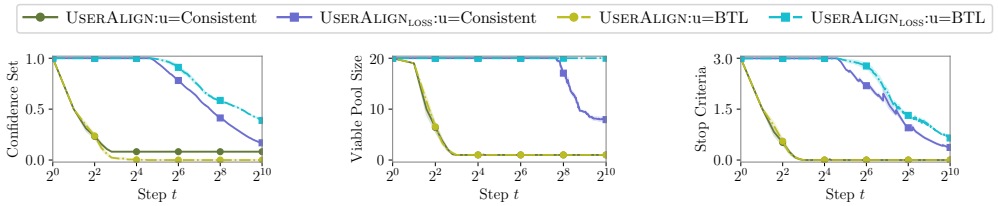
where $\beta_t = 10d \log\left(\frac{St}{4d} + e\right) + 2((e-2) + S) \log \frac{1}{\delta}$ [19]. Since $\mathcal{L}_t$ is convex, the confidence set Θ_t is also convex. The SOLVE procedure is presented in Algorithm 3. After obtaining $\widehat{\theta}_t$ and Θ_t, we select the first response $y_t^{(1)}$ as the one that maximizes the win-rate objective in Eq. (1) under $\widehat{\theta}_t$. Next, we choose the second response $y_t^{(2)}$ to maximize the regret of $y_t^{(1)}$ within Θ_t. If this maximum regret (i.e., $B(t)$ in line 9 of Algorithm 2) is below the given ϵ threshold, we output $y_t^{(1)}$.



(a) Loss-based confidence set Θ_t in Eq. (3) shrinks slowly due to incremental log-loss updates.



(b) Practical confidence set $\Theta_t \cap \mathcal{H}_t$ shrinks aggressively due to version-space elimination.



(c) Normalized area of confidence set, viable response pool size, and stopping criteria over interaction steps. The practical confidence set shrinks significantly faster.

Figure 2: Geometric convergence in a two-dimensional synthetic domain. We consider a 2D preference space where each candidate response is represented by a point randomly sampled from the ball of radius 0.5. For each run, the ground-truth user preference $\theta^\star$ is sampled uniformly from the circle with radius 3. The plots compare the rate at which the loss-based and practical confidence sets shrink over successive pairwise comparisons.

Theoretical Analysis. The following proposition shows that the loss-based confidence set Θ_t contains the true parameter $\theta^\star$ with high probability [19]. Proofs are provided in the supplementary material.

Proposition 1. *For the confidence set Θ_t defined in Eq. (3), we have: $\mathbb{P}\left[\forall t \geq 0, \theta^\star \in \Theta_t\right] \geq 1-\delta$.*

Note that for any θ', $\mathcal{L}_t(\theta) - \mathcal{L}_t(\theta') \leq \mathcal{L}_t(\theta) - \mathcal{L}_t(\widehat{\theta}_t) \leq \beta_t$. Therefore, even if only an approximate estimate of $\widehat{\theta}_t$ is obtained, the high-probability guarantee that $\theta^\star \in \Theta_t$ still holds. The following theorem shows that UserAlign$_{\text{Loss}}$ identifies an ϵ-near-optimal response with high probability.

Theorem 1. *Let τ be the stopping round of* UserAlign$_{\text{Loss}}$*, and $y^\star$ is defined in Eq. (1). Then, the response $y_\tau^{(1)}$ returned by* UserAlign$_{\text{Loss}}$ *satisfies $\mathbb{P}[\langle \theta^\star, \phi(x, y^\star) - \phi(x, y_\tau^{(1)})\rangle \leq \epsilon] \geq 1-\delta$.*

Below, we show that the stopping time of UserAlign$_{\text{Loss}}$ is bounded with high probability.

Theorem 2. *Let τ be the stopping round of* UserAlign$_{\text{Loss}}$*. Define $K = |\mathcal{Y}_{cand}|$, and $\Omega := \frac{S^2 \kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{cand}}\}^2}\left(d + \log\frac{1}{\delta}\right)$. Then, with probability at least $1-\delta$, we have: $\tau \leq \mathcal{O}(\Omega K^2 \cdot \log(\Omega K^2))$.*

4.2 Practical Confidence Set in UserAlign

Despite strong convergence guarantees, UserAlign$_{\text{Loss}}$ requires a larger number of user preference queries to identify the best (win-rate maximizing) response in practice. This is because the loss-based confidence set defined in Eq. (3) shrinks slowly due to incremental log-loss updates (see Figure 2). This behavior is expected under the stochastic BTL model of user preferences. To overcome this challenge, the key idea is to treat the user's preference feedback as consistent and

Table 2: Domain specifications for our experiments involving personalized text and image generation.

Domain	Preference Space	# Questions	# Users	Response Pool	Example Question
`food2d`	2d (domain-specific)	10	3	20	What should I cook for dinner tonight?
`food64d`	64d (Potion [52])	10	3	20	What should I cook for dinner tonight?
`travel64d`	64d (Potion [52])	10	3	20	Which travelling destination would surprise me in the best possible way?
`visual512d`	512d (OpenCLIP [53, 54])	10	3	40	Generate concept art for a videogame hero that players would instantly connect with.
`dsp64d`	64d (Potion [52])	100	3	20	Describe the main character of Shakespeare's play Hamlet.

noise-free[2], and incorporate it into the theoretical framework to identify the best response more efficiently. Specifically, by leveraging version-space elimination via intersecting halfspaces consistent with observed preference tuples [20, 21], we can aggressively shrink the confidence set. Given the preference dataset $\mathcal{D}_t = \{(x_\tau, y_\tau^{(1)}, y_\tau^{(2)}, r_\tau)\}_{\tau=0}^{t-1}$, we define the consistent half-spaces set as:

$$\mathcal{H}_t := \left\{\theta \in \mathbb{R}^d : r_\tau \cdot \langle \theta, z_\tau \rangle - (1 - r_\tau) \cdot \langle \theta, z_\tau \rangle \geq 0 \text{ for all } \tau \in [t-1]\right\}, \tag{4}$$

where $z_\tau = \phi(x_\tau, y_\tau^{(1)}) - \phi(x_\tau, y_\tau^{(2)})$. Using this consistent set of halfspaces, we refine the loss-based confidence set as $\Theta_t \leftarrow \Theta_t \cap \mathcal{H}_t$, enabling rapid identification of a near-optimal response with fewer queries; in case Θ_t becomes empty, we set it as $\Theta_t = \{\widehat{\theta}_t\}$. We refer to the resulting algorithm with the updated confidence set as our main method, UserAlign. The computational efficiency of UserAlign is discussed in Appendix F.1. In the following section, we empirically demonstrate the effectiveness of UserAlign in quickly selecting personalized responses.

5 Experimental Evaluation

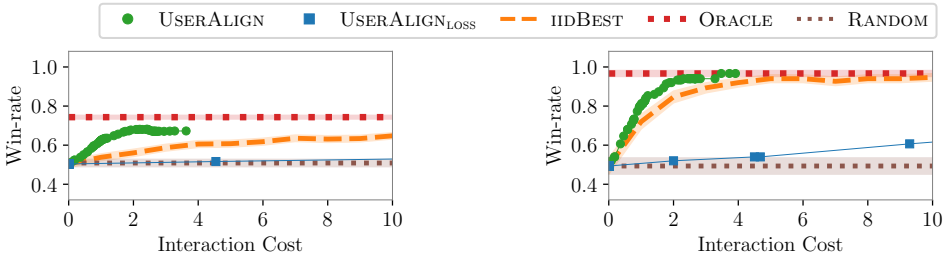
To thoroughly evaluate our method, we consider a diverse set of domains (see Table 2). We begin with `food2d`, enabling controlled experiments with users modeled through BTL in an interpretable 2D space. We then assess real-world domains (`food64d`, `travel64d`, `visual512d`) to test performance with complex GPT-simulated personas. Finally, we include `dsp64d` based on an existing large-scale benchmark to rigorously examine scalability and robustness.

Domain `food2d`. This domain defines Θ as a 2D space with dimensions 'spiciness' and 'veginess', each feature ranging from -1 (no spice/animal protein) to 1 (high spice/plant-based protein). Building on this space, we define the set of questions, construct user models, generate response pools, and obtain feature representations. First, we construct $\mathcal{X}$ as 10 food recommendation questions. Second, for each question, we sample 3 user preferences $\theta^\star$ uniformly on the circle of radius $S = 3$ in $\mathbb{R}^2$ (i.e., $\|\theta^\star\|_2 = 3$). We conduct experiments with two types of user behavior: p_u modeled through BTL, and p_u considered consistent and noise-free. Third, we generate a candidate pool $\mathcal{Y}_{\text{cand}}$ of 20 responses per question $x \in \mathcal{X}$ with GPT-4o [51]. Finally, we obtain $\phi(x, y)$ by asking GPT-4o to map a question-response pair to the 2D 'spiciness'/'veginess' space, given the domain description as context.

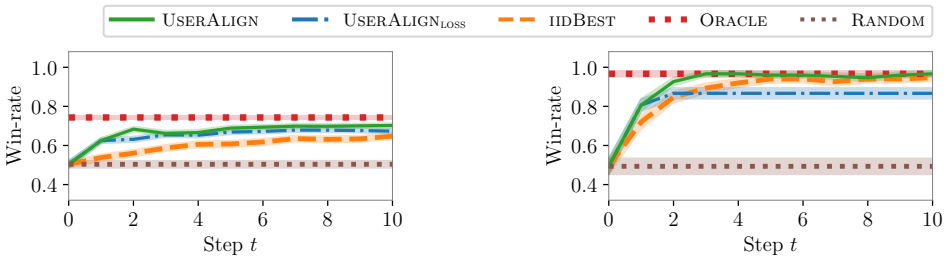
Domains `food64d`**,** `travel64d`**, and** `visual512d`. We define Θ as $\{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq S\}$, with dimensionality d based on pretrained embeddings and $S = 3$. First, we define $\mathcal{X}$ as 10 questions per domain (see Table 2). Second, for each question, we construct 3 brief persona descriptions representing user preferences, and simulate user behavior p_u by prompting GPT-4o-mini [55] (temperature $= 0$) to select preferred responses conditioned on these descriptions (see Figure 1). Third, we construct $\mathcal{Y}_{\text{cand}}$ by generating 20 responses per question with GPT-4o for text domains (`food64d`, `travel64d`) and 40 images per question with GPT-Image-1 [56] for `visual512d`. Finally, we obtain $\phi(x, y)$ using pretrained models, specifically a sentence transformer for the 64-dimensional text domains [52] and an OpenCLIP variant for the 512-dimensional vision domain [53, 57, 54]. The framework is modular, so specialized (handcrafted or learned) embeddings can replace the pretrained ones to capture finer nuances when needed.

Domain `dsp64d`. For this domain, we similarly define Θ with dimensionality based on pretrained embeddings and $S = 3$. First, we define $\mathcal{X}$ as 100 questions sampled from the Domain Specific

[2]Our work focuses on short-term, task-specific interaction sessions, where users typically have clear and stable preferences. In such cases, assuming consistent and noise-free feedback is both practical and realistic.



(a) u = BTL: Win-rate vs. interaction cost (b) u = Consistent: Win-rate vs. interaction cost



(c) u = BTL: Win-rate w.r.t. increasing steps (d) u = Consistent: Win-rate w.r.t. increasing steps

Figure 3: Results on `food2d` across two types of user behavior. Top row shows win-rate vs. interaction cost trade-off; here UserAlign and UserAlign$_{\text{Loss}}$ results show scatter plot corresponding to varying values of ϵ. Bottom row shows win-rate per increasing interaction steps; here UserAlign and UserAlign$_{\text{Loss}}$ are run for a given number of steps as mentioned in Footnote 3. In these plots, Oracle and Random are flat lines, and iidBest results are reported at different number of given steps.

Preference (DSP) dataset [58, 35]. Second, for each question, we sample 3 user descriptions from the dataset to simulate user behavior p_u, prompting GPT-4o-mini to select preferred responses given each description. Third, we construct the candidate pool $\mathcal{Y}_{\text{cand}}$ for each question using GPT-4o, following the same approach as above. Finally, we obtain $\phi(x, y)$ using the same sentence transformer encoder as for the other text domains, yielding 64-dimensional representations.

5.1 Evaluation Setup

Candidate pool generation. For each question, we generate a diverse pool of candidate responses by following two sampling approaches. We obtain half of the candidate responses by sampling from GPT-4o or GPT-Image-1 at high temperature, conditioned only on the question, resulting in unbiased responses. We obtain the remaining half by prompting the generative model to first reason about possible diverse interests relevant to the question, then generate one response per interest, thereby obtaining greater diversity in the pool. We provide full details about pool generation in the supplementary material, and also report additional experiments using different candidate pools.

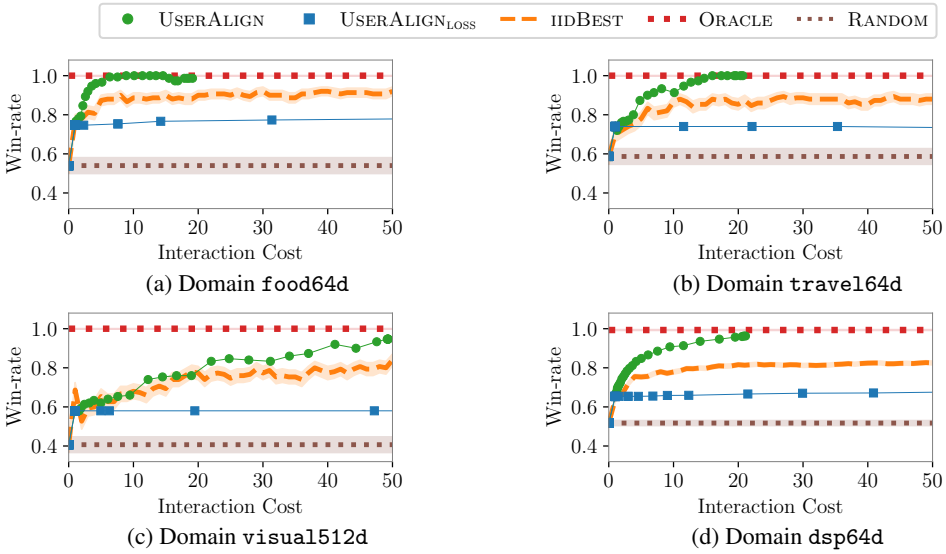
Metrics. We consider two metrics: (i) *interaction cost* as the number of interaction steps taken by a given method before outputting a final response; (ii) *win-rate* $p_u[y \succ \widetilde{y} \mid x]$ of the output response y against baseline $\widetilde{y}$, where $\widetilde{y}$ is generated by the original model (GPT-4o or GPT-Image-1) with zero sampling temperature for the same question x (see Section 3). The win-rate is computed using different types of simulated user preference models p_u, with u = BTL, u = Consistent, or u = GPT.

5.2 Evaluated Methods

Oracle and Random. These two baseline methods provide upper/lower performance bounds. Oracle selects the response with highest utility using explicit knowledge of the user. Random selects a response uniformly at random from the candidate pool. These methods don't have any interaction cost.

iidBest. This method collects user preferences over a fixed number of steps, each time randomly sampling a pair of responses. At the end, it computes the MLE $\widehat{\theta}$ of the user's preference as in Eq. (2), and selects the response maximizing the utility, i.e., $\widehat{y} = \arg\max_{y \in \mathcal{Y}_{\text{cand}}} \langle \widehat{\theta}, \phi(x, y) \rangle$.

UserAlign and UserAlign$_{\text{Loss}}$. These two methods select pairs using Algorithm 2. Our main method, UserAlign, is based on the practical confidence set introduced in Section 4.2.



(a) Domain `food64d` (b) Domain `travel64d`

(c) Domain `visual512d` (d) Domain `dsp64d`

Figure 4: Win-rate vs. interaction cost trade-off on domains with arbitrary preference spaces. Here UserAlign and UserAlign$_{\text{Loss}}$ results show scatter plot corresponding to varying values of ϵ. Oracle and Random are flat lines, and iidBest results are reported at different number of given steps.

UserAlign$_{\text{Loss}}$ is based on the theoretical confidence set, introduced in Section 4.1. Both methods are parameterized by (ϵ, δ) which determines their stopping condition (see lines 1 and 10 in Algorithm 2). We set $\delta = 0.05$ for all the experiments. When reporting results, we will vary the value of ϵ in $[0, S]$ to get a trade-off between interaction cost and win-rate.[3]

5.3 Evaluation Results

All results are averaged over five random seeds, questions, and user personas (see Table 2).

Results w.r.t. different types of users in 2D. Figure 3 shows the results on `food2d` across two types of user behaviors (u = BTL and u = Consistent). UserAlign achieves high win-rate at a low interaction cost, outperforming UserAlign$_{\text{Loss}}$, iidBest, and Random. Moreover, Figures 3a and 3b showcase that UserAlign is effective in automatically deciding how many interaction steps are needed to achieve competitive win-rates based on its stopping criterion.

Results in an arbitrary preference space. Figure 4 shows the results on four domains with arbitrary preference spaces, where the user preferences are given by GPT-based simulated personas. UserAlign achieves competitive win-rates at lower interaction costs, outperforming UserAlign$_{\text{Loss}}$, iidBest, and Random across all domains. As an illustrative example, Figure 1 highlights that UserAlign picks informative comparison queries, and can identify a high-quality response quickly even in high-dimensional settings. Overall, these findings highlight the robustness and adaptability of our method across diverse domains and representation spaces.

6 Evaluation with Human Users

Next, to go beyond simulated user preference models considered above (u = BTL, u = Consistent, u = GPT), we evaluate the methods with an additional user behavior type, u = Human, with preferences coming from human users.

Evaluation setup. We consider two domains, `food64d` and `visual512d`, due to the high cost involved in this evaluation. We compare three methods (UserAlign, iidBest, and Random) using the same candidate pool sizes as in the earlier experiments (`food64d` has 20 responses per question and `visual512d` has 40 images per question). To ensure a fair comparison and keep cognitive load manageable, we fix the per-domain interaction budget to 10 for `food64d` and 20 for `visual512d`, and

[3] We will additionally look at dynamics when running these algorithms for a fixed number of steps without using ϵ-based stopping; here, the algorithm will resort to i.i.d. sampling of $y^{(2)}$ if no viable response remains.

Table 3: Results of the `food64d` and `visual512d` evaluation with human users under fixed interaction budgets. The table reports win-rate versus the zero-temperature baseline at two different interaction steps for two personalization conditions.

Method	`food64d`				`visual512d`			
	With-persona		**Without-persona**		**With-persona**		**Without-persona**	
	$t = 5$	$t = 10$	$t = 5$	$t = 10$	$t = 10$	$t = 20$	$t = 10$	$t = 20$
RANDOM	44.2 (3.2)	44.2 (3.2)	43.8 (3.2)	43.8 (3.2)	46.9 (3.2)	46.9 (3.2)	42.3 (3.2)	42.3 (3.2)
IIDBEST	71.7 (4.1)	78.3 (3.8)	73.3 (4.0)	76.7 (3.9)	75.6 (3.9)	81.5 (3.6)	78.3 (3.8)	79.2 (3.7)
USERALIGN	82.5 (3.5)	89.2 (2.8)	79.2 (3.7)	85.8 (3.2)	81.7 (3.5)	90.0 (2.7)	79.8 (3.7)	82.4 (3.5)

we report results at interaction steps $t \in \{5, 10\}$ for `food64d` and at $t \in \{10, 20\}$ for `visual512d`. Personalization is varied with two conditions to test robustness to how preferences are specified. In the with-persona condition, participants see a concise persona and are asked to roleplay it (similar to Section 5). In the without-persona condition, no persona is shown and participants choose according to their own preferences, reflecting greater individual variability and a more challenging setting. Method identities are blinded to the users throughout.

Web application and participation sessions. We provide an overview of the user study setup below, and full details are available in Appendix E. We developed a web application to expose the methods through an interactive interface. We recruited a total of 960 participants on Amazon Mechanical Turk, split uniformly across the domains, methods, and personalization conditions. Before a participation session began, each participant was randomly assigned a domain (`food64d` or `visual512d`), a question from that domain's question set, an interaction method (USERALIGN or IIDBEST), and a personalization condition (with-persona or without-persona). Each session has three stages. Stage 1 presents the question and the assigned personalization instruction. Stage 2 consists of the fixed number of pairwise comparisons based on the interaction budget mentioned above. In Stage 3, participants compare three final candidates against the zero-temperature baseline for their assigned question: the method's selection at the later interaction step, the selection at the earlier interaction step, and a candidate chosen by the RANDOM method. On average, a session lasted about 7.5 minutes and participants received a compensation of 1.80 USD for each session.

Results. In each domain, for each of USERALIGN and IIDBEST, we collected $n = 120$ evaluation sessions per setting; for RANDOM, which does not depend on interaction steps, we recorded $2n = 240$ samples. Table 3 reports win-rates across both domains and personalization conditions. Under matched interaction budgets, USERALIGN consistently outperforms IIDBEST, with larger gains at the later interaction step in each domain and higher absolute performance in the with-persona condition. These results highlight that the improvements observed for the simulated user behavior in Section 5 also carry over when user preferences are provided by humans.

7 Concluding Discussions

We introduced USERALIGN, a novel inference-time method for efficiently aligning generative model responses to user preferences via sequential pairwise comparisons. Through theoretical analysis and empirical evaluation across text and image generation domains, we demonstrated the efficacy of our method, which provides substantial speed-ups and cost reductions.

Next, we discuss a few limitations of our work and outline a future plan to address them. First, we considered pairwise comparisons as the feedback modality; it would be useful to explore richer forms of user feedback such as rankings or free-form inputs. Second, our method relies on a pre-generated response pool; it would be interesting to study how we can guide the generations adaptively during the preference elicitation process. Finally, our method treats a user's question independently; it would be useful to leverage historical user preferences data, e.g., from previous questions, to reduce interaction cost. As broader implications of our work, we note that adopting personalized alignment methods in AI systems raises ethical considerations, including transparency in how preferences are collected/used and guarding against potential bias or overfitting to noisy feedback. Future deployment of such methods in real-world would require proactive safeguards to ensure that personalization enhances user experience without compromising ethical aspects.

Acknowledgments and Disclosure of Funding

Nachiket Kotalwar did this work during an internship at the Max Planck Institute for Software Systems (MPI-SWS), Germany. Funded/Co-funded by the European Union (ERC, TOPS, 101039090). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. Aligning Large Language Models with Human: A Survey. *CoRR*, abs/2307.12966, 2023.

[2] Stephen Casper et al. Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback. *In Proceedings of Transactions on Machine Learning Research (TMLR)*, 2023.

[3] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[4] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-Tuning Language Models from Human Preferences. *CoRR*, abs/1909.08593, 2019.

[5] Long Ouyang et al. Training language models to follow instructions with human feedback. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[6] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

[7] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model Alignment as Prospect Theoretic Optimization. *CoRR*, abs/2402.01306, 2024.

[8] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[9] Chanwoo Park, Mingyang Liu, Dingwen Kong, Kaiqing Zhang, and Asuman E. Ozdaglar. RLHF from Heterogeneous Feedback via Personalization and Preference Aggregation. *CoRR*, abs/2405.00254, 2024.

[10] Xinyu Li, Zachary C. Lipton, and Liu Leqi. Personalized Language Modeling from Personalized Human Feedback. *CoRR*, abs/2402.05133, 2024.

[11] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Raghavi Chandu, Chandra Bhagavatula, and Yejin Choi. The Unlocking Spell on Base LLMs: Rethinking Alignment via In-Context Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[12] James Y. Huang, Sailik Sengupta, Daniele Bonadiman, Yi'an Lai, Arshit Gupta, Nikolaos Pappas, Saab Mansour, Katrin Kirchhoff, and Dan Roth. DeAL: Decoding-time Alignment for Large Language Models. *CoRR*, abs/2402.06147, 2024.

[13] Yisong Yue and Thorsten Joachims. Interactively Optimizing Information Retrieval Systems as a Dueling Bandits Problem. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[14] Dorsa Sadigh, Anca D. Dragan, Shankar Sastry, and Sanjit A. Seshia. Active Preference-Based Learning of Reward Functions. In *Robotics: Science and Systems XIII, Massachusetts Institute of Technology*, 2017.

[15] Sarah Filippi, Olivier Cappé, Aurélien Garivier, and Csaba Szepesvári. Parametric Bandits: The Generalized Linear Case. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2010.

[16] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved Algorithms for Linear Stochastic Bandits. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2011.

[17] Marc Abeille, Louis Faury, and Clément Calauzènes. Instance-Wise Minimax-Optimal Algorithms for Logistic Bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

[18] Liyuan Xu, Junya Honda, and Masashi Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[19] Junghyun Lee, Se-Young Yun, and Kwang-Sung Jun. Improved Regret Bounds of (Multinomial) Logistic Bandits via Regret-to-Confidence-Set Conversion. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

[20] Nick Littlestone. Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm. *Machine Learning*, 1987.

[21] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

[22] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize with human feedback. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

[23] Reiichiro Nakano et al. WebGPT: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021.

[24] Haikang Deng and Colin Raffel. Reward-Augmented Decoding: Efficient Controlled Text Generation With a Unidirectional Reward Model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[25] Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. ARGS: Alignment as Reward-Guided Search. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[26] Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. TreeBoN: Enhancing Inference-Time Alignment with Speculative Tree-Search and Best-of-N Sampling. *CoRR*, abs/2410.16033, 2024.

[27] Nirjhar Das, Souradip Chakraborty, Aldo Pacchiano, and Sayak Ray Chowdhury. Active Preference Optimization for Sample Efficient RLHF. *CoRR*, abs/2402.10500, 2024.

[28] Kaixuan Ji, Jiafan He, and Quanquan Gu. Reinforcement Learning from Human Feedback with Active Queries. *CoRR*, abs/2402.09401, 2024.

[29] Adam X. Yang, Maxime Robeyns, Thomas Coste, Jun Wang, Haitham Bou-Ammar, and Laurence Aitchison. Bayesian Reward Models for LLM Alignment. *CoRR*, abs/2402.13210, 2024.

[30] Zichen Liu, Changyu Chen, Chao Du, Wee Sun Lee, and Min Lin. Sample-Efficient Alignment for LLMs. *CoRR*, abs/2411.01493, 2024.

[31] William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active Preference Learning for Large Language Models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[32] Vikranth Dwaracherla, Seyed Mohammad Asghari, Botao Hao, and Benjamin Van Roy. Efficient Exploration for LLMs. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[33] Luckeciano Carvalho Melo, Panagiotis Tigas, Alessandro Abate, and Yarin Gal. Deep Bayesian Active Learning for Preference Modeling in Large Language Models. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[34] Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. PAD: Personalized Alignment at Decoding-time. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[35] Mingye Zhu, Yi Liu, Lei Zhang, Junbo Guo, and Zhendong Mao. On-the-fly Preference Alignment via Principle-Guided Decoding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[36] Zhaowei Zhang, Fengshuo Bai, Qizhi Chen, Chengdong Ma, Mingzhi Wang, Haoran Sun, Zilong Zheng, and Yaodong Yang. Amulet: ReAlignment During Test Time for Personalized Preference Adaptation of LLMs. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[37] Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing Reinforcement Learning from Human Feedback with Variational Preference Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[38] Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hannaneh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized Soups: Personalized Large Language Model Alignment via Post-hoc Parameter Merging. *CoRR*, abs/2310.11564, 2023.

[39] Ruizhe Shi, Yifang Chen, Yushi Hu, Alisa Liu, Hanna Hajishirzi, Noah A. Smith, and Simon S. Du. Decoding-Time Language Model Alignment with Multiple Objectives. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[40] Avinandan Bose, Zhihan Xiong, Yuejie Chi, Simon Shaolei Du, Lin Xiao, and Maryam Fazel. LoRe: Personalizing LLMs via Low-Rank Reward Modeling. *CoRR*, abs/2504.14439, 2025.

[41] Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. PAL: Sample-Efficient Personalized Reward Modeling for Pluralistic Alignment. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

[42] Javier González, Zhenwen Dai, Andreas C. Damianou, and Neil D. Lawrence. Preferential Bayesian Optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[43] Minhyeon Oh, Seungjoon Lee, and Jungseul Ok. Active Preference-based Learning for Multi-dimensional Personalization. *CoRR*, abs/2411.00524, 2024.

[44] Zhouhang Xie et al. A Survey on Personalized and Pluralistic Preference Alignment in Large Language Models. *CoRR*, abs/2504.07070, 2025.

[45] Jian Guan, Junfei Wu, Jia-Nan Li, Chuanqi Cheng, and Wei Wu. A Survey on Personalized Alignment - The Missing Piece for Large Language Models in Real-World Applications. *CoRR*, abs/2503.17003, 2025.

[46] Erdem Biyik and Dorsa Sadigh. Batch Active Preference-Based Learning of Reward Functions. In *Proceedings of the Annual Conference on Robot Learning (CoRL)*, 2018.

[47] Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The K-armed Dueling Bandits Problem. In *Proceedings of the Annual Conference on Learning Theory (COLT)*, 2009.

[48] Masrour Zoghi, Shimon Whiteson, Rémi Munos, and Maarten de Rijke. Relative Upper Confidence Bound for the K-Armed Dueling Bandit Problem. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2014.

[49] Aadirupa Saha and Pierre Gaillard. Versatile Dueling Bandits: Best-of-both World Analyses for Learning from Relative Preferences. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2022.

[50] Louis Faury, Marc Abeille, Clément Calauzènes, and Olivier Fercoq. Improved Optimistic Algorithms for Logistic Bandits. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020.

[51] OpenAI. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/, 2024.

[52] Model2Vec: Turn any Sentence Transformer into a Small Fast Model. https://github.com/MinishLab/model2vec, 2024.

[53] Gabriel Ilharco et al. OpenCLIP. https://doi.org/10.5281/zenodo.5143773, 2021.

[54] Samir Yitzhak Gadre et al. DataComp: In search of the next generation of multimodal datasets. *CoRR*, abs/2304.14108, 2023.

[55] OpenAI. GPT-4o mini: Advancing Cost-efficient Intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/, 2024.

[56] OpenAI. GPT Image 1. https://platform.openai.com/docs/models/gpt-image-1, 2025.

[57] Alec Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[58] Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. Everyone Deserves A Reward: Learning Customized Human Preferences. *CoRR*, abs/2309.03126, 2023.

[59] Jack Sherman and Winifred J Morrison. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 1950.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Introduction (Section 1) introduce USERALIGN, its bandit-based formulation, theoretical guarantees, and empirical advantages, which match the results in Section 4.1, Section 5, and Section 6.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss limitations in Section 7.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: All assumptions are stated in Section 4 and proofs are provided in appendices.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 provides the details for the experimental evaluation.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide open access code and data in a Github repository (Section 1).

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify these experimental details in Section 5.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: In Section 5, we state that we use multiple random seeds and average across seeds, user personas, and questions. Figures 3 and 4 shade the standard error computed using Python libraries, so that the variance information is visible. Our evaluation with human users (Section 6) likewise reports win-rates with standard errors (see Table 3).

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

Justification: We include details about compute resources in the appendices.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our work follows the NeurIPS Code of Ethics by adhering to research-ethics norms, discussing risks of bias/privacy, and briefly outlining mitigation strategies (Section 7).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Section 7 addresses both societal risks and opportunities.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We do not release a foundation model or sensitive dataset. USERALIGN is a post-hoc decision algorithm run over responses generated by other models.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: We use and cite external models and datasets only for evaluation purposes. As we do not modify or redistribute these assets, we do not explicitly mention their licenses.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code and created data under the CC BY-NC-SA 4.0 license. Full documentation is provided alongside these assets.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Section 6 describes the participant setup and compensation. Appendix E includes interface screenshots and the full on-screen instructions (plain text).

15. **Institutional Review Board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: We have provided details about the instructions shown to participants in Appendix E, and the study followed a standard protocol for eliciting human annotations with participants recruited from crowdsourcing platforms. There were no specific risks involved in this study, and the participants were fully informed about the study details.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

A Table of Contents

In this section, we briefly describe the content provided in the paper's appendices.

- Section B provides further results for `food2d`, showcasing a worked example of personalized alignment and evaluations with GPT-based simulated users.
- Section C presents win-rate vs. cost trade-off results in tabular form, then offers a brief head-to-head comparison of the methods.
- Section D gives further details about the procedures for generating candidate response pools and reports experiments with unbiased response pools.
- Section E provides additional details about the web-application used for human evaluation.
- Section F provides a breakdown of API usage, computational efficiency, and costs.
- Section G contains complete proofs for the theoretical results in the main paper.

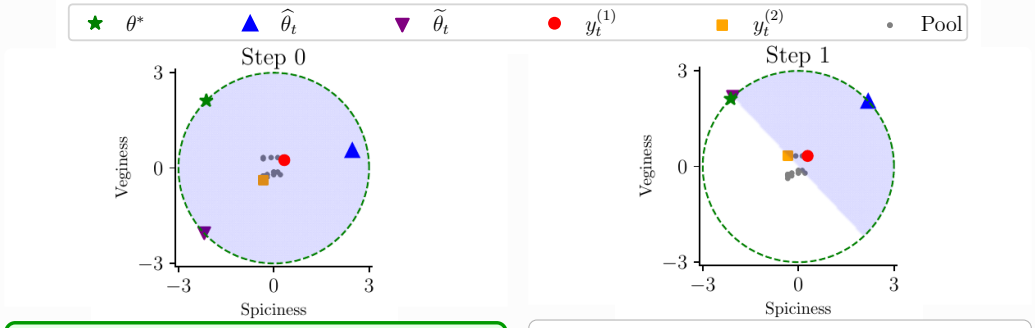
B Additional Results for `food2d` domain

1 Input: What should I cook for dinner tonight?

2 System generates a diverse pool of responses:

1. How about trying a creamy garlic chicken pasta with a side of roasted vegetables?
2. How about trying a garlic butter shrimp pasta with a fresh green salad on the side?
3. How about trying a creamy garlic Tuscan chicken with spinach and sun-dried tomatoes, served over pasta or rice?
4. How about trying a delicious and easy-to-make chicken stir-fry with vegetables and rice?
5. How about trying a delicious chicken stir-fry with vegetables and rice for dinner tonight?
6. How about trying a creamy garlic chicken with sautéed spinach and mashed potatoes for a comforting and delicious meal?
7. How about a comforting chicken stir-fry with vegetables and rice for a quick and delicious dinner?
8. How about trying a creamy garlic parmesan chicken with roasted vegetables on the side?
9. How about trying a creamy garlic parmesan chicken with roasted vegetables and a side of mashed potatoes?
10. How about trying a creamy garlic parmesan chicken with roasted vegetables on the side?
11. How about a spicy chickpea and vegetable curry with a medley of bell peppers, zucchini, and spinach, served over basmati rice?
12. How about a comforting vegetable risotto with fresh herbs like basil and parsley, where the creamy arborio rice perfectly ...
13. How about a spicy roasted vegetable stir-fry with a hint of chili, served over quinoa for a nutritious and flavorful plant-based meal?
14. How about a comforting vegetable risotto with creamy Arborio rice, sautéed mushrooms, and tender asparagus, finished with a sprinkle ...
15. How about a hearty roast chicken with garlic mashed potatoes and a side of steamed broccoli for a satisfying, protein-rich meal?
16. How about a juicy ribeye steak with a side of garlic butter mushrooms for a rich, savory meal?
17. How about sizzling chicken fajitas with a touch of jalapeño for that perfect hint of heat, served alongside a vibrant mix of ...
18. How about a spicy chicken stir-fry with bell peppers and a hint of garlic, served over jasmine rice for a satisfying yet mildly fiery meal?
19. Ignite your taste buds with a fiery Thai green curry loaded with fresh vegetables and a generous splash of chili heat.
20. How about a comforting vegetable stir-fry with broccoli, bell peppers, and tofu, served over fluffy jasmine rice, seasoned with ...

3 USERALIGN eliciting user preferences via pairwise comparisons:



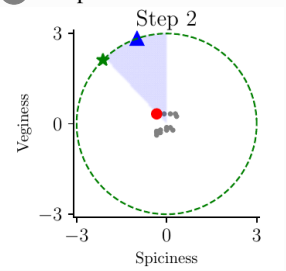
Ignite your taste buds with a fiery Thai green curry loaded with fresh vegetables and a generous splash of chili heat.

How about a spicy chickpea and vegetable curry with a medley of bell peppers, zucchini, and spinach, served over basmati rice?

How about a juicy ribeye steak with a side of garlic butter mushrooms for a rich, savory meal?

How about a comforting vegetable risotto with fresh herbs like basil and parsley, where the creamy arborio rice perfectly complements the flavors of sautéed mushrooms, peas, and asparagus?

4 Output



How about a comforting vegetable risotto with fresh herbs like basil and parsley, where the creamy arborio rice perfectly complements the flavors of sautéed mushrooms, peas, and asparagus?

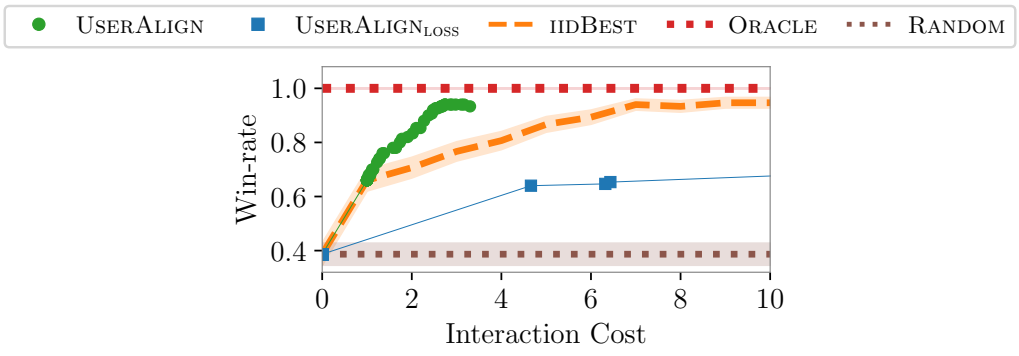
Figure 5: Illustrative example in the `foods2d` domain showing inference-time personalized alignment (see also Figures 1 and 2). In Stage 3, $y_t^{(1)}$ appears above $y_t^{(2)}$ at each step (preferred highlighted). The user is simulated by GPT-4o-mini, conditioned on the persona: "*A plant-based eater who avoids all heat and meat (spiciness:* -1.0*, veginess:* 1.0*), preferring gentle, nourishing dishes with no spice.*"

B.1 Illustrative Example of Personalized Alignment

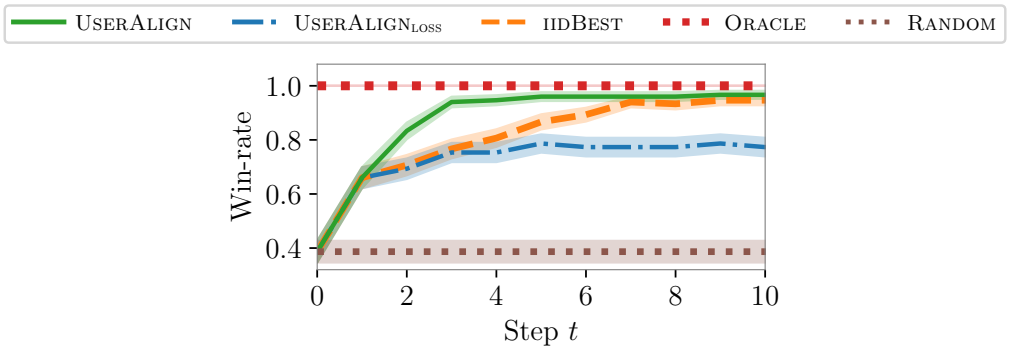
Figure 5 provides a concrete step-by-step illustration of inference-time personalized alignment in the `foods2d` domain. In addition to showing each stage of the interaction (i.e, the user question, candidate response generation, preference elicitation, and final response selection; also see Figure 1), the figure also visualizes how the version space is progressively reduced throughout the process (also see Figure 2). This integrated example highlights both the workflow of UserAlign and the mechanism by which it narrows the set of plausible user preferences.

B.2 Empirical Results with GPT-based Users

Figure 6 shows results on `food2d` with GPT-based simulated users (also see Figure 3). UserAlign consistently achieves high win-rate at a low interaction cost, confirming that UserAlign remains effective in the 2D setting even when user preferences are simulated by a large language model.



(a) Win-rate vs. interaction cost



(b) Win-rate w.r.t. increasing steps

Figure 6: Results on `food2d` for u = GPT. Top plot shows win-rate vs. interaction cost trade-off; here UserAlign and UserAlign$_{\text{Loss}}$ results show scatter plot corresponding to varying values of ϵ. Bottom plot shows win-rate per increasing interaction steps; here UserAlign and UserAlign$_{\text{Loss}}$ are run for a given number of steps as mentioned in Footnote 3. In these plots, Oracle and Random are flat lines, and iidBest results are reported at different number of given steps.

C Results in Tabular Form

Tables 4 and 5 offer additional details regarding the results, complementary to Figure 4. The number of steps is capped at 199. Colored entries for UserAlign and UserAlign$_{\text{Loss}}$ correspond to $\epsilon = 0.0$, the default and most interpretable alignment setting. For iidBest, highlighted entries are chosen to match the interaction cost of the main UserAlign configuration, enabling direct comparison. Oracle and Random are included as fixed reference baselines. UserAlign consistently reaches high win-rates at low interaction cost, especially as the stopping threshold ϵ decreases, confirming its efficiency in identifying user-aligned responses with minimal queries. The reported standard errors are small, indicating reliable and stable performance.

Table 6 complements these results with direct head-to-head win-rates of UserAlign versus iidBest. These results confirm that the superior win-rates observed against a fixed baseline also extend to one-on-one comparisons across all domains.

Table 4: Win-rate vs. interaction cost trade-off on domains with arbitrary preference spaces (`food64d` and `travel64d` domains). Results are presented as mean (sem), complementing the results shown in Figure 4. For readability, we report win-rates as percentages.

Method	`food64d`		`travel64d`	
	Win-rate (%)	Cost	Win-rate (%)	Cost
UserAlign ($\epsilon = 3.0$)	54.00 (4.07)	0.00 (0.00)	58.67 (4.02)	0.00 (0.00)
UserAlign ($\epsilon = 2.0$)	74.67 (3.55)	1.00 (0.00)	74.00 (3.58)	1.00 (0.00)
UserAlign ($\epsilon = 1.0$)	84.67 (2.94)	2.19 (0.07)	76.00 (3.49)	1.95 (0.07)
UserAlign ($\epsilon = 0.5$)	100.00 (0.00)	11.39 (0.23)	94.67 (1.83)	11.57 (0.23)
UserAlign ($\epsilon = 0.4$)	100.00 (0.00)	13.47 (0.22)	97.33 (1.32)	13.82 (0.22)
UserAlign ($\epsilon = 0.3$)	98.67 (0.94)	15.49 (0.25)	100.00 (0.00)	16.00 (0.23)
UserAlign ($\epsilon = 0.2$)	97.33 (1.32)	16.85 (0.29)	100.00 (0.00)	18.31 (0.29)
UserAlign ($\epsilon = 0.1$)	98.67 (0.94)	18.28 (0.30)	100.00 (0.00)	19.67 (0.23)
UserAlign ($\epsilon = 0.0$)	98.67 (0.94)	19.13 (0.24)	100.00 (0.00)	20.69 (0.21)
UserAlign$_{\text{Loss}}$ ($\epsilon = 3.0$)	54.00 (4.07)	0.00 (0.00)	58.67 (4.02)	0.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 2.0$)	74.67 (3.55)	1.00 (0.00)	74.00 (3.58)	1.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 1.0$)	82.67 (3.09)	180.52 (4.70)	76.00 (3.49)	195.04 (2.26)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.5$)	84.67 (2.94)	199.00 (0.00)	76.67 (3.45)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.4$)	84.67 (2.94)	199.00 (0.00)	76.67 (3.45)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.3$)	84.67 (2.94)	199.00 (0.00)	76.67 (3.45)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.2$)	84.67 (2.94)	199.00 (0.00)	76.67 (3.45)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.1$)	84.67 (2.94)	199.00 (0.00)	76.67 (3.45)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.0$)	84.67 (2.94)	199.00 (0.00)	76.67 (3.45)	199.00 (0.00)
iidBest ($t = 0$)	54.00 (4.07)	0.00 (0.00)	58.67 (4.02)	0.00 (0.00)
iidBest ($t = 5$)	87.33 (2.72)	5.00 (0.00)	78.00 (3.38)	5.00 (0.00)
iidBest ($t = 10$)	88.67 (2.59)	10.00 (0.00)	86.67 (2.78)	10.00 (0.00)
iidBest ($t = 15$)	90.00 (2.45)	15.00 (0.00)	85.33 (2.89)	15.00 (0.00)
iidBest ($t = 20$)	90.00 (2.45)	20.00 (0.00)	85.33 (2.89)	20.00 (0.00)
iidBest ($t = 25$)	90.00 (2.45)	25.00 (0.00)	87.33 (2.72)	25.00 (0.00)
iidBest ($t = 50$)	92.00 (2.22)	50.00 (0.00)	88.00 (2.65)	50.00 (0.00)
iidBest ($t = 75$)	92.67 (2.13)	75.00 (0.00)	90.00 (2.45)	75.00 (0.00)
iidBest ($t = 100$)	92.67 (2.13)	100.00 (0.00)	90.00 (2.45)	100.00 (0.00)
iidBest ($t = 150$)	93.33 (2.04)	150.00 (0.00)	91.33 (2.30)	150.00 (0.00)
iidBest ($t = 199$)	93.33 (2.04)	199.00 (0.00)	90.67 (2.38)	199.00 (0.00)
Oracle	100.00 (0.00)	0.00 (0.00)	100.00 (0.00)	0.00 (0.00)
Random	54.00 (4.07)	0.00 (0.00)	58.67 (4.02)	0.00 (0.00)

Table 5: Win-rate vs. interaction cost trade-off on domains with arbitrary preference spaces (`visual512d` and `dsp64d` domains). Results are presented as mean (sem), complementing the results shown in Figure 4. For readability, we report win-rates as percentages.

Method	`visual512d`		`dsp64d`	
	Win-rate (%)	Cost	Win-rate (%)	Cost
UserAlign ($\epsilon = 3.0$)	40.67 (4.01)	0.00 (0.00)	51.73 (1.29)	0.00 (0.00)
UserAlign ($\epsilon = 2.0$)	58.00 (4.03)	1.00 (0.00)	65.33 (1.23)	1.00 (0.00)
UserAlign ($\epsilon = 1.0$)	65.33 (3.89)	7.58 (0.25)	70.73 (1.17)	1.49 (0.02)
UserAlign ($\epsilon = 0.5$)	86.00 (2.83)	34.09 (0.68)	84.80 (0.93)	5.03 (0.07)
UserAlign ($\epsilon = 0.4$)	92.00 (2.22)	41.12 (0.74)	88.53 (0.82)	7.53 (0.09)
UserAlign ($\epsilon = 0.3$)	93.33 (2.04)	47.57 (0.71)	91.33 (0.73)	11.53 (0.11)
UserAlign ($\epsilon = 0.2$)	94.67 (1.83)	49.31 (0.72)	94.67 (0.58)	17.07 (0.13)
UserAlign ($\epsilon = 0.1$)	94.67 (1.83)	49.31 (0.72)	96.00 (0.51)	20.58 (0.14)
UserAlign ($\epsilon = 0.0$)	94.67 (1.83)	49.31 (0.72)	96.27 (0.49)	21.12 (0.14)
UserAlign$_{\text{Loss}}$ ($\epsilon = 3.0$)	40.67 (4.01)	0.00 (0.00)	51.73 (1.29)	0.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 2.0$)	58.00 (4.03)	1.00 (0.00)	65.33 (1.23)	1.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 1.0$)	59.33 (4.01)	199.00 (0.00)	71.20 (1.17)	164.82 (1.93)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.5$)	59.33 (4.01)	199.00 (0.00)	72.47 (1.15)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.4$)	59.33 (4.01)	199.00 (0.00)	72.47 (1.15)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.3$)	59.33 (4.01)	199.00 (0.00)	72.47 (1.15)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.2$)	59.33 (4.01)	199.00 (0.00)	72.47 (1.15)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.1$)	59.33 (4.01)	199.00 (0.00)	72.47 (1.15)	199.00 (0.00)
UserAlign$_{\text{Loss}}$ ($\epsilon = 0.0$)	59.33 (4.01)	199.00 (0.00)	72.47 (1.15)	199.00 (0.00)
IIDBest ($t = 0$)	40.67 (4.01)	0.00 (0.00)	51.73 (1.29)	0.00 (0.00)
IIDBest ($t = 5$)	65.33 (3.89)	5.00 (0.00)	75.33 (1.11)	5.00 (0.00)
IIDBest ($t = 10$)	68.00 (3.81)	10.00 (0.00)	77.87 (1.07)	10.00 (0.00)
IIDBest ($t = 15$)	69.33 (3.76)	15.00 (0.00)	79.73 (1.04)	15.00 (0.00)
IIDBest ($t = 20$)	77.33 (3.42)	20.00 (0.00)	81.60 (1.00)	20.00 (0.00)
IIDBest ($t = 25$)	74.67 (3.55)	25.00 (0.00)	81.67 (1.00)	25.00 (0.00)
IIDBest ($t = 50$)	84.00 (2.99)	50.00 (0.00)	82.47 (0.98)	50.00 (0.00)
IIDBest ($t = 75$)	83.33 (3.04)	75.00 (0.00)	83.60 (0.96)	75.00 (0.00)
IIDBest ($t = 100$)	86.67 (2.78)	100.00 (0.00)	84.27 (0.94)	100.00 (0.00)
IIDBest ($t = 150$)	86.00 (2.83)	150.00 (0.00)	84.87 (0.93)	150.00 (0.00)
IIDBest ($t = 199$)	84.67 (2.94)	199.00 (0.00)	84.47 (0.94)	199.00 (0.00)
Oracle	100.00 (0.00)	0.00 (0.00)	99.33 (0.21)	0.00 (0.00)
Random	40.67 (4.01)	0.00 (0.00)	51.73 (1.29)	0.00 (0.00)

Table 6: Head-to-head win-rate (%) comparison between UserAlign ($\epsilon = 0.0$) and IIDBest across four domains. Results are presented as mean (sem).

Comparison	Win-rate (%)			
	`food64d`	`travel64d`	`visual512d`	`dsp64d`
UserAlign vs. IIDBest ($t = 20$)	96.67 (1.47)	93.33 (2.04)	85.33 (2.89)	91.00 (0.74)
UserAlign vs. IIDBest ($t = 25$)	97.33 (1.32)	95.33 (1.72)	85.33 (2.89)	90.33 (0.76)
UserAlign vs. IIDBest ($t = 50$)	96.67 (1.47)	95.33 (1.72)	87.33 (2.72)	90.07 (0.77)

D Pool Generation Details and Additional Experiments

D.1 Details about Pool Generation Used in Section 5

Below we provide details about the pool generation that was used for evaluation in Section 5.

Pool generation procedure for `food64d`, `travel64d`, and `visual512d`. As introduced in Section 5.1, for each question, we generate a diverse pool of candidate responses by following two sampling approaches.

- We obtain half of the candidate responses by sampling from GPT-4o or GPT-Image-1 temperature 0.5, conditioned only on the question, resulting in unbiased responses. We generate 10 responses per question in `food64d` and `travel64d`, and 20 in `visual512d`.
- We obtain the remaining half by prompting the generative model to first reason about possible diverse interests relevant to the question, at temperature 0.8. For each generated interest, we then prompt GPT-4o (for text) or GPT-Image-1 (for vision) at temperature 0.5 to produce a candidate response. We generate 10 interests per question in `food64d` and `travel64d`, and 20 in `visual512d`, with one response generated per interest.

Pool generation procedure for `dsp64d`. Similar to the procedure discussed above, here we also generate a diverse pool of candidate responses by following slightly different approaches.

- We obtain half of the candidate responses by sampling from GPT-4o at temperature 0.5, conditioned only on the question, resulting in unbiased responses. We generate 10 responses per question.
- We obtain the remaining half by using the interests provided in the DSP dataset. We generate 5 responses for each interest using GPT-4o at temperature 0.5, and from this collection, we sample 10 responses to obtain the remaining half.

D.2 Additional Experiments with Different Candidate Pools

Below we provide evaluation results for experiments using different candidate pool to further assess the utility and robustness of our method.

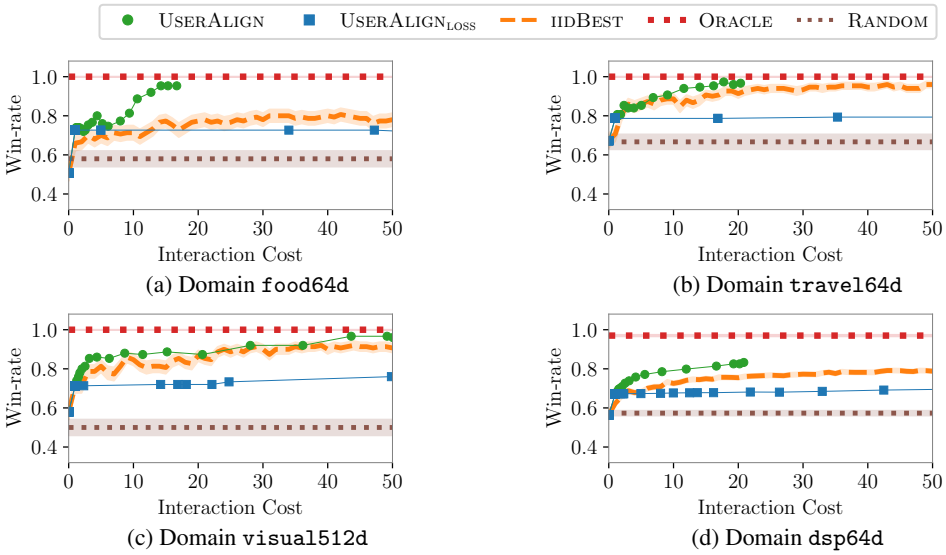
Unbiased response pool construction. For each question, we generate a pool of candidate responses by following one sampling approach. More concretely, responses are sampled from GPT-4o or GPT-Image-1 (at temperature 0.5), conditioned only on the question. The pool size remains the same as in the main experiments, i.e., 20 for text domains and 40 for the image-based domain.

Results. Figure 7 summarizes the results with these pools of unbiased responses. USERALIGN continues to achieve high win-rate at low interaction cost. These results demonstrate that USERALIGN remains effective and outperform baselines also when applied to this variant of candidate pools.

D.3 Practicality of Pairwise Comparisons vs. Best-of-N Selection

As Best-of-N requires users to scan a potentially large pool and pick a single winner, this approach becomes impractical as N grows. Pairwise comparisons reduce this burden by focusing on one local decision at a time. Our approach relies solely on pairwise feedback, avoids repeating pairs, and maintains an incumbent that is continually challenged, encouraging early exploration and later convergence.

To demonstrate this, we experimented with increasing N. Table 7 shows the performance of USERALIGN at $N{=}1000$, where a usable Best-of-N interface is no longer realistic. USERALIGN maintains high win-rates, demonstrating scalability without increasing user effort.



(a) Domain `food64d` (b) Domain `travel64d`

(c) Domain `visual512d` (d) Domain `dsp64d`

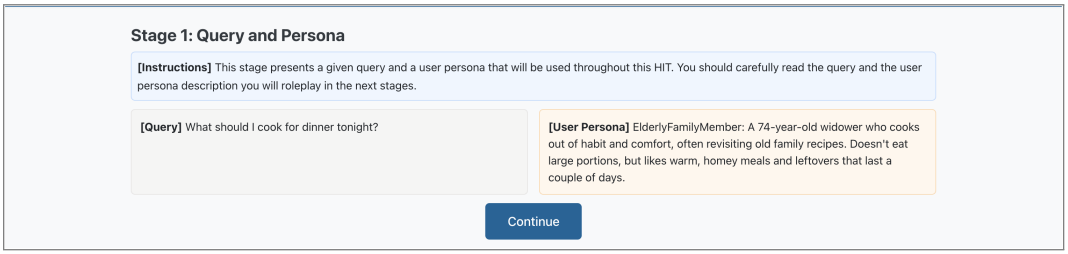
Figure 7: Win-rate vs. interaction cost trade-off on domains with arbitrary preference spaces with fully unbiased pools. Here UserAlign and UserAlign$_{\text{loss}}$ results show scatter plot corresponding to varying values of ϵ. Oracle and Random are flat lines, and iidBest results are reported at different number of given steps.

Table 7: Win-rate (%, sem) on `food64d` across interaction budgets t for a large pool size $N = 1000$.

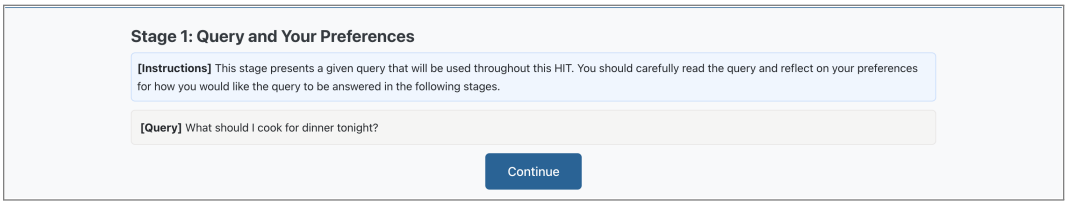
Method	$t = 0$	$t = 5$	$t = 10$	$t = 20$
Random	52.00 (4.09)	52.00 (4.09)	52.00 (4.09)	52.00 (4.09)
iidBest	52.00 (4.09)	72.00 (3.68)	76.00 (3.50)	82.67 (3.10)
UserAlign	52.00 (4.09)	86.67 (2.78)	90.67 (2.38)	96.67 (1.47)

E Additional Details for Evaluation with Human Users

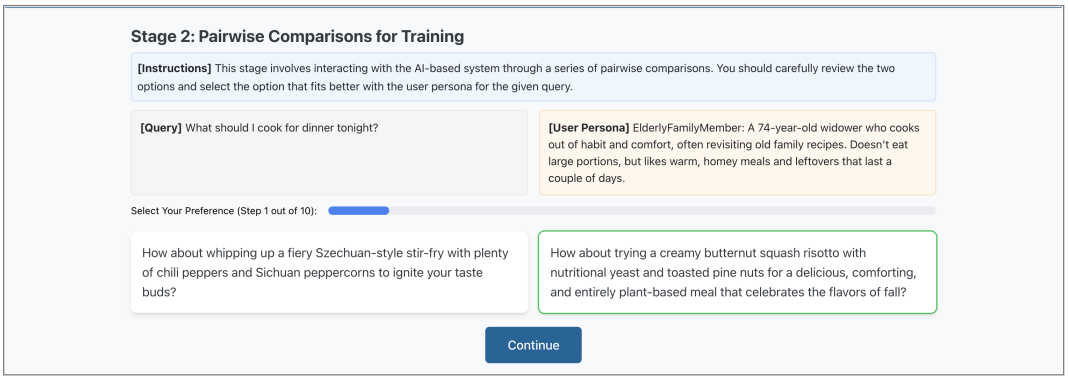
Figures 8 and 9 expand on the web application interface described in Section 6. Figure 8 illustrates the `food64d` text-based workflow, showing the Stage 1 onboarding (for both personalization conditions), the Stage 2 comparisons, and the Stage 3 evaluation against baseline. Figure 9 presents the corresponding `visual512d` image-based workflow.



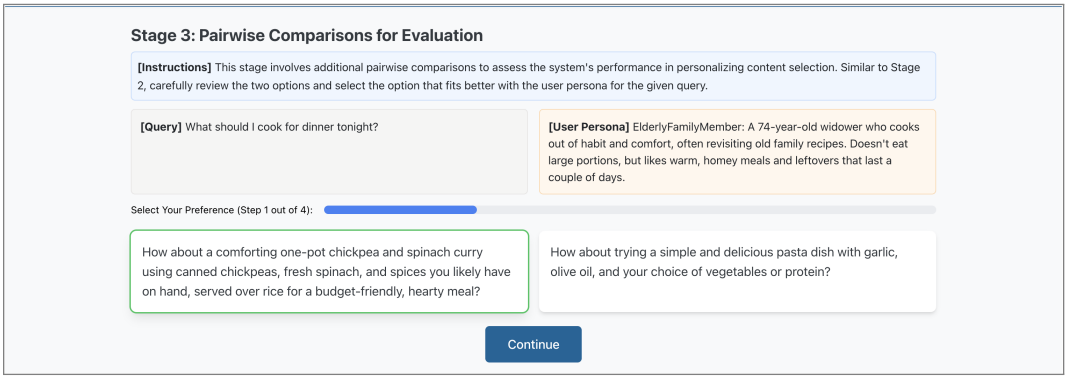
(a) Stage 1 for the `food64d` domain and with-persona condition.



(b) Stage 1 for the `food64d` domain and without-persona condition.

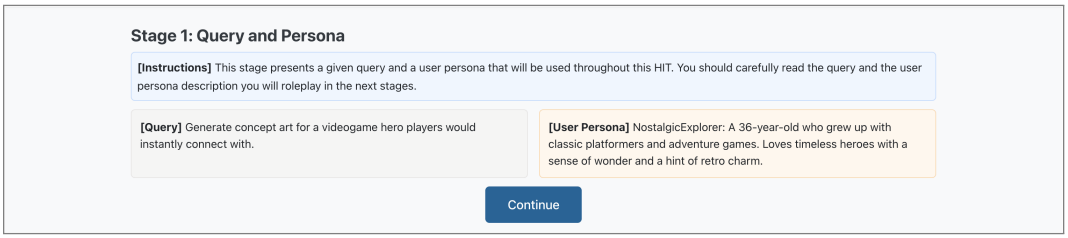


(c) Stage 2 for the `food64d` domain and with-persona condition.

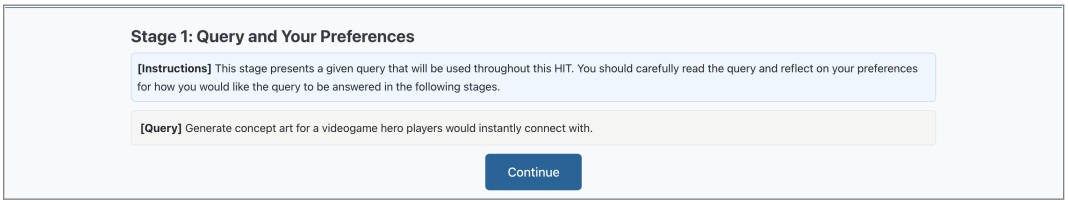


(d) Stage 3 for the `food64d` domain and with-persona condition.

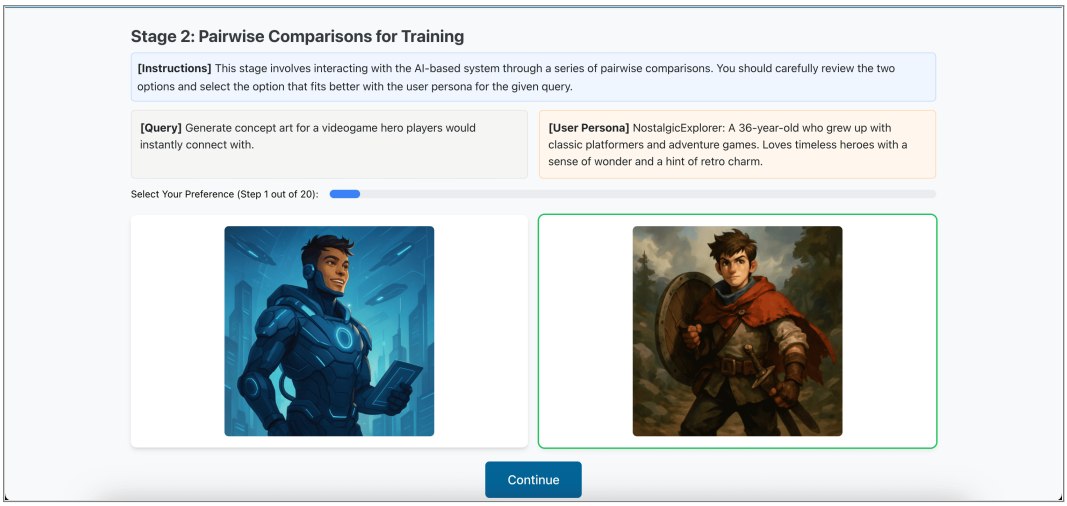
Figure 8: Screenshots from the web application for the `food64d` text-based workflow.



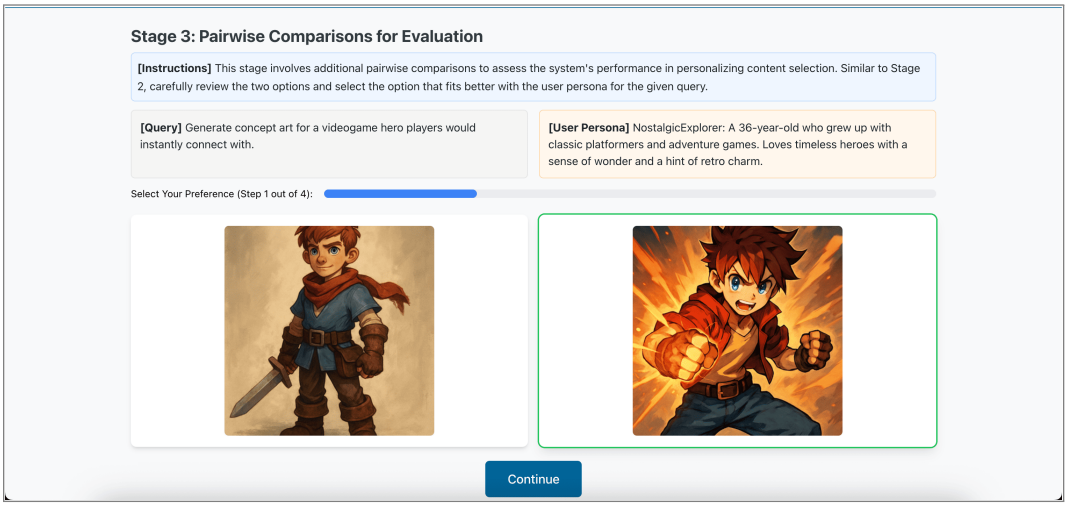
(a) Stage 1 for the `visual512d` domain and with-persona condition.



(b) Stage 1 for the `visual512d` domain and without-persona condition.



(c) Stage 2 for the `visual512d` domain and with-persona condition.



(d) Stage 3 for the `visual512d` domain and with-persona condition.

Figure 9: Screenshots from the web application for the `visual512d` image-based workflow.

E.1 Flow and On-Screen Instructions (Plain Text)

Landing. Participants first see an overview of the study and the system. The screen shows: "Here, you will interact with an AI-based system. The system aims to provide personalized content for a given query and a user persona. The system will infer user preferences through pairwise comparisons while interacting with you. We are conducting this study as part of a research project".

Stage 1. First, participants see the query and the personalization condition. The screen shows the instructions, the query and, in the with-persona condition, also the user persona card; in the without-persona condition, it shows only the query. In the with-persona condition, the header shows "Query and Persona", and the instructions read: "This stage presents a given query and a user persona that will be used throughout this HIT. Please read the query and the persona you will roleplay in the next stages". In the without-persona condition, the header shows "Query and Your Preferences", and the instructions read: "This stage presents a given query that will be used throughout this HIT. Please read the query and reflect on your preferences for how you would like the query to be answered in the following stages".

Stage 2. Next, participants perform pairwise comparisons. On top, the screen shows the instructions, the query and, in the with-persona condition, also the user persona card; in the without-persona condition, it shows only the query. The header shows "Pairwise Comparisons for Training". In the with-persona condition, the instructions read: "Review the two options and select the one that better fits the user persona for the given query". In the without-persona condition, the instructions read: "Review the two options and select the one that best fits your own preferences for the given query".

Stage 3. Finally, participants complete additional pairwise comparisons for evaluation. On top, the screen shows the instructions, the query and, in the with-persona condition, also the user persona card; in the without-persona condition, it shows only the query. The header shows "Pairwise Comparisons for Evaluation". In this stage, participants compare, one matchup at a time, the system's selected candidates at the two chosen interaction steps and a RANDOM candidate against the shared zero-temperature baseline. In the with-persona condition, the instructions read: "Review the two options and select the one that better fits the user persona for the given query". In the without-persona condition, the instructions read: "Review the two options and select the one that best aligns with your own preferences for the given query".

F Wall-clock, Compute, and API Costs

All experiments ran on a compute node with dual AMD EPYC 7702 64-core processors (128 cores total) and 2TB DDR4 ECC memory (2933MHz).

F.1 Computational Efficiency of USERALIGN

The optimization problem of computing $\widehat{\theta}_t$ in Eq. (2) is convex, with a convex objective $\mathcal{L}_t(\cdot)$ and a convex constraint set $\{\theta \in \mathbb{R}^d : \|\theta\|_2 \leq S\}$. Moreover, the confidence set Θ_t defined in Eq. (3) is also convex. The first response $y_t^{(1)} \leftarrow \arg\max_{y \in \mathcal{Y}_{\text{cand}}} \langle \widehat{\theta}_t, \phi(x, y) \rangle$ can be computed via $|\mathcal{Y}_{\text{cand}}|$ inner product evaluations. The second response $(y_t^{(2)}, \widetilde{\theta}_t) \leftarrow \arg\max_{(y', \theta) \in \mathcal{Y}_{\text{cand}} \times \Theta_t} \langle \theta, \phi(x, y') - \phi(x, y_t^{(1)}) \rangle$ requires solving $|\mathcal{Y}_{\text{cand}}|$ convex optimization problems: for each $y' \in \mathcal{Y}_{\text{cand}}$, solve $\widetilde{\theta}_t(y') \leftarrow \arg\max_{\theta \in \Theta_t} \langle \theta, \phi(x, y') - \phi(x, y_t^{(1)}) \rangle$, which has a linear objective and convex constraints. Then, select $y_t^{(2)} \leftarrow \arg\max_{y' \in \mathcal{Y}_{\text{cand}}} \langle \widetilde{\theta}_t(y'), \phi(x, y') - \phi(x, y_t^{(1)}) \rangle$ via another $|\mathcal{Y}_{\text{cand}}|$ inner product evaluations.

Table 8 reports average wall-time for a single step t and its breakdown. As can be seen from these results, the wall-time in a single step is under a second, making it usable for real-world settings.

Table 8: Average wall-time for a step t for USERALIGN across representative domains, broken down by the different computations done by the algorithm. Results are presented as mean values in seconds.

Domain	Wall-time Total (s)	Wall-time $\widehat{\theta}_t$ (s)	Wall-time $y_t^{(1)}$ (s)	Wall-time $y_t^{(2)}$ (s)
`food64d` ($\lvert\mathcal{Y}_{\text{cand}}\rvert$ =20)	0.1287	0.0100	0.0002	0.1187
`visual512d` ($\lvert\mathcal{Y}_{\text{cand}}\rvert$ =40)	0.8719	0.0328	0.0003	0.8392

F.2 API and Embedding Costs

Next, we give more details about the costs related to API calls and embedding. All costs are for a single run (steps capped at 49) of USERALIGN with $\epsilon = 0$, one question, and one GPT-based simulated user.

`food64d` **domain.** Candidate pool generation with GPT-4o cost $0.01; embedding all candidates was done locally with Potion [52] and took 0.01 seconds. The pairwise comparison stage with GPT-4o-mini cost $0.01.

`visual512d` **domain.** Candidate pool generation with GPT-Image-1 cost $1.71; embedding with OpenCLIP [53, 54] was done locally and took 12.13 seconds. The pairwise comparison stage cost $0.12 using GPT-4o-mini.

F.3 Notes About the Solver

All optimization problems are solved with the `cvxpy` Python package using the `CLARABEL` conic solver with default settings. If `CLARABEL` is numerically unstable, the implementation falls back to `ECOS`, and then to `SCS` with `eps=1e-6` and a 50,000 iteration cap.

G Proofs

Proof of Proposition 1. The result follows directly from Theorem 1 of [19]. □

Proof of Theorem 1. Let $\theta^\star \in \Theta_t$. If $\langle \theta^\star, \phi(x, y^\star) - \phi(x, y_\tau^{(1)})\rangle > \epsilon$ holds, i.e., the returned response $y_\tau^{(1)}$ is worse than the best response $y^\star$ by ϵ, then we have:

$$\begin{aligned}\langle \theta^\star, \phi(x, y^\star) - \phi(x, y_\tau^{(1)})\rangle &> \epsilon \\ &\geq \langle \widetilde{\theta}_\tau, \phi(x, y_\tau^{(2)}) - \phi(x, y_\tau^{(1)})\rangle \\ &\geq \langle \theta^\star, \phi(x, y^\star) - \phi(x, y_\tau^{(1)})\rangle,\end{aligned}$$

where the second last inequality is due to the stopping condition of Algorithm 2, and the last inequality is due to $(y_t^{(2)}, \widetilde{\theta}_t) \leftarrow \arg\max_{(y',\theta)\in\mathcal{Y}_{\text{cand}}\times\Theta_t} \langle \theta, \phi(x, y') - \phi(x, y_\tau^{(1)})\rangle$ and $\theta^\star \in \Theta_t$. Note that $\theta^\star \in \Theta_t$ holds with probability at least $1-\delta$ according to Proposition 1. □

Proof Sketch of Theorem 2. The proof involves the following key steps:

- In Lemma 1, we obtain an upper bound on $\|\theta - \widehat{\theta}_t\|_{H_t(\theta^\star)}$.
- In Lemma 2, we obtain an upper bound on $\|z_t\|_{H_t^{-1}(\theta^\star)}$, where $z_t = \phi(x, y_t^{(2)}) - \phi(x, y_t^{(1)})$.
- In Lemma 3, we use the above bounds to upper bound the number of times the pair of responses $(y^{(1)}, y^{(2)})$ is selected before the stopping time τ; then, summing over all the response pairs provides an upper bound on the stopping time τ.

□

Lemma 1. *Let $z_s = \phi(x, y_s^{(2)}) - \phi(x, y_s^{(1)})$ and $H_t(\theta^\star) = \sum_{s=1}^{t-1} \dot{\mu}(\langle \theta^\star, z_s\rangle) z_s z_s^\top + \lambda I_d$ with $\lambda = \frac{1}{4S^2(2+2S)}$. Then, for any $\theta \in \Theta_t$, the following holds with probability at least $1-\delta$:*

$$\|\theta - \widehat{\theta}_t\|_{H_t(\theta^\star)} \leq 2S\sqrt{d\log\left(e + \frac{St}{d}\right) + \log\frac{1}{\delta}}.$$

Proof of Lemma 1. Note that $\theta, \widehat{\theta}_t \in \Theta_t$. By using the triangle inequality, we have:

$$\begin{aligned}\|\theta - \widehat{\theta}_t\|_{H_t(\theta^\star)} &\leq \|\theta - \theta^*\|_{H_t(\theta^\star)} + \|\widehat{\theta}_t - \theta^*\|_{H_t(\theta^\star)} \\ &\leq S\sqrt{d\log\left(e + \frac{St}{d}\right) + \log\frac{1}{\delta}} + S\sqrt{d\log\left(e + \frac{St}{d}\right) + \log\frac{1}{\delta}},\end{aligned}$$

where the last inequality holds with probability at least $1-\delta$ due to Lemma 6 of [19]. □

Lemma 2. *Let $z_s = \phi(x, y_s^{(2)}) - \phi(x, y_s^{(1)})$ and $H_t(\theta^\star) = \sum_{s=1}^{t-1} \dot{\mu}(\langle \theta^\star, z_s\rangle) z_s z_s^\top + \lambda I_d$ with $\lambda = \frac{1}{4S^2(2+2S)}$. Then, we have:*

$$\|z_t\|_{H_t^{-1}(\theta^\star)} \leq \sqrt{\frac{\kappa^\star_{\mathcal{X},\mathcal{Y}}}{|\mathcal{E}_{y_t^{(1)}, y_t^{(2)}}(t-1)|}}$$

Proof of Lemma 2. Let us define $\widetilde{z}_t = \sqrt{\dot{\mu}(\langle \theta^\star, z_t\rangle)} z_t$. Note that $H_t(\theta^\star) = \sum_{s=1}^{t-1} \widetilde{z}_s \widetilde{z}_s^\top + \lambda_t I_d$. Further, we have:

$$\|z_t\|^2_{H_t^{-1}(\theta^\star)} = \frac{1}{\dot{\mu}(\langle \theta^\star, z_t\rangle)} \|\widetilde{z}_t\|^2_{H_t^{-1}(\theta^\star)} \leq \kappa^\star_{\mathcal{X},\mathcal{Y}} \cdot \|\widetilde{z}_t\|^2_{H_t^{-1}(\theta^\star)}.$$

Let $z(y,y') = \phi(x,y') - \phi(x,y)$ and $\widetilde{z}(y,y') = \sqrt{\dot{\mu}(\langle\theta^\star, z(y,y')\rangle)}z(y,y')$. Then, note that $H_t(\theta^\star)$ can be written as follows:

$$\begin{aligned} H_t(\theta^\star) &= \lambda I_d + \sum_{(y,y')\in\mathcal{Y}_{\text{cand}}\times\mathcal{Y}_{\text{cand}}} |\mathcal{E}_{y,y'}(t-1)| \cdot \widetilde{z}(y,y')\widetilde{z}(y,y')^\top \\ &= A + B + C, \end{aligned}$$

where

$$\begin{aligned} A &= \lambda I_d \\ B &= |\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)| \cdot \widetilde{z}(y_t^{(1)},y_t^{(2)})\widetilde{z}(y_t^{(1)},y_t^{(2)})^\top \\ C &= \sum_{(y,y')\in\mathcal{Y}_{\text{cand}}\times\mathcal{Y}_{\text{cand}}\setminus(y_t^{(1)},y_t^{(2)})} |\mathcal{E}_{y,y'}(t-1)| \cdot \widetilde{z}(y,y')\widetilde{z}(y,y')^\top \end{aligned}$$

Note that $\lambda I_d + zz^\top$ is a positive definite matrix in $\mathbb{R}^{d\times d}$ for any $z \in \mathbb{R}^d$ and $\lambda > 0$. Then, by repeatedly applying Sherman-Morrison formula [59] (see Lemma 3 of [18]), we get:

$$\begin{aligned} &\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|^2_{H_t^{-1}(\theta^\star)} \\ &= \widetilde{z}(y_t^{(1)},y_t^{(2)})^\top H_t^{-1}(\theta^\star)\widetilde{z}(y_t^{(1)},y_t^{(2)}) \\ &\le \widetilde{z}(y_t^{(1)},y_t^{(2)})^\top (A+B)^{-1}\widetilde{z}(y_t^{(1)},y_t^{(2)}) \\ &= \widetilde{z}(y_t^{(1)},y_t^{(2)})^\top \left[\frac{I_d^{-1}}{\lambda_t} - \frac{|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|\frac{I_d^{-1}}{\lambda_t}\widetilde{z}(y_t^{(1)},y_t^{(2)})\widetilde{z}(y_t^{(1)},y_t^{(2)})^\top\frac{I_d^{-1}}{\lambda_t}}{1+|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|\widetilde{z}(y_t^{(1)},y_t^{(2)})^\top\frac{I_d^{-1}}{\lambda_t}\widetilde{z}(y_t^{(1)},y_t^{(2)})}\right]\widetilde{z}(y_t^{(1)},y_t^{(2)}) \\ &= \frac{\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^2}{\lambda_t} - \frac{\frac{|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|}{\lambda_t^2}\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^4}{1+\frac{|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|}{\lambda_t}\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^2} \\ &= \frac{\frac{\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^2}{\lambda_t}}{1+\frac{|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|}{\lambda_t}\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^2} \\ &\le \frac{\frac{\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^2}{\lambda_t}}{\frac{|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|}{\lambda_t}\|\widetilde{z}(y_t^{(1)},y_t^{(2)})\|_2^2} \\ &= \frac{1}{|\mathcal{E}_{y_t^{(1)},y_t^{(2)}}(t-1)|}. \end{aligned}$$

□

Lemma 3. *Let τ be the stopping round of* UserAlign$_{\text{Loss}}$. *Further, we define $\Delta_{\mathcal{Y}_{cand}} := \min_{y,y'\in\mathcal{Y}_{cand};y\neq y'}\langle\theta^\star, \phi(x,y) - \phi(x,y')\rangle$. Then, with probability at least $1-\delta$, we have:*

$$\tau \le \frac{4S^2K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{cand}}\}^2}\left[d\log\left(e + \frac{S\tau}{d}\right) + \log\frac{1}{\delta}\right] + K^2.$$

Proof of Lemma 3. Let τ be the stopping time of the algorithm. For any two response $y^{(1)}, y^{(2)} \in \mathcal{Y}_{\text{cand}}$, we define:

$$\mathcal{E}_{y^{(1)},y^{(2)}}(\tau) := \left\{\widetilde{t} \in [\tau] : \text{responses } y^{(1)} \text{ and } y^{(2)} \text{ are selected by Algorithm 2 for feedback}\right\}.$$

For every time step $\widetilde{t} \in \mathcal{E}_{y^{(1)},y^{(2)}}(\tau)$, we have:

$$y^{(1)} = y_{\widetilde{t}}^{(1)} = \arg\max_{y\in\mathcal{Y}_{\text{cand}}}\langle\widehat{\theta}_{\widetilde{t}}, \phi(x,y)\rangle \tag{5}$$

$$(y^{(2)}, \widetilde{\theta}_{\widetilde{t}}) \;=\; (y^{(2)}_{\widetilde{t}}, \widetilde{\theta}_{\widetilde{t}}) \;=\; \underset{(y',\theta)\in\mathcal{Y}_{\text{cand}}\times\Theta_{\widetilde{t}}}{\arg\max} \langle \theta, \phi(x,y') - \phi(x,y^{(1)})\rangle \tag{6}$$

Note that $\widehat{\theta}_{\widetilde{t}}, \widetilde{\theta}_{\widetilde{t}} \in \Theta_{\widetilde{t}}$, and $\theta^\star \in \Theta_{\widetilde{t}}$ with probability at least $1-\delta$.

Since the stopping condition is violated in $\widetilde{t}$, we have:

$$\langle \widetilde{\theta}_{\widetilde{t}}, \phi(x,y^{(2)}) - \phi(x,y^{(1)})\rangle \;\geq\; \epsilon. \tag{7}$$

Also, due to Eq. (6), we have:

$$\begin{aligned}\langle \widetilde{\theta}_{\widetilde{t}}, \phi(x,y^{(2)}) - \phi(x,y^{(1)})\rangle \;&\geq \langle \theta^\star, \phi(x,y^{(2)}) - \phi(x,y^{(1)})\rangle \\ &\geq \min_{y,y'\in\mathcal{Y}_{\text{cand}};y\neq y'} \langle \theta^\star, \phi(x,y) - \phi(x,y')\rangle \\ &= \Delta_{\mathcal{Y}_{\text{cand}}},\end{aligned} \tag{8}$$

where the second inequality is due to the observation that $y^{(2)} \neq y^{(1)}$ (because of stopping condition violation). Then, by combining Eq. (7) and Eq. (8), we get:

$$\langle \widetilde{\theta}_{\widetilde{t}}, \phi(x,y^{(2)}) - \phi(x,y^{(1)})\rangle \;\geq\; \max\left\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\right\}. \tag{9}$$

Further, due to Eq. (5), we have:

$$\langle \widehat{\theta}_{\widetilde{t}}, \phi(x,y^{(1)}) - \phi(x,y^{(2)})\rangle \;\geq\; 0. \tag{10}$$

Then, by combining Eq. (9) and Eq. (10), we get:

$$\begin{aligned}\max\left\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\right\} \;&\leq \langle \widetilde{\theta}_{\widetilde{t}} - \widehat{\theta}_{\widetilde{t}}, \phi(x,y^{(2)}) - \phi(x,y^{(1)})\rangle \\ &\leq \|\widetilde{\theta}_{\widetilde{t}} - \widehat{\theta}_{\widetilde{t}}\|_{H_{\widetilde{t}}(\theta^\star)} \cdot \|\phi(x,y^{(2)}) - \phi(x,y^{(1)})\|_{H^{-1}_{\widetilde{t}}(\theta^\star)} \\ &\leq 2S\sqrt{d\log\left(e + \frac{S\widetilde{t}}{d}\right) + \log\frac{1}{\delta}} \cdot \|\phi(x,y^{(2)}) - \phi(x,y^{(1)})\|_{H^{-1}_{\widetilde{t}}(\theta^\star)},\end{aligned}$$

where the last inequality holds with probability at least $1-\delta$ due to Lemma 1.

Now, let $\widetilde{t}$ be the largest value in $\mathcal{E}_{y^{(1)},y^{(2)}}(\tau)$, i.e., the last time the pair of responses $(y^{(1)}, y^{(2)})$ selected before time step τ. Then, for the time step $\widetilde{t}$, we have:

$$\begin{aligned}\max\left\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\right\} \;&\leq 2S\sqrt{d\log\left(e + \frac{S\widetilde{t}}{d}\right) + \log\frac{1}{\delta}} \cdot \|\phi(x,y^{(2)}) - \phi(x,y^{(1)})\|_{H^{-1}_{\widetilde{t}}(\theta^\star)} \\ &\leq 2S\sqrt{d\log\left(e + \frac{S\widetilde{t}}{d}\right) + \log\frac{1}{\delta}} \cdot \sqrt{\frac{\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\left|\mathcal{E}_{y^{(1)},y^{(2)}}(\widetilde{t}-1)\right|}}\end{aligned}$$

where the first inequality holds with probability at least $1-\delta$ due to Lemma 1 and the second inequality is due to Lemma 2. Hence, we can bound:

$$\begin{aligned}\left|\mathcal{E}_{y^{(1)},y^{(2)}}(\tau)\right| \;&= \left|\mathcal{E}_{y^{(1)},y^{(2)}}(\widetilde{t}-1)\right| + 1 \\ &\leq \frac{4S^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\left\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\right\}^2}\left[d\log\left(e + \frac{S\widetilde{t}}{d}\right) + \log\frac{1}{\delta}\right] + 1 \\ &\leq \frac{4S^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\left\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\right\}^2}\left[d\log\left(e + \frac{S\tau}{d}\right) + \log\frac{1}{\delta}\right] + 1\end{aligned}$$

By summing over all the pairs of responses, we get:

$$\begin{aligned}\tau \;&= \sum_{(y^{(1)},y^{(2)})\in\mathcal{Y}_{\text{cand}}\times\mathcal{Y}_{\text{cand}}} \left|\mathcal{E}_{y^{(1)},y^{(2)}}(\tau)\right| \\ &\leq \frac{4S^2K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\left\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\right\}^2}\left[d\log\left(e + \frac{S\tau}{d}\right) + \log\frac{1}{\delta}\right] + K^2\end{aligned}$$

□

Full Proof of Theorem 2. Let $\alpha = \log\left(e + \frac{S\tau}{d}\right)$. Then, $\tau = \frac{d}{S}(e^\alpha - e)$. Thus, we can write the inequality in Lemma 3 as follows:

$$\frac{d}{S}(e^\alpha - e) \;\le\; \frac{4S^2K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2}\left[d\alpha + \log\frac{1}{\delta}\right] + K^2,$$

which we can write as follows:

$$e^\alpha \;\le\; \frac{4S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2}\cdot\alpha + \frac{4S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2 d}\log\frac{1}{\delta} + \frac{K^2S}{d}$$

By letting $w = e^\alpha$, we have:

$$w \;\le\; A\log w + B$$

where $A = \frac{4S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2}$, and $B = \frac{4S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2 d}\log\frac{1}{\delta} + \frac{K^2S}{d}$.

Below, we consider the case $w > e$. If $w \le e$, we already get the bound. For $w \ge e$, we have $\log w \ge 1$, so

$$w \;\le\; A\log w + B \;\le\; A\log w + B\log w \;=\; C\log w,$$

where $C := A + B$. Hence, we get:

$$\frac{w}{\log w} \;\le\; C.$$

Define $f(t) = \frac{t}{\log t}$ for $t \ge e$. One can easily check that $f'(t) > 0$ for $t > e$. Thus, f is strictly increasing on (e, ∞). Therefore, the inequality $f(w) = \frac{w}{\log w} \le C$ forces

$$w \;\le\; t_0,$$

where $t_0 > e$ is the unique solution of $f(t_0) = C$. Let $t_1 = 3C\log C$. Then, for $C \ge 3$, we have:

$$\log t_1 \;=\; \log 3 + \log C + \log\log D \;\le\; \log C + \log C + \log C \;=\; 3\log C.$$

Hence, we get

$$f(t_1) \;=\; \frac{t_1}{\log t_1} \;\ge\; \frac{3C\log C}{3\log C} \;=\; C \;=\; f(t_0).$$

Finally, due to monotonicity of $f(t)$ for $t > e$, we have:

$$t_0 \;\le\; t_1 \;=\; 3C\log C, \text{ for all } C \ge 3.$$

Thus, we have: $w \le \max\{e, t_0\} \le \max\{e, t_1\}$. Therefore, $\tau = \frac{d}{S}(w - e) \le \frac{d}{S}w \le \frac{d}{S}\max\{e, t_1\} \le \frac{d}{S}t_1$. Then, with probability at least $1 - \delta$, we have:

$$\begin{aligned}
\tau \;&\le \frac{3d}{S}\cdot(A + B)\log(A + B) \\
&= 3\cdot\left[\frac{4S^2K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2}\left(d + \log\frac{1}{\delta}\right) + K^2\right]\log\left(\frac{4S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2 d}\left(d + \log\frac{1}{\delta}\right) + \frac{K^2S}{d}\right) \\
&\le 3\cdot\left[\frac{8S^2K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2}\left(d + \log\frac{1}{\delta}\right)\right]\log\left(\frac{8S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2 d}\left(d + \log\frac{1}{\delta}\right)\right) \\
&= 24\cdot\Omega\cdot K^2\cdot\log\left(\frac{8S}{d}\cdot\Omega\cdot K^2\right) \\
&= \mathcal{O}(\Omega K^2\cdot\log(\Omega K^2)),
\end{aligned}$$

where $\Omega := \frac{S^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2}\left(d + \log\frac{1}{\delta}\right)$.

Since $|\langle\theta, \phi(x,y) - \phi(x,y')\rangle| \le \|\theta\|_2\cdot\|\phi(x,y) - \phi(x,y')\|_2 \le S$, we have $\epsilon \le S$. Further note that $\kappa^\star_{\mathcal{X},\mathcal{Y}} \ge 4$, $K \ge 1$, and $S \ge 1$ (we can scale up S). Thus, we have $C = A + B \ge A = \frac{4S^3K^2\kappa^\star_{\mathcal{X},\mathcal{Y}}}{\max\{\epsilon, \Delta_{\mathcal{Y}_{\text{cand}}}\}^2} \ge 16S \ge 3$. □