ACTIVE QUERY SYNTHESIS FOR LEARNING FROM PREFERENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

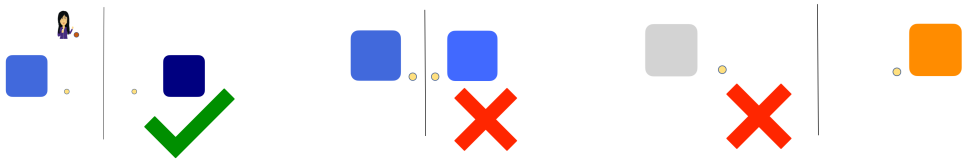
Most response models for pairwise preference comparisons assume all feedback is equally reliable, overlooking that comparisons between near identical or highly dissimilar options are often ambiguous. We introduce a *confidence aware response model*, where response reliability depends on the similarity of the compared options. A user study we conducted shows it models human choices better than standard models. Separately, active learning reduces the cost of estimating preferences, but pool-based methods evaluate the available candidate queries each round. Building on our response model, we propose *Info-Synth*, an active query synthesis framework. It selects informative queries by maximizing mutual information in closed form, up to a one-dimensional search, avoiding pool-based search entirely. For fixed datasets, we introduce two approximation strategies, *Pair M-dist* and *Pair Opt-dist*, to select the closest available pairs of points. We evaluate our framework on synthetic preference learning, a constrained text-summary dataset, and continuous-space controller gain tuning for a simulated robot. In each case, it matches or exceeds the sample efficiency of pool-based active learning while reducing the computational cost.

1 INTRODUCTION

Learning from preferences (Fürnkranz & Hüllermeier, 2010) underlies a wide range of modern systems, even though the problems look quite different at surface-level. An online shopping platform recommends the next product a customer is likely to buy by learning which of two items a customer would prefer (Rendle et al., 2012), a reinforcement learning agent learns a reward function for a robotic task by comparing pairs of behaviors rather than requiring a hand-crafted reward signal (Christiano et al., 2017), and a large language model is aligned to human values by training on which of two candidate responses people prefer (Ouyang et al., 2022). What ties these together is a shared strategy: rather than asking someone to assign an explicit numerical rating to each option, we instead ask them to compare pairs of options and infer their underlying preference from the response (David, 1963; Thurstone, 1927). Comparisons of this kind are often far easier to elicit reliably than absolute judgements (Thurstone, 1927), which is why this framing extends naturally to settings well beyond recommendation systems, such as an engineer's judgement of which system configuration performs better (Sadigh et al., 2017; Csomay-Shanklin et al., 2022). In this paper, we study two such problems: preference learning over items, and hyperparameter tuning through subjective pairwise judgements.

We adopt an active learning approach (Settles, 2009) to select pairwise comparison queries that estimate the user's preference with as few queries as possible. A good query is both *informative* and can be answered with *high confidence* (Bıyık et al., 2019). As illustrated in Fig. 1, comparisons between items that are nearly identical (Fig. 1b) or completely different (Fig. 1c) have unreliable responses. In contrast, ideal queries (Fig. 1a) have an intermediate level of similarity between the items which is likely to result in more confident responses. To explain this observation, we introduce the *confidence aware response model* that accounts for response reliability during query selection. We demonstrate through a *color estimation study* using query responses provided by humans to compare different response models, that our model is indeed successful at selecting *informative* and *higher confidence* queries.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Ideal query (b) Points are too close (c) Points are too far away

Figure 1: Illustrations of pairwise comparison queries with different levels of similarity between them. (a) An ideal query: involves items that are similar enough to compare and different enough to choose between (b) nearly identical items: the answer is a coin flip (c) very different items: have no basis for comparison

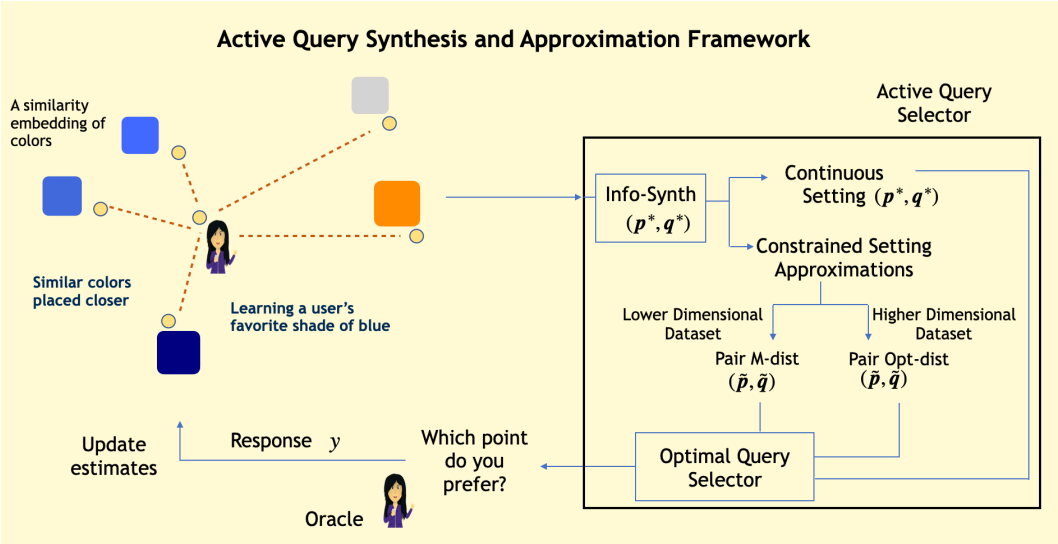


Figure 2: Visualization of the active query synthesis and approximation framework, using a color similarity embedding to estimate a user's preferred shade of blue. *Info-Synth* first generates an optimal continuous query $(\boldsymbol{p}^*, \boldsymbol{q}^*)$. In the continuous setting, this query is used directly. In the constrained setting, it is approximated for a fixed dataset using either *Pair M-dist* or *Pair Opt-dist*, depending on dimensionality and the output is $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})$. The output of the query selector is either $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ or $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})$, depending on the problem setting. The oracle's response (y) to the selected query is then used to update the user preference estimates.

Standard pool-based active learning methods involve exhaustive scoring of the available candidate pairs in each iteration, which can be computationally expensive. Directly synthesizing queries instead is more efficient and even allows us to generate query points that do not exist in the original dataset (Wang et al., 2015; Sadigh et al., 2017). Building on this idea, we develop an active query synthesis method, *Info-Synth*, for the proposed response model that generates informative queries within a continuous space by maximizing the mutual information between the response and the underlying model parameters. As illustrated in Fig. 2, when we can obtain a response directly to the synthesized query $(\boldsymbol{p}^*, \boldsymbol{q}^*)$, we refer to it as the *continuous setting*. When restricted to a fixed dataset though (the *constrained setting* branch), plainly mapping this continuous query to the nearest available points fails to approximate the informativeness of the optimal query pair. To address this, we propose two dataset-specific approximation strategies to find allowable pairs $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})$: *Pair M-dist*, which uses a *Mahalanobis* distance metric for lower-dimensional datasets, and *Pair Opt-dist*, which uses an optimality criteria based heuristic for higher-dimensional ones.

As illustrated by the color-based example in Fig. 2, *Info-Synth* can be applied to preference learning over items. To demonstrate its applicability to the second problem, hyperparameter tuning, we consider the practical challenge of controller gain tuning. In robotic and other cyber-physical systems, *optimal* performance lacks a rigid objective function and instead depends on an engineer's subjective intuition to resolve qualitative trade-offs (e.g., whether a trajectory is "too jittery" or a robot "cuts corners" too aggressively). Since these judgments are subjective and each candidate gain setting is costly to evaluate via simulation (Bergstra & Bengio, 2012), active preference learning offers a structured way to tune gains from human feedback without a predefined reward function (Csomay-Shanklin et al., 2022). Gain tuning is also an ideal application for our continuous query space. Since

a robot's behavioral sensitivity to parameter changes is often nonlinear, sampling directly from a continuous space allows our method to dynamically identify the most informative and perceptible comparisons.

We evaluate our methods in three different settings: (i) preference learning on synthetic data in both continuous and constrained settings (ii) constrained preference learning on a text summary dataset restricted to allowable pairs, and (iii) continuous-space controller gain tuning for a simulated mobile robot.

To summarize, below are the contributions of this work:

1. A novel *confidence aware response model* (Sec. 3) for pairwise comparisons that, when combined with the *Info-Synth* (Sec. 4) query synthesis framework, results in selection of optimal and high-confidence queries, thereby promoting improved performance demonstrated by the *user color estimation study* (Sec. 6.1)
2. Two strategies to find approximations of the optimal continuous queries using existing points in a dataset
   (a) *Pair M-dist*, a Mahalanobis distance-based method for lower dimensional datasets (Sec. 5)
   (b) *Pair Opt-dist*, an optimality criteria based heuristic and a more computationally tractable method for higher dimensional datasets (Sec. 5)
3. Empirical demonstrations of our framework's performance on real-world preference learning and practical robotic gain tuning tasks (Sec. 6)

2 BACKGROUND

The problem of learning from preferences can be setup as estimating an unknown quantity of interest, $\boldsymbol{w} \in \mathcal{A} \subset \mathbb{R}^d$, representing what a user or evaluator wants - a preferred item, or a preferred system configuration - from their responses to pairwise comparison queries ("Which among A and B do you prefer?"). We assume the space of candidates $\mathcal{A}$ is represented in a similarity embedding, where the similar candidates are placed closer to each other and dissimilar ones farther apart. To model preferences over this space, we adopt the *ideal point model* (Coombs, 1950), which assumes there exists the user's ideal point $\boldsymbol{w} \in \mathcal{A}$, that best explains their preferences, with candidates closer to $\boldsymbol{w}$ preferred over those farther away (Jamieson & Nowak, 2011; Xu & Davenport, 2020; Chumbalov et al., 2020; Chen et al., 2025). We seek to find a solution iteratively by asking pairwise comparison queries, and updating our estimate based on the responses obtained. We use a Bayesian framework with MCMC sampling to estimate the user preferences (details in Sec. A).

Since real users rarely respond with perfect certainty, we require a probabilistic model for the comparison responses. Suppose that the response to a query, Y is a binary random variable where $Y = 1$ denotes candidate $\boldsymbol{p}$ is preferred to $\boldsymbol{q}$. A popular model based on the *Bradley-Terry* model (Bradley & Terry, 1952), used extensively in this literature (Canal et al., 2019; Xu & Davenport, 2020; Chen et al., 2025) to account for the noisy responses is

$$P(Y = 1|W = \boldsymbol{w}) = \Phi\left(k(\|\boldsymbol{w} - \boldsymbol{q}\|_2^2 - \|\boldsymbol{w} - \boldsymbol{p}\|_2^2)\right), \tag{1}$$

which can also be written as

$$P(Y|\boldsymbol{w}) = \Phi\left(k(\boldsymbol{a}^\top \boldsymbol{w} - \tau)\right) = \Phi\left(k\|\boldsymbol{a}\|(\hat{\boldsymbol{a}}^\top \boldsymbol{w} - \hat{\tau})\right)$$

where
$\boldsymbol{a} = 2(\boldsymbol{p} - \boldsymbol{q})$ and $\tau = \|\boldsymbol{p}\|^2 - \|\boldsymbol{q}\|^2$ represent the normal vector and threshold of a hyperplane h bisecting items $\boldsymbol{p}$ and $\boldsymbol{q}$ with $h = (\boldsymbol{a}, \tau) : \boldsymbol{a}^\top \boldsymbol{m} - \tau = 0$ where $\boldsymbol{m} \in \mathbb{R}^d$,
$\|\boldsymbol{a}\| = 2\|\boldsymbol{p} - \boldsymbol{q}\|$, $\hat{\boldsymbol{a}} = \boldsymbol{a}/\|\boldsymbol{a}\|$, $\hat{\tau} = \tau/\|\boldsymbol{a}\|$,
k represents the noise parameter, and
$\Phi(x) = 1/(1 + e^{-x})$ is the logistic function.

A common setting for the noise parameter is using a constant value with $k = k_0$, which we refer to as the *plain* response model in this paper. This model favors queries that are far from each other. Other settings include a normalized noise (referred to as the *normalized* model) with $k = k_0\|\boldsymbol{a}\|^{-1}$ where

the response probabilities depend only on the hyperplane and not the query distances, a decaying noise with $k = k_0 \exp(-\|\boldsymbol{a}\|)$ (Canal et al., 2019), and a normalized noise with probit response model (Chumbalov et al., 2020). In practice, as illustrated in Fig. 1 humans typically are most confident in answering pairwise comparisons when the points are located at an ideal distance from each other. When the points are located too close to each other, it might be hard for the user to select one of them and the response can be ambiguous. When the points are too far from each other, they might lose a common ground of comparison and again result in an unreliable response. We compare our proposed response model with the *plain* and *normalized* models in a user study and discuss the results in Sec. 6.1.

2.1 Other related work

We discuss the works most related to ours in detail here and a survey of broader work in Sec. B. The method proposed in Bıyık et al. (2019) uses the mutual information criterion to select queries that balances the uncertainty of the query and the human ability to provide a response to encourage selection of high confidence queries. This is equivalent to the pool based query selection guided by mutual information. Active query synthesis has been explored for SVM-based classification, where the optimal query is synthesized near the decision boundary (Wang et al., 2015). To obtain an actual data point that can be labeled, this synthesized query's nearest neighbor in the unlabeled dataset is then selected. Synthesis has also been applied to pairwise comparison-based (Chumbalov et al., 2020) search to generate optimal query pairs, using the mutual information objective to guide synthesis. To turn a synthesized pair into an actual pair from the dataset that can be labeled, the method then computes the individual nearest neighbors of the two synthesized points. The nearest neighbor approximation works well for a single point but not for a pair of points since it doesn't approximate the pairwise geometry.

3 Modified Response Model

We modify the response model in (1) to include the query distance based reliability of an answer. To achieve this, we model the noise variance as a function of the query distances which is consistent with frameworks explored in heteroskedastic regression (Chaudhuri et al., 2017; Das et al., 2023). Specifically, we assume:

$$d_1 = \|\boldsymbol{w} - \boldsymbol{p}\|_2^2 + e_1 \quad \text{and} \quad d_2 = \|\boldsymbol{w} - \boldsymbol{q}\|_2^2 + e_2$$

where e_1 and e_2 are noises with standard deviations $\sigma_0\|\boldsymbol{w} - \boldsymbol{p}\|_2^2$ and $\sigma_0\|\boldsymbol{w} - \boldsymbol{q}\|_2^2$ respectively.

The resulting probability model is as below:

$$P(Y|\boldsymbol{w}) = \Phi\left(\frac{\|\boldsymbol{w} - \boldsymbol{q}\|_2^2 - \|\boldsymbol{w} - \boldsymbol{p}\|_2^2}{\sigma_0\sqrt{\|\boldsymbol{w} - \boldsymbol{q}\|_2^4 + \|\boldsymbol{w} - \boldsymbol{p}\|_2^4}}\right) = \Phi\left(f(\boldsymbol{w})\right).$$

It can also be written as

$$P(Y|\boldsymbol{w}) = \Phi\left(\frac{\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})}{\sigma_0\sqrt{2\left(\|\boldsymbol{w} - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})\right)^2}}\right) \tag{2}$$

$\boldsymbol{a}$ represents the normal vector of the separating hyperplane as before,
$\boldsymbol{b} = \frac{\boldsymbol{p}+\boldsymbol{q}}{2}$ represents the midpoint of $\boldsymbol{p}$ and $\boldsymbol{q}$, and
$\Phi(x)$ is the CDF of a symmetric noise distribution (i.e., $\Phi(-x) = 1 - \Phi(x)$ and $\Phi'(-x) = \Phi'(x)$).

This model, referred to as the *confidence aware response model* enables us to select queries with higher confidence responses which are typically pairs of points that are at an ideal distance from each other, which is neither too close nor too far.

4 Query Synthesis

We wish to select queries that result in maximal improvement of our estimate of the preference point W. The information gain of a query is evaluated by computing the Mutual Information (MI)

Algorithm 1 Info-Synth

Input: User distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, noise parameter σ_0, S samples drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
$\boldsymbol{b}^* \leftarrow \boldsymbol{\mu}$ (equiprobable outcome condition)
$\boldsymbol{v}_1 \leftarrow$ principal eigenvector of $\boldsymbol{\Sigma}$
$r^* \leftarrow \arg\min_r \mathbb{E}\left[H\left(\Phi(f(W|\boldsymbol{a} = r\boldsymbol{v}_1)\right)\right)]$
$\boldsymbol{a}^* \leftarrow 4r^*\boldsymbol{v}_1$, $\boldsymbol{p}^* \leftarrow \boldsymbol{b}^* + r^*\boldsymbol{v}_1$ and $\boldsymbol{q}^* \leftarrow \boldsymbol{b}^* - r^*\boldsymbol{v}_1$
Output: $(\boldsymbol{p}^*, \boldsymbol{q}^*)$

between the user point and the response to the query. This can be written as:

$$I(W; Y|(\boldsymbol{p}, \boldsymbol{q})) = H(W) - H(W|Y, (\boldsymbol{p}, \boldsymbol{q})).$$

Using the symmetry of mutual information, we can write

$$I(W; Y|(\boldsymbol{p}, \boldsymbol{q})) = H(Y|(\boldsymbol{p}, \boldsymbol{q})) - H(Y|W, (\boldsymbol{p}, \boldsymbol{q})). \tag{3}$$

Since evaluating all candidate queries is computationally expensive, we instead directly optimize for the most informative query $(\boldsymbol{p}^*, \boldsymbol{q}^*)$. Further, since every pairwise comparison query can also be characterized by its corresponding parameters as derived in (2), we compute the optimal parameters $(\boldsymbol{a}^*, \boldsymbol{b}^*)$ that maximize the information gain.

Reparametrizing the MI expression

Re-writing (3) by conditioning on the hyperplane normal vector and query midpoint instead, we get

$$I(W; Y|(\boldsymbol{a}, \boldsymbol{b})) = H(Y|(\boldsymbol{a}, \boldsymbol{b})) - H(Y|W, (\boldsymbol{a}, \boldsymbol{b})). \tag{4}$$

We now optimize the information gain objective in (4) by separately analyzing the two terms. The first term $H(Y|(\boldsymbol{a}, \boldsymbol{b}))$ is the marginal entropy of the binary response Y. Since binary entropy is maximized when the two outcomes are equiprobable, this term is maximized when the outcomes are equiprobable. Under a Gaussian posterior assumption over W, this condition induces a symmetry constraint that places the query midpoint $\boldsymbol{b}$ at the posterior mean, $\boldsymbol{\mu}$. The second term $\mathop{\mathbb{E}}_{W}[H(Y|W, (\boldsymbol{a}, \boldsymbol{b}))]$ measures the expected conditional entropy of Y given W. It is minimized when the conditional probability $P(Y|\boldsymbol{w})] = \Phi(f(\boldsymbol{w}))$ is as certain as possible. This leads to an optimization over the hyperplane normal vector $\boldsymbol{a}$, which results in $\boldsymbol{a}$ aligning with the principal eigenvector of the posterior covariance $\boldsymbol{\Sigma}$. Below, we provide details about the optimal query.

The first term in (4) can be written as

$$H(Y|(\boldsymbol{a}, \boldsymbol{b})) = H\left(\mathop{\mathbb{E}}_{W}\left[P(Y = 1|(\boldsymbol{a}, \boldsymbol{b}), W)\right]\right) = H(\mathbb{E}[\Phi(f(W))]).$$

We know that entropy for a binary random variable is maximum when both the outcomes are equiprobable, i.e. when

$$\mathbb{E}[P(Y = 1)] = \mathbb{E}[\Phi(f(W))] = 0.5.$$

Assuming $W \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the equi-probability criterion results (more details in Sec. C.2) in

$$\boldsymbol{b}^* = \frac{\boldsymbol{p}^* + \boldsymbol{q}^*}{2} = \boldsymbol{\mu}. \tag{5}$$

This results in the optimal pair of points $(\boldsymbol{p}^*, \boldsymbol{q}^*)$ being of the form $\boldsymbol{p}^* = \boldsymbol{\mu} + \boldsymbol{x}^*$ and $\boldsymbol{q}^* = \boldsymbol{\mu} - \boldsymbol{x}^*$. (Derivation in Sec. C.3)

The optimal value of $\boldsymbol{a}$, $\boldsymbol{a}^*$ can be computed as the minimizer of the second term in (4), as below:

$$\boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} \quad H(Y|W, (\boldsymbol{a}, \boldsymbol{b})) = \arg\min_{\boldsymbol{a}} \quad \mathop{\mathbb{E}}_{W}[H(Y|W, (\boldsymbol{a}, \boldsymbol{b}))] = \arg\min_{\boldsymbol{a}} \quad \mathop{\mathbb{E}}_{W}[H(\Phi(f(W)))].$$

This optimization results in $\boldsymbol{a}^* = 4r^*\boldsymbol{v}_1$ where $\boldsymbol{v}_1$ is the principal eigenvector of $\boldsymbol{\Sigma}$ and r^* is computed via the 1-D optimization $\arg\min_r \mathbb{E}\left[H\left(\Phi(f(W|\boldsymbol{a} = r\boldsymbol{v}_1)\right)\right)]$. (Derivation in Sec. C.4 and Sec. C.5)

The resulting synthesized optimal query is $\boldsymbol{p}^* = \boldsymbol{\mu} + r^*\boldsymbol{v}_1$ and $\boldsymbol{q}^* = \boldsymbol{\mu} - r^*\boldsymbol{v}_1$. We refer to this framework as *Info-Synth*, as presented in Alg. 1. The cost of this synthesis is $O(Sd^2)$ since it involves the estimation of the covariance matrix, $\boldsymbol{\Sigma}$ using S Monte Carlo (MC) samples.

5 QUERY APPROXIMATION

If we have the flexibility to query any pair of points, we can query the synthesized pair of points directly. Else, we need to approximate the them with the best points in the dataset. Most works Chumbalov et al. (2020); Wang et al. (2015) on synthesizing active queries select the nearest neighbors in the dataset as the closest approximations. This works well for a single point, but extending it plainly to a pair of points can produce a geometry entirely unrelated to the synthesized optimal pair. We propose two new methods for selecting the best approximate points that can be applied to datasets depending on their dimensionality. Both employ a filter and refine strategy, where the queries are first filtered for informativeness based on approximate criteria and then refined using exact MI evaluations. The two methods are described below and outlined in Alg. 2.

5.1 PAIR M-DIST

The first method is a Mahalanobis distance based framework. Euclidean distance based methods rely on distances in the data space and points that are close in Euclidean distance may not have same informativeness. To address this, we introduce a Mahalanobis distance based selection that uses a Taylor series approximation of the mutual information. Here, the negative Hessian acts as the metric for selecting candidate points that better approximate the informativeness of the optimal query. The detailed derivations can be found in the appendix (Sec. D) and here we provide an overview of the method.

First, we find the nearest pairs to the optimal pair of points by computing the Mahalanobis distances and selecting the top β fraction of pairs that minimize the distance as below:

$$\mathcal{L}_1(\boldsymbol{p}, \boldsymbol{q}) = \frac{1}{2}(\Delta(\boldsymbol{p}, \boldsymbol{q}))^\top(-\boldsymbol{H}((\boldsymbol{p}^*, \boldsymbol{q}^*)))(\Delta(\boldsymbol{p}, \boldsymbol{q})) = \frac{1}{2}\|\Delta(\boldsymbol{p}, \boldsymbol{q})\|^2_{\boldsymbol{M}} \tag{6}$$

where $\Delta(\boldsymbol{p}, \boldsymbol{q}) = (\boldsymbol{p}^*, \boldsymbol{q}^*) - (\boldsymbol{p}, \boldsymbol{q})$, $\boldsymbol{H}((\boldsymbol{p}^*, \boldsymbol{q}^*)) = \nabla^2_{(\boldsymbol{p},\boldsymbol{q})} I((\boldsymbol{p}^*, \boldsymbol{q}^*))$ is the Hessian matrix and $\boldsymbol{M} = -\boldsymbol{H}((\boldsymbol{p}^*, \boldsymbol{q}^*))$.

Then, we find the most informative pair among these via MI evaluations for these top pairs. The computational cost of this method is $O((S + P)d^2 + \beta PSd)$ where P is the total number of pairs and d is the dimension of the dataset. $O(Sd^2)$ represents the cost of optimal query synthesis and computing the Hessian using Monte Carlo estimation. $O(Pd^2)$ represents the cost of computing the Mahalanobis distances while $O(\beta PSd)$ is the cost of MI evaluation for βP pairs at $O(Sd)$ per pair.

5.2 PAIR OPT-DIST

Since computing the hessian is very expensive in higher dimensions, we use a more computationally tractable heuristic for the filtering in higher-dimensional datasets. This heuristic (more details in Sec. E) is enabled by the *optimality criterion* computed by the *query synthesis framework* in Sec. 4. It measures deviation from the optimality criterion of $\boldsymbol{b}^* = \boldsymbol{\mu}$ and $\boldsymbol{a}^* = 4r^*\boldsymbol{v}_1$ and is given by

$$\mathcal{L}_2(\boldsymbol{p}, \boldsymbol{q}, \eta) = \frac{1}{d}\sum_{i=1}^{d} \frac{\left(\boldsymbol{v}_i^\top(\boldsymbol{b} - \boldsymbol{\mu})\right)^2}{\lambda_i} + \eta\frac{\|(\boldsymbol{p} - \boldsymbol{q}) - 2r^*\boldsymbol{v}_1\|_2^2}{4{r^*}^2}. \tag{7}$$

We refer to this method as *Pair Opt-Dist*, where the top γ fraction of the queries that minimize $\mathcal{L}_2(\boldsymbol{p}, \boldsymbol{q}, \eta)$ are first selected and then refined via MI evaluation. Similarly, the total cost of this method is $O((S + P)d^2 + \gamma PSd)$ with $O(Pd^2)$ representing the cost of computing the first term. Although Pair Opt-dist and Pair M-dist share the same big-O complexity, Pair M-dist is typically more expensive in practice, since it requires computing the Hessian.

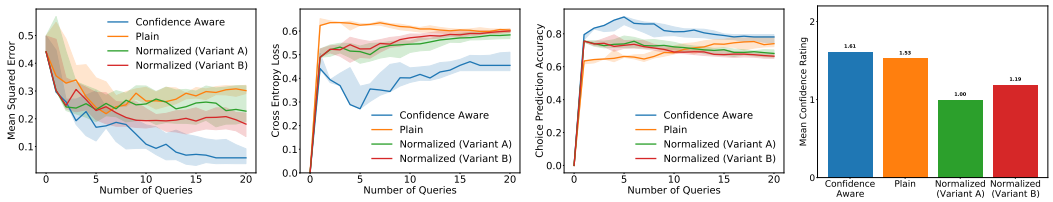
6 EXPERIMENTS

6.1 HUMAN COLOR ESTIMATION STUDY

We conducted an IRB exempt user study with 20 participants. Each participant was shown a target color and answered 20 sequential queries of the form 'Which among these two colors is closer to the

Algorithm 2 ApproxQuery

Input: Dataset $D \subset \mathbb{R}^d$, synthesized query $(\boldsymbol{p}^*, \boldsymbol{q}^*)$, posterior $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, method (M-dist or Opt-dist), filter fraction ρ, scaling constant η (Opt-dist only)
$z^* \leftarrow (\boldsymbol{p}^*, \boldsymbol{q}^*)$
$\mathcal{P} \leftarrow \{(\boldsymbol{p}, \boldsymbol{q}) : \boldsymbol{p}, \boldsymbol{q} \in D,\ \boldsymbol{p} \neq \boldsymbol{q}\}$ (candidate pairs)
if method is M-dist **then**
  $\boldsymbol{M} \leftarrow -\boldsymbol{H}(\boldsymbol{z}^*)$ (negative Hessian, estimated via Monte Carlo)
  for $(\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{P}$ **do**
    $\boldsymbol{z} \leftarrow (\boldsymbol{p}, \boldsymbol{q})$
    $\mathcal{L}_1(\boldsymbol{p}, \boldsymbol{q}) \leftarrow (\boldsymbol{z}^* - \boldsymbol{z})^\top \boldsymbol{M} (\boldsymbol{z}^* - \boldsymbol{z})$ (Mahalanobis distance)
  end for
else ▷ method is Opt-dist
  for $(\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{P}$ **do**
    $\boldsymbol{b} \leftarrow (\boldsymbol{p} + \boldsymbol{q})/2$
    $\mathcal{L}_2(\boldsymbol{p}, \boldsymbol{q}, \eta) \leftarrow \frac{1}{d} \sum_{i=1}^{d} \frac{\left(\boldsymbol{v}_i^\top (\boldsymbol{b} - \boldsymbol{\mu})\right)^2}{\lambda_i} + \eta \frac{\|(\boldsymbol{p}-\boldsymbol{q}) - 2r^* \boldsymbol{v}_1\|_2^2}{4r^{*2}}$
  end for
end if
$\mathcal{C} \leftarrow$ top ρ fraction of $\mathcal{P}$ with minimum scores (filter step)
$(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}}) \leftarrow \arg\max_{(\boldsymbol{p}', \boldsymbol{q}') \in \mathcal{C}} I(W; Y \mid (\boldsymbol{p}', \boldsymbol{q}'))$ (refine via MI evaluation)
Output: $(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})$



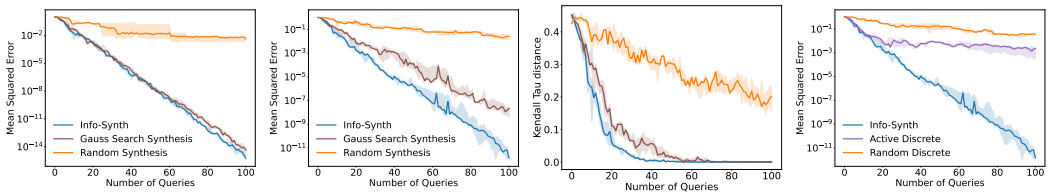
(a) (b) (c) (d)

Figure 3: Color estimation performance across different query response models with the metrics - (a) MSE between the ground truth and the estimated color, measuring the estimation performance, (b) Cross Entropy loss of the predicted probabilities & (c) Choice Predictive accuracy - both measuring the goodness of fit of the models, (d) Mean confidence ratings associated with the query responses. For (a) and (b), lower values and for (c) and (d), higher values indicate better performance respectively.

target?'. We compare the performance of three response models - *confidence aware* model, *plain* and *normalized* models. For the *normalized* model, we considered two synthesis strategies - Variant A: optimizes based on the response model directly and Variant B: employs the method in Chumbalov et al. (2020). The estimation error, cross-entropy and accuracy of each model's predictions on the participant's choices, and mean self-reported confidence, averaged over all the participants, are illustrated in Fig. 3. Our *confidence aware* model is the best on all four, and its queries receive the highest confidence, thereby proving it selects queries that balance both informativeness and response confidence. Even though the *plain* response model also selects queries with high confidence, it doesn't result in a good performance. More details of the study are in Sec. F.1.

6.2 Preference Learning on Synthetic Datasets

We run experiments on synthetic datasets (details in Sec. F.2) for both continuous and constrained settings. We compare the performance of our methods to multiple baselines. *Gauss Search - Synthesis* and *Discrete* - refer to the active query synthesis method in Chumbalov et al. (2020) and its discrete approximation respectively. *Active Discrete* refers to the pool based method of evaluating candidate queries for optimality based on mutual information. *Random - Synthesis* and *Discrete* - refer to the methods where a pair of points are sampled at random over the continuous space and selected at random from the candidate pairs respectively. (Method names inspired by Sadigh et al. (2017).) The nearest point approximation method is referred to as *NN Approx* and the k-NN based method as *k-NN Approx* (details in Sec. F.2)).



(a) $\sigma_0 = 0.001$ (b) $\sigma_0 = 0.1$ (c) $\sigma_0 = 0.1$ (d) $\sigma_0 = 0.1$

Figure 4: Query synthesis performance comparison between different AL methods and Random on synthetic datasets. The plots correspond to datasets in 4D with 1000 points comparing different synthesis methods ((a), (b), (c)) and 500 (d) points comparing synthesis with discrete methods. In the Kendall Tau distance plots, the y-axis corresponds to the KT distance computed between the estimated *user-item* distances against the ground truth *user-item* distances. MSE and KT distance measure the estimation and ranking performance respectively. For both metrics, lower values indicate a better performance.

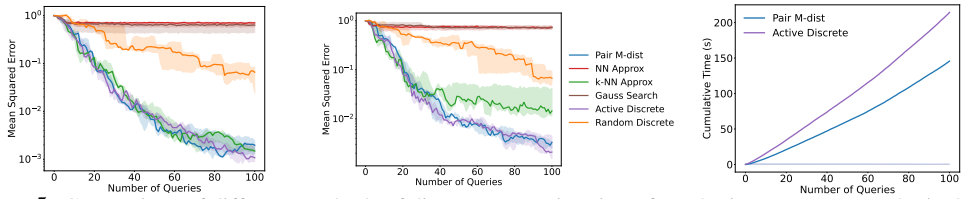


Figure 5: Comparison of different methods of discrete approximation of synthetic query on a synthetic dataset in 10 D with 500 points (left), 100 points (center) and time taken (right) for 100 points.

Active Discrete is the most computationally expensive method with a cost of $O(PSd)$ for P queries while *k-NN Approx* has a cost of $O(Nd+\alpha PSd)$ with cost $O(Nd)$ to compute the nearest neighbors and $O(\alpha PSd)$ for MI evaluation of α fraction of the total pairs.

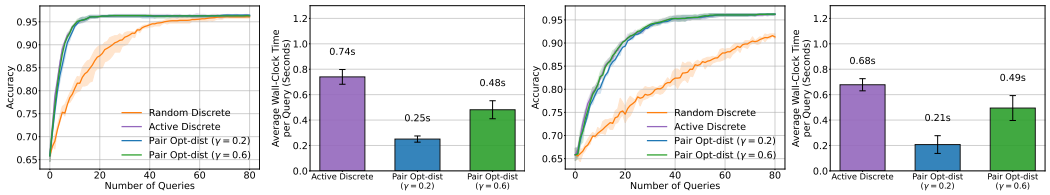
As illustrated in Fig. 4, *Info-Synth* outperforms the baseline *Gauss Search Synthesis* and *Random Synthesis*. While Gauss Search Synthesis has a very similar performance at a lower noise level of $\sigma_0 = 0.001$, we can clearly see the difference when $\sigma_0 = 0.1$. Although *Gauss Search Synthesis* is not a perfect baseline since it is not designed for our specific probability model, we use it as a baseline due to lack of relevant baselines. Further, comparison of *Info-Synth* with the discrete methods *Active Discrete* and *Random Discrete* shows the benefits of optimizing in a continuous space. We have additional results in Sec. F.2.3 for more values of σ_0.

For the discrete approximations, we observe in Fig. 5 that even though *k-NN Approx* performs as well as *Pair M-dist* and *Active Discrete* when there are 500 points, it saturates beyond a certain point for 100 points. Our method achieves the same performance as *Active Discrete*, while being faster as illustrated in the plot on the right. The fractions α and β are selected such that the costs of *k-NN Approx* and *Pair M-dist* are approximately equal. We also discuss ablation studies in Sec. F.2.3 and results for user-estimator model mismatch in Sec. F.2.3.

6.3 Preference Learning on the Reddit Summary Dataset

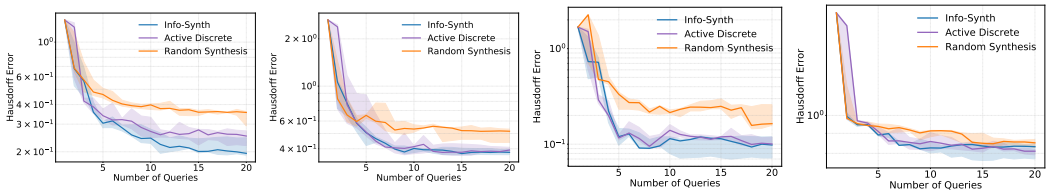
We also conducted experiments on the Reddit Summary TL;DR dataset where we learn user preferences for text summaries through active querying (Stiennon et al., 2020; Li et al., 2024). Users are provided sets consisting of a paragraph and its two candidate summaries for which they provide a pairwise preference label. To obtain summary embeddings, we adopt the same methodology as Chen et al. (2025) where the original text and its summary are concatenated and then passed to the pretrained OPT-350M transformer model that is appended with a pretrained frozen embedding space projection head MLP (Zhang et al., 2022). The output embedding of this MLP head, corresponding to the last token of this concatenated sentence, serves as the summary representation embedding. This process is repeated for the second summary candidate, resulting in two distinct embeddings corresponding to the same paragraph. These embeddings are projected to a space of 128 dimensions using PCA.

For the experiments, we actively query responses to the summary pairs and estimate the user as a point in the same embedding space. We selected a user at random from this dataset and compared the performance of different AL methods for query selection. We use the *Pair Opt-dist* approximation methodology for query selection. Figure 6 shows how our proposed approximation scheme



(a) $\sigma = 0.1$ (b) $\sigma = 0.1$ (c) $\sigma = 0.3$ (d) $\sigma = 0.3$

Figure 6: Performance analysis on the Reddit Summary TL;DR dataset for different AL methods. Our proposed approximation method, *Pair Opt-dist* is shown for two different filtering levels of $\gamma = 0.6$ (green) and $\gamma = 0.2$ (blue). Here γ represents total fraction of queries used for selection. (a) and (b) show the accuracy and average query response time for $\sigma = 0.1$ while (c) and (d) show these results for $\sigma = 0.3$. As random querying doesn't incur any execution costs, the results of Random Discrete are omitted in average query response plots.



(a) (b) (c) (d)

Figure 7: Trajectory tracking error comparison for different experiments. The plots represent the performance with error aggregation over different initial states for high curvature (a) and standard sinusoidal (b) trajectories, and error aggregation over different trajectories with an initial heading error (c) and lateral error (d).

can achieve a performance comparable to *Active Discrete* with a much shorter query approximation time. We can see that reducing γ diminishes the average query computation time, but at the cost of sample efficiency, requiring a greater number of queries to achieve performance parity with *Active Discrete*. We provide further experiment details, additional ablation studies, and evaluation across more users in Sec.F.3. Across all users, our method consistently matches performance with significantly reduced query response time.

6.4 Gain Tuning

In this section, we apply our active query methodology to learn the parameters of a trajectory-tracking controller for a unicyle robot with three parameters k_x, k_y, k_θ. The experimental setup is discussed in Sec. F.4 and here, we discuss the results. We evaluate the performance and sample-efficiency of our proposed method by plotting the tracking error convergence over time (number of queries). We compare our method *Info-Synth* to a pool based active method, *Active Discrete* that samples points in the parameter space and evaluates different candidates for optimality to choose the best one. The baseline *Random Synthesis* queries a randomly sampled gain in the continuous parameter space. We consider performance over different trajectories and initial states.

As illustrated in Fig. 7, we observe that our method achieves the best performance in all the experiments, thus illustrating the contribution of both active learning and query synthesis in a continuous space. *Active Discrete* performs consistently better than *Random Synthesis*, even achieving a similar performance as our method in some cases. *Info-Synth* wins even when its performance is similar to *Active Discrete* as the latter involves MI evaluations of multiple candidate queries.

7 Conclusion

In this work, we addressed two distinct challenges in preference learning: the unreliability of ambiguous pairwise comparisons, and the computational cost of standard pool-based active learning. We introduced a *confidence aware response model*, where response reliability depends on the similarity of the compared options, and validated it against standard models through a user color estimation study. We developed *Info-Synth* method for the response model to select queries, and two approximation strategies, *Pair M-dist* and *Pair Opt-dist* for fixed datasets. Across synthetic preference learning, text summarization, and robotic gain tuning, our framework offers an efficient and structured alternative to pool-based methods.

AI USE STATEMENT

In this work, we used generative AI tools to help implement code for experiments, including implementing few of the methods, setting up the robot controller, formatting code and plots, and building the user study GUI. The authors directed the design decisions and reviewed all outputs. Generative AI tools were similarly used in collaboration with the authors to brainstorm ideas when working through proof strategies for the mathematical claims and to co-draft proof arguments, with the authors leading the mathematical reasoning throughout. We have not used generative AI tools for developing research ideas, generating synthetic datasets, or interpreting results and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to help polish parts of the writing and to assist in finding relevant references. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the authors, and AI-suggested proof strategies were independently checked and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The code for all the experiments in the paper has been provided.

REFERENCES

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of machine learning research*, 13(2), 2012.

Felix Berkenkamp, Angela P Schoellig, and Andreas Krause. Safe controller optimization for quadrotors with gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 491–496. IEEE, 2016.

Erdem Biyik and Dorsa Sadigh. Batch active preference-based learning of reward functions. In *Conference on robot learning*, pp. 519–528. PMLR, 2018.

Erdem Bıyık, Malayandi Palan, Nicholas C Landolfi, Dylan P Losey, and Dorsa Sadigh. Asking easy questions: A user-friendly approach to active reward learning. *arXiv preprint arXiv:1910.04365*, 2019.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Eric Brochu, Tyson Brochu, and Nando De Freitas. A bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 103–112, 2010.

Gregory Canal, Andy Massimino, Mark Davenport, and Christopher Rozell. Active embedding search via noisy paired comparisons. In *International Conference on Machine Learning*, pp. 902–911. PMLR, 2019.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76:1–32, 2017.

Kamalika Chaudhuri, Prateek Jain, and Nagarajan Natarajan. Active heteroscedastic regression. In *International Conference on Machine Learning*, pp. 694–702. PMLR, 2017.

Daiwei Chen, Yi Chen, Aniket Rege, Zhi Wang, and Ramya Korlakai Vinayak. Pal: Sample-efficient personalized reward modeling for pluralistic alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pp. 137–144, 2005.

Daniyar Chumbalov, Lucas Maystre, and Matthias Grossglauser. Scalable and efficient comparison-based search without features. In *International Conference on Machine Learning*, pp. 1995–2005. PMLR, 2020.

Clyde H Coombs. Psychological scaling without a unit of measurement. *Psychological review*, 57 (3):145, 1950.

Joao PL Coutinho, Ivan Castillo, and Marco S Reis. Human-in-the-loop controller tuning using preferential bayesian optimization. *IFAC-PapersOnLine*, 58(14):13–18, 2024.

Noel Csomay-Shanklin, Maegan Tucker, Min Dai, Jenna Reher, and Aaron D Ames. Learning controller gains on bipedal walking robots via user preferences. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 10405–10411. IEEE, 2022.

Aniket Das, Dheeraj M Nagaraj, Praneeth Netrapalli, and Dheeraj Baby. Near optimal heteroscedastic regression with symbiotic learning. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 3696–3757. PMLR, 2023.

Herbert Aron David. *The method of paired comparisons*, volume 12. London, 1963.

Rida T Farouki. The bernstein polynomial basis: A centennial retrospective. *Computer Aided Geometric Design*, 29(6):379–419, 2012.

Johannes Fürnkranz and Eyke Hüllermeier. Preference learning: An introduction. In *Preference learning*, pp. 1–17. Springer, 2010.

Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential bayesian optimization. In *International Conference on Machine Learning*, pp. 1282–1291. PMLR, 2017.

Kevin G Jamieson and Robert D Nowak. Active ranking using pairwise comparisons. *arXiv preprint arXiv:1109.3701*, 2011.

Kejun Li, Maegan Tucker, Erdem Bıyık, Ellen Novoseller, Joel W Burdick, Yanan Sui, Dorsa Sadigh, Yisong Yue, and Aaron D Ames. Roial: Region of interest active learning for characterizing exoskeleton gait preference landscapes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3212–3218. IEEE, 2021.

Xinyu Li, Ruiyang Zhou, Zachary C Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133*, 2024.

Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1557–1563. IEEE, 2017.

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

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In *Proceedings of Robotics: Science and Systems (RSS)*, 2017.

Burr Settles. Active learning literature survey. 2009.

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

L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927. doi: 10.1037/h0070288.

Maegan Tucker, Ellen Novoseller, Claudia Kann, Yanan Sui, Yisong Yue, Joel W Burdick, and Aaron D Ames. Preference-based learning for exoskeleton gait optimization. In *2020 IEEE international conference on robotics and automation (ICRA)*, pp. 2351–2357. IEEE, 2020.

Liantao Wang, Xuelei Hu, Bo Yuan, and Jianfeng Lu. Active learning via query synthesis and nearest neighbour search. *Neurocomputing*, 147:426–434, 2015.

Austin Xu and Mark Davenport. Simultaneous preference and metric learning from paired comparisons. *Advances in Neural Information Processing Systems*, 33:454–465, 2020.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

A PROBLEM SETUP

A query consists of presenting a pair of items $(\boldsymbol{p}, \boldsymbol{q}) \in \mathcal{D}$ to the user $\boldsymbol{w}$ and asking which among the two they prefer. We alternate between obtaining a response $Y \in \{0, 1\}$ to each query and updating our estimate of the user point. The full method is presented in Alg. 3.

Suppose that the user is denoted by the random variable $W \in \mathbb{R}^d$ with a prior density of $p_0(W)$. The posterior distribution after obtaining responses to n queries is given by

$$p(W) \propto p_0(W) \prod_{j=1}^{n} p(Y_j|W)$$

where $\{Y_j\}_{j=1}^n$ denote the set of all responses obtained for n queries involving pairs of items - $\{(\boldsymbol{p}_1, \boldsymbol{q}_1), (\boldsymbol{p}_2, \boldsymbol{q}_2), \ldots, (\boldsymbol{p}_n, \boldsymbol{q}_n)\}$ respectively. The user point is estimated as the mean of the posterior distribution

$$\hat{\boldsymbol{w}} = \mathbb{E}[W|Y^n].$$

In the experiments, we use MCMC sampling to generate samples from the posterior distribution. Even though the query synthesis framework assumes a Gaussian approximation of the posterior, the probabilities are computed using the MCMC samples. The Gaussian approximation is computed by taking the mean and covariance of the MCMC samples.

Algorithm 3 Active Query Synthesis

Input: Prior $p_0(W)$, oracle $\mathcal{O}$, number of queries T, noise parameter σ_0, setting (continuous or constrained), dataset D (if constrained), method (M-dist or Opt-dist), filter fraction ρ
$p(W) \leftarrow p_0(W)$
for $t = 1$ **to** T **do**
  $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t \leftarrow$ posterior mean and covariance of $p(W)$ (estimated via MCMC)
  $(\boldsymbol{p}^*, \boldsymbol{q}^*) \leftarrow$ INFO-SYNTH$(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \sigma_0)$
  if setting is continuous **then**
    $(\boldsymbol{p}_t, \boldsymbol{q}_t) \leftarrow (\boldsymbol{p}^*, \boldsymbol{q}^*)$
  else
    $(\boldsymbol{p}_t, \boldsymbol{q}_t) \leftarrow$ APPROXQUERY$(D, (\boldsymbol{p}^*, \boldsymbol{q}^*), \boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t, \text{mode}, \rho)$
  end if
  $y_t \leftarrow$ oracle response to query $(\boldsymbol{p}_t, \boldsymbol{q}_t)$
  $p(W) \leftarrow p(W) \cdot P(y_t \mid W, \boldsymbol{p}_t, \boldsymbol{q}_t)$ (posterior update)
end for
$\hat{w} \leftarrow \mathbb{E}[W \mid y_1, \ldots, y_T]$
Output: $\hat{w}$

B Additional Related Work

Feedback modalities: Related efforts have also explored adding additional feedback modalities beyond binary comparisons, such as coactive feedback, which allows users to iteratively improve proposed solutions (Tucker et al., 2020) and ordinal labels where users assign items to ordered preference categories (e.g., poor to excellent), yielding richer information about relative quality than binary comparisons alone (Li et al., 2021). These additional feedback modalities have been shown to accelerate preference learning under structured noise assumptions but impose a higher cognitive load for the human providing the feedback.

Hyperparameter Tuning: The application of preference based adaptive learning for tuning parameters is commonly used when it is difficult to obtain a quantitative objective function value for the desired performance. Many works have demonstrated that Bayesian optimization can efficiently learn system parameters using only human preference feedback over presented alternatives (Brochu et al., 2010; González et al., 2017). In robotics, similar ideas have been applied to learn reward functions (Sadigh et al., 2017; Biyik & Sadigh, 2018) based on the responses obtained to pairwise comparisons of trajectories using an active query synthesis approach. Bayesian Optimization has also been applied for controller tuning by adapting the framework to handle human preference feedback (Coutinho et al., 2024; Csomay-Shanklin et al., 2022), enforce safety constraints (Berkenkamp et al., 2016), or optimize efficiency through multi-fidelity data sources (Marco et al., 2017).

C Probability Model and Query Synthesis

C.1 Derivation of the Anisotropic Denominator

We define the points $\boldsymbol{p}$ and $\boldsymbol{q}$ in terms of their midpoint $\boldsymbol{b}$ and the hyperplane normal vector $\boldsymbol{a}$. Following the definitions in Sec. 3, we have $\boldsymbol{a} = 2(\boldsymbol{p} - \boldsymbol{q})$ and $\boldsymbol{b} = \frac{\boldsymbol{p}+\boldsymbol{q}}{2}$. This implies

$$\boldsymbol{p} = \boldsymbol{b} + \frac{\boldsymbol{a}}{4}, \quad \boldsymbol{q} = \boldsymbol{b} - \frac{\boldsymbol{a}}{4}$$

Let the squared distances from the user point $\boldsymbol{w}$ to items $\boldsymbol{p}$ and $\boldsymbol{q}$ be expanded around the midpoint $\boldsymbol{b}$

$$\|\boldsymbol{w} - \boldsymbol{p}\|_2^2 = \left\|(\boldsymbol{w} - \boldsymbol{b}) - \frac{\boldsymbol{a}}{4}\right\|_2^2 = \|\boldsymbol{w} - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16} - \frac{\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})}{2}$$
$$\|\boldsymbol{w} - \boldsymbol{q}\|_2^2 = \left\|(\boldsymbol{w} - \boldsymbol{b}) + \frac{\boldsymbol{a}}{4}\right\|_2^2 = \|\boldsymbol{w} - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16} + \frac{\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})}{2}$$

To simplify the notation, let

$$U = \|\boldsymbol{w} - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}, \quad V = \frac{\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})}{2}$$

Substituting these into the denominator expression $D(\boldsymbol{w}) = \sigma_0\sqrt{\|\boldsymbol{w} - \boldsymbol{p}\|_2^4 + \|\boldsymbol{w} - \boldsymbol{q}\|_2^4}$

$$D(\boldsymbol{w}) = \sigma_0\sqrt{(U - V)^2 + (U + V)^2}$$
$$D(\boldsymbol{w}) = \sigma_0\sqrt{(U^2 + V^2 - 2UV) + (U^2 + V^2 + 2UV)}$$
$$D(\boldsymbol{w}) = \sigma_0\sqrt{2U^2 + 2V^2}$$

Substituting the full expressions for U and V back into the result yields the final non-isotropic form

$$D(\boldsymbol{w}) = \sigma_0\sqrt{2\left(\|\boldsymbol{w} - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})\right)^2}$$

C.2 Approximating distribution of $f(W)$ as a Gaussian

Let $W \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $f(W) = \frac{N(W)}{D(W)}$ where $N(W) = \boldsymbol{a}^\top W - \tau$ and
$D(W) = \sigma_0\sqrt{2\left(\|W - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top(\boldsymbol{w} - \boldsymbol{b})\right)^2}$.

We use moment matching for the approximation:

$$f(W) \sim \mathcal{N}(\mu_f, \sigma_f^2) \text{ where } \mu_f = \mathbb{E}[f(W)] \text{ and } \sigma_f^2 = \mathrm{var}(f(W)).$$

The expected probability can be expressed as

$$\mathbb{E}[\Phi(f(W))] \approx \Phi\left(\frac{\mu_f}{g(\sigma_f)}\right)$$

where $\Phi(x)$ is the noise CDF with the symmetry $\Phi(x) + \Phi(-x) = 1$ and $g(\cdot)$ is an appropriate function of σ_f depending on the specific Φ.

From the equiprobable outcome condition, we have $\mathbb{E}[\Phi(f(W))] = 0.5$. A key characteristic of symmetric link functions such as Φ is that

$$\Phi(0) = 0.5 \quad \text{and} \quad \Phi(x) + \Phi(-x) = 1.$$

From this, it follows that for any symmetric link function

$$\mathbb{E}[\Phi(f(W))] = 0.5 \iff \mu_f = 0.$$

For $\mathbb{E}[f(W)]$ to be 0, the numerator $N(W)$ needs to be an odd function about $\boldsymbol{\mu}$ and the denominator $D(W)$ an even function about $\boldsymbol{\mu}$. We examine both the terms by substituting $W = \boldsymbol{\mu} + \Delta$.

We have,

$$N(\boldsymbol{\mu} + \Delta) = \boldsymbol{a}^\top(\boldsymbol{\mu} + \Delta) - \tau = (\boldsymbol{a}^\top\boldsymbol{\mu} - \tau) + \boldsymbol{a}^\top\Delta.$$

For $N(W)$ to be odd about $\boldsymbol{\mu}$, the constant term should be 0. We have

$$\boldsymbol{a}^\top\boldsymbol{\mu} - \tau = 0$$
$$\text{i.e. } 2(\boldsymbol{p} - \boldsymbol{q})^\top\boldsymbol{\mu} - \left(\|\boldsymbol{p}\|^2 - \|\boldsymbol{q}\|^2\right) = 0$$
$$\implies \boldsymbol{\mu} = \frac{\boldsymbol{p} + \boldsymbol{q}}{2} = \boldsymbol{b}.$$

When this holds, $N(\boldsymbol{\mu} + \Delta) = \boldsymbol{a}^\top\Delta$ which satisfies

$$N(\boldsymbol{\mu} - \Delta) = -N(\boldsymbol{\mu} + \Delta).$$

Further, we have

$$D(W) = \sigma_0\sqrt{2\left(\|W - \boldsymbol{\mu}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top(W - \boldsymbol{\mu})\right)^2}.$$

Under the condition, we substitute $W = \boldsymbol{\mu} + \Delta$ to obtain

$$D(\boldsymbol{\mu} + \Delta) = \sigma_0\sqrt{2\left(\|\Delta\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top\Delta\right)^2},$$

thus making the denominator an even function of Δ where

$$D(\boldsymbol{\mu} - \Delta) = D(\boldsymbol{\mu} + \Delta).$$

For $\boldsymbol{p} = \boldsymbol{\mu} + \boldsymbol{x}$ and $\boldsymbol{q} = \boldsymbol{\mu} - \boldsymbol{x}$, the approximation of the expectation becomes exact due to symmetry as derived in the main paper.

C.3 Expected Probability

When $\boldsymbol{b} = \boldsymbol{\mu}$, the expected probability is exactly 0.5 as explained below.

1. Anti-symmetry
   Let $W = \boldsymbol{\mu} + \Delta$ where $\Delta \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$. Substituting this into $f(W)$ yields

   $$f(\boldsymbol{\mu} + \Delta) = \frac{\boldsymbol{a}^\top\Delta}{\sigma_0\sqrt{2\left(\|\Delta\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top\Delta\right)^2}}.$$

   Since the numerator is linear (odd) and the denominator is quadratic (even) in Δ, $f(W)$ is odd about $\boldsymbol{\mu}$, i.e.

   $$f(\boldsymbol{\mu} - \Delta) = -f(\boldsymbol{\mu} + \Delta).$$

2. Distributional Symmetry
   Since the PDF $p(\Delta)$ is symmetric and f is odd, the random variable $Y = f(W)$ is symmetrically distributed around zero. This implies $p(y) = p(-y)$ and $\mathbb{E}[f(W)] = 0$.
3. Expected Probability
   Any link function satisfying $\Phi(-x) = 1 - \Phi(x)$ results in

$$\mathbb{E}[\Phi(f)] = \int_0^\infty [\Phi(f) + \Phi(-f)]p(f)\, df = \int_0^\infty (1)p(f)\, df = 0.5.$$

C.4 Finding the optimal direction

The optimal value of $\boldsymbol{a}$, $\boldsymbol{a}^*$ can be computed as the minimizer of the second term in equation (4) illustrated as below:

$$\boldsymbol{a}^* = \arg\min_{\boldsymbol{a}} \quad H(Y|W, (\boldsymbol{a}, \boldsymbol{b})) = \arg\min_{\boldsymbol{a}} \quad \mathop{\mathbb{E}}_{W}[H(Y|W, (\boldsymbol{a}, \boldsymbol{b}))] = \arg\min_{\boldsymbol{a}} \quad \mathop{\mathbb{E}}_{W}[H(\Phi(f(W)))].$$

The above entropy is minimized when $\Phi(f(W))$ is close to either 0 or 1. This corresponds to maximizing $|f(W)|$ and since $W \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, we maximize $\mathbb{E}[|f(W)|]$. We have

$$\tilde{\boldsymbol{a}} = \arg\max_{\boldsymbol{a}} \ \mathbb{E}[|f(W)|] = \arg\max_{\boldsymbol{a}} \ \mathbb{E}\left[\frac{|\boldsymbol{a}^\top(W - \boldsymbol{b})|}{\sigma_0\sqrt{2\left(\|W - \boldsymbol{b}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top(W - \boldsymbol{b})\right)^2}}\right].$$

From (5), we have $\boldsymbol{b}^* = \boldsymbol{\mu}$ which results in

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}} \ \mathbb{E}_{W\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\frac{|\boldsymbol{a}^\top(W - \boldsymbol{\mu})|}{\sigma_0\sqrt{2\left(\|W - \boldsymbol{\mu}\|_2^2 + \frac{\|\boldsymbol{a}\|_2^2}{16}\right)^2 + \frac{1}{2}\left(\boldsymbol{a}^\top(W - \boldsymbol{\mu})\right)^2}}\right].$$

We have $\boldsymbol{p}^* = \boldsymbol{\mu} + \boldsymbol{x}^*$ and $\boldsymbol{q}^* = \boldsymbol{\mu} - \boldsymbol{x}^*$ where $\boldsymbol{a}^* = 4\boldsymbol{x}^*$. Further, let $S = W - \boldsymbol{\mu} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$. Rewriting the optimization objective in terms of $\boldsymbol{x}$ and S, we obtain

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} \ \mathbb{E}_{S\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}\left[\frac{4|\boldsymbol{x}^\top S|}{\sigma_0\sqrt{2\left(\|S\|_2^2 + \|\boldsymbol{x}\|_2^2\right)^2 + 2\left(2\boldsymbol{x}^\top S\right)^2}}\right] = \arg\max_{\boldsymbol{x}} F(\boldsymbol{x}).$$

Setting $\boldsymbol{x} = r\boldsymbol{u}$, we obtain

$$r^*, \boldsymbol{u}^* = \arg\max_{r,\boldsymbol{v}} \ \mathbb{E}\left[\frac{4r|\boldsymbol{u}^\top S|}{\sigma_0\sqrt{2\left(\|S\|_2^2 + r^2\right)^2 + 8r^2\left(\boldsymbol{u}^\top S\right)^2}}\right]. \tag{8}$$

The ratio is maximized for $\boldsymbol{u} = \boldsymbol{v}_1$ where $\boldsymbol{v}_1$ is the principal eigenvector of $\boldsymbol{\Sigma}$. Optimizing (8) for r^* results in it being independent of σ_0. However, in practice we find that the optimal magnitude does depend on σ_0. Hence, the optimal magnitude is obtained by one-dimensional optimization as below:

$$r^* = \arg\min_{r} \mathop{\mathbb{E}}_{S}\left[H\left(\Phi\left(\frac{4r\ \boldsymbol{v}_1^\top S}{\sigma_0\sqrt{2\left(\|S\|_2^2 + r^2\right)^2 + 8r^2\left(\boldsymbol{v}_1^\top S\right)^2}}\right)\right)\right]$$

with scope for varying with σ_0, since σ_0 enters nonlinearly inside the link function - entropy composition. The resulting normal vector of the optimal hyperplane is $\boldsymbol{a}^* = 4r\boldsymbol{v}_1$. Finally, the synthesized optimal query is $\boldsymbol{p}^* = \boldsymbol{\mu} + r^*\boldsymbol{v}_1$ and $\boldsymbol{q}^* = \boldsymbol{\mu} - r^*\boldsymbol{v}_1$.

Intuitively, in (8) choosing a larger value of r can make both the numerator and denominator larger. Thus, we want to select a moderate value of r. This moderate value of the magnitude is what encourages the selection of pairs of points that are at an ideal distance, neither too close nor too far from each other, thus resulting in selection of higher response-confidence queries.

C.5 Optimizing expected entropy

Proposition 1. *Let $S \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ with $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$ positive definite, and let $\boldsymbol{v}_1$ denote the principal eigenvector of $\mathbf{\Sigma}$ with eigenvalue λ_1. The maximizer of*

$$F(\boldsymbol{x}) = \mathbb{E}_{S\sim\mathcal{N}(\mathbf{0},\mathbf{\Sigma})}\left[\frac{4|\boldsymbol{x}^\top S|}{\sigma_0\sqrt{2\left(\|S\|_2^2+\|\boldsymbol{x}\|_2^2\right)^2+8\left(\boldsymbol{x}^\top S\right)^2}}\right] \tag{9}$$

has the form $\boldsymbol{x}^* = r^*\boldsymbol{v}_1$, where

$$r^* = \underset{r}{arg\,min}\ \underset{S}{\mathbb{E}}\left[H\left(\Phi\left(\frac{4r\ \boldsymbol{v}_1^\top S}{\sigma_0\sqrt{2\left(\|S\|_2^2+r^2\right)^2+8r^2\left(\boldsymbol{v}_1^\top S\right)^2}}\right)\right)\right].$$

Proof. **Step 1: Optimal direction.** We write $\boldsymbol{x} = r\boldsymbol{u}$ with $r > 0$ and $\|\boldsymbol{u}\|_2 = 1$. For a fixed realization of S, let $a := \|S\|_2^2$ and $t := (\boldsymbol{u}^\top S)^2$. Let the integrand be

$$f(\boldsymbol{u}, S) = \frac{4r\sqrt{t}}{\sigma_0\sqrt{2(a+r^2)^2+8r^2t}}.$$

Treating a and r as constants, differentiate f^2 with respect to t:

$$\frac{d}{dt}\left[\frac{16t}{2(a+r^2)^2+8r^2t}\right] = \frac{32(a+r^2)^2}{\left[2(a+r^2)^2+8r^2t\right]^2} > 0.$$

Hence $f(\boldsymbol{u}, S)$ is strictly increasing in $|\boldsymbol{u}^\top S|$ for every fixed S and $r > 0$. Since expectation preserves pointwise monotone ordering,

$$\underset{\|\boldsymbol{u}\|_2=1}{\arg\max}\,\mathbb{E}[f(\boldsymbol{u}, S)] = \underset{\|\boldsymbol{u}\|_2=1}{\arg\max}\,\mathbb{E}\left[\varphi\left(|\boldsymbol{u}^\top S|\right)\right]$$

for some increasing function φ. For $S \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$,

$$\boldsymbol{u}^\top S \sim \mathcal{N}(\mathbf{0},\ \boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}), \quad \text{so} \quad |\boldsymbol{u}^\top S| \overset{d}{=} \sqrt{\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}}|C|, \quad C \sim \mathcal{N}(0,1).$$

Therefore,

$$\mathbb{E}\left[\varphi\left(|\boldsymbol{u}^\top S|\right)\right] = \mathbb{E}\left[\varphi\left(\sqrt{\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}}\,|C|\right)\right],$$

which is increasing in $\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}$. The term $\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}$ is maximized over the unit sphere at $\boldsymbol{u} = \boldsymbol{v}_1$, so

$$\boldsymbol{u}^\star = \boldsymbol{v}_1.$$

Since $T_u = \boldsymbol{u}^\top S$ is a one-dimensional Gaussian with mean $\mathbf{0}$ and variance $\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}$, we can rewrite it using the reparameterization trick as $T_u = \sqrt{\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}}\,C$, where $C \sim \mathcal{N}(0,1)$. Therefore, the expectation becomes: $\mathbb{E}[\varphi(|T_u|)] = \mathbb{E}[\varphi(\sqrt{\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}}\,|C|)]$. Because φ is a monotonically increasing function, this expectation strictly increases as the scale factor $\sqrt{\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}}$ (and thus the variance $\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}$) increases. Hence, the expectation is maximized by maximizing the variance $\boldsymbol{u}^\top\mathbf{\Sigma}\boldsymbol{u}$. This is a classic Rayleigh quotient problem, which is maximized when $\boldsymbol{u}$ is the principal eigenvector $\boldsymbol{v}_1$ of $\mathbf{\Sigma}$.

Step 2: Optimal magnitude via 1D optimization. With $\boldsymbol{u} = \boldsymbol{v}_1$ fixed, define $Z \sim \mathcal{N}(0, \mathbf{I}_d)$ via the spectral decomposition $S = U\Lambda^{1/2}Z$, so that

$$\boldsymbol{v}_1^\top S = \sqrt{\lambda_1}\,Z_1, \qquad \|S\|_2^2 =: Q = \sum_{i=1}^{d}\lambda_i Z_i^2.$$

The reduced 1D objective is

$$\phi(r) = \frac{4r\sqrt{\lambda_1}}{\sigma_0}\mathbb{E}\left[\frac{|Z_1|}{\sqrt{2(Q+r^2)^2+8r^2\lambda_1 Z_1^2}}\right].$$

Computing the optimal magnitude via 1D numerical optimization of $\phi(r)$ results in $\tilde{r}$ being independent of σ_0. In practice, we observe that the optimal magnitude varies with σ_0 and hence we compute it as

$$r^* = \arg\min_r \mathop{\mathbb{E}}_S \left[H\left(\Phi(f(r\boldsymbol{v}_1, S))\right)\right] \tag{10}$$

initialized at $r_0 = \sqrt{\mathrm{tr}(\Sigma)}$

Conclusion. Combining both steps, the maximizer of $F(\boldsymbol{x})$ is

$$\boxed{\boldsymbol{x}^* = r^* \boldsymbol{v}_1,}$$

where r^* is computed via 1D optimization.

□

In order to get an intuition for the optimal magnitude, let's try to compute its approximate value. We have $\mathbb{E}[|\boldsymbol{x}^\top S|] = \sqrt{\frac{2\lambda_1}{\pi}} r$, and $\mathbb{E}[\|S\|^2 + \|\boldsymbol{x}\|^2] = \mathrm{Tr}(\Sigma) + r^2$, and thus

$$|\tilde{\boldsymbol{x}}| \approx \arg\max_{|\boldsymbol{x}|} \frac{\mathbb{E}[4|\boldsymbol{x}^\top S|]}{\mathbb{E}\left[\sigma_0 \sqrt{2\left(\|S\|^2 + r^2\right)^2 + 8\left(\boldsymbol{x}^\top S\right)^2}\right]} \approx \arg\max_r \frac{4\sqrt{\frac{2\lambda_1}{\pi}} r}{\sqrt{2\left(\mathrm{Tr}(\Sigma) + r^2\right)^2 + \frac{16\lambda_1}{\pi} r^2}} = m(r)$$

where λ_1 is the principal eigenvalue of $\boldsymbol{\Sigma}$.

To find the optimal value of r, we differentiate m^2 with respect to r and equate it to 0 to obtain

$$\tilde{r} = \|\tilde{\boldsymbol{x}}\| = \sqrt{\mathrm{Tr}(\Sigma)}.$$

The magnitude $\|\boldsymbol{x}\|$ now depends on the $\mathrm{Tr}(\Sigma)$ which represents the total variance across all dimensions of the random vector W. As the posterior shrinks, so does $\|\boldsymbol{x}\|$ and hence, the selected points move closer to $\boldsymbol{\mu}$. In practice, we observe that the optimal magnitude varies with σ_0 and thus, use 1D optimization to compute its value. The principal eigenvector $\boldsymbol{v}_1$ is still the optimal direction.

D Calculating Mahalanobis distances

D.1 Metric Derivation

Let $I(\boldsymbol{z}) := I(\boldsymbol{p}, \boldsymbol{q}) := I(Y; W|\boldsymbol{p}, \boldsymbol{q})$ with $\boldsymbol{z}^*$ being the maximizer.

The Taylor series expansion of $I(\boldsymbol{z})$ around $\boldsymbol{z}^*$ is:

$$I(\boldsymbol{z}) \approx I(\boldsymbol{z}^*) + \nabla I(\boldsymbol{z}^*)^\top (\boldsymbol{z} - \boldsymbol{z}^*) + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}^*)^\top \nabla^2_{\boldsymbol{z}} I(\boldsymbol{z}^*)(\boldsymbol{z} - \boldsymbol{z}^*).$$

Since $\boldsymbol{z}^*$ is the maximizer, the gradient $\nabla I(\boldsymbol{z}^*) = 0$. Thus, the approximation simplifies to:

$$I(\boldsymbol{z}) \approx I(\boldsymbol{z}^*) + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}^*)^\top \boldsymbol{H}(\boldsymbol{z}^*)(\boldsymbol{z} - \boldsymbol{z}^*) \tag{11}$$

where $\boldsymbol{H}(\boldsymbol{z}^*) = \nabla^2_{\boldsymbol{z}} I(\boldsymbol{z}^*)$ is the Hessian matrix.

Consider the difference:

$$I(\boldsymbol{z}^*) - I(\boldsymbol{z}) = \frac{1}{2}(\boldsymbol{z}^* - \boldsymbol{z})^\top (-\boldsymbol{H}(\boldsymbol{z}^*))(\boldsymbol{z}^* - \boldsymbol{z})$$

which can also be written as

$$\Delta I(\boldsymbol{z}) = \frac{1}{2}(\Delta \boldsymbol{z})^\top (-\boldsymbol{H}(\boldsymbol{z}^*))(\Delta \boldsymbol{z}) = \frac{1}{2}\|\Delta \boldsymbol{z}\|^2_{\boldsymbol{M}}.$$

where $\Delta I(\boldsymbol{z}) = I(\boldsymbol{z}^*) - I(\boldsymbol{z})$, $\Delta \boldsymbol{z} = \boldsymbol{z}^* - \boldsymbol{z}$ and $\boldsymbol{M} = -\boldsymbol{H}(\boldsymbol{z}^*)$.

Deriving the Hessian

Consider

$$I(W;Y|(\boldsymbol{p},\boldsymbol{q})) = H(Y|(\boldsymbol{p},\boldsymbol{q})) - H(Y|W,(\boldsymbol{p},\boldsymbol{q})).$$

For $\pi = \mathbb{E}_W[\mathbb{P}(Y=1|W,(\boldsymbol{p},\boldsymbol{q}))]$, we have

$$\begin{aligned} H(Y|(\boldsymbol{p},\boldsymbol{q})) &= H\left(\mathbb{E}_W[Y|W,(\boldsymbol{p},\boldsymbol{q})]\right) \\ &= H(\pi) \\ &= -\pi\log\pi - (1-\pi)\log(1-\pi). \end{aligned}$$

Further, for $p = \mathbb{P}(Y=1|W,(\boldsymbol{p},\boldsymbol{q}))$, we have

$$\begin{aligned} -H(Y|W,(\boldsymbol{p},\boldsymbol{q})) &= -\mathbb{E}_W[H(Y|W,(\boldsymbol{p},\boldsymbol{q}))] \\ &= -\mathbb{E}_W[-p\log p - (1-p)\log(1-p)] \\ &= \mathbb{E}_W[p\log p + (1-p)\log(1-p)]. \end{aligned}$$

The objective function can now be written as

$$I(\boldsymbol{p},\boldsymbol{q}) = H(\pi) + \mathbb{E}_W\left[g(f(\boldsymbol{w}))\right] \tag{12}$$

where

1. $$f(\boldsymbol{w}) = \frac{\|\boldsymbol{w}-\boldsymbol{q}\|_2^2 - \|\boldsymbol{w}-\boldsymbol{p}\|_2^2}{\sigma_0\sqrt{\|\boldsymbol{w}-\boldsymbol{q}\|_2^4 + \|\boldsymbol{w}-\boldsymbol{p}\|_2^4}}$$
2. The link function and entropy-related term g, where $\Phi(x)$ is noise distribution CDF
$$g(f) = \Phi(f)\log(\Phi(f)) + \Phi(-f)\log(\Phi(-f))$$

D.2 Gradient Derivation

The gradient with respect to $\boldsymbol{p}$ is obtained via the chain rule

$$\begin{aligned} \nabla_{\boldsymbol{p}} I(\boldsymbol{p},\boldsymbol{q}) &= \frac{dH}{d\pi}\nabla_{\boldsymbol{p}}\pi + \nabla_{\boldsymbol{p}}\left(\mathbb{E}_W\left[g(f(\boldsymbol{w}))\right]\right) \\ &= \frac{dH}{d\pi}\nabla_{\boldsymbol{p}}\pi + \mathbb{E}_W\left[\frac{dg}{df}\nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right] \end{aligned}$$

Derivation of $\frac{dH}{d\pi}$

$$\frac{dH}{d\pi} = \log\left(\frac{1-\pi}{\pi}\right)$$

Derivation of $\nabla_{\boldsymbol{p}}\pi$

$$\nabla_{\boldsymbol{p}}\pi = \mathbb{E}_W\left[\Phi'(f)\nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right]$$

Derivation of $\frac{dg}{df}$

We use the property $\Phi'(-f) = \Phi'(f)$.

$$\begin{aligned} \frac{dg}{df} &= \left[\Phi'(f)\log(\Phi(f)) + \Phi(f)\frac{\Phi'(f)}{\Phi(f)}\right] - \left[\Phi'(-f)\log(\Phi(-f)) + \Phi(-f)\frac{\Phi'(-f)}{\Phi(-f)}\right] \\ &= \Phi'(f)\log(\Phi(f)) + \Phi'(f) - \Phi'(-f)\log(\Phi(-f)) - \Phi'(-f) \\ &= \Phi'(f)\left[\log(\Phi(f)) - \log(\Phi(-f))\right] \\ &= \Phi'(f)\log\left(\frac{\Phi(f)}{\Phi(-f)}\right). \end{aligned}$$

Thus, we have

$$\frac{dg}{df} = \Phi'(f)\log\left(\frac{\Phi(f)}{\Phi(-f)}\right). \tag{13}$$

DERIVATION OF $\nabla_{\boldsymbol{p}} f(\boldsymbol{w})$

Let $A = \|\boldsymbol{w} - \boldsymbol{q}\|^2$, $B = \|\boldsymbol{w} - \boldsymbol{p}\|^2$, $S = \sqrt{A^2 + B^2}$, $N = A - B$ and $D = \sigma_0 S$.

We have

$$\nabla_{\boldsymbol{p}} N = 2(\boldsymbol{w} - \boldsymbol{p})$$
$$\nabla_{\boldsymbol{p}} S = \frac{1}{2S} \nabla_{\boldsymbol{p}} (A^2 + B^2) = \frac{B}{S} \left(-2(\boldsymbol{w} - \boldsymbol{p})\right) \tag{14}$$

Using the quotient rule $\nabla_{\boldsymbol{p}}(N/D) = ((\nabla_{\boldsymbol{p}} N)D - N(\nabla_{\boldsymbol{p}} D))/D^2$, we get

$$\begin{aligned}
\nabla_{\boldsymbol{p}} f &= \frac{2(\boldsymbol{w} - \boldsymbol{p})D - N(B/S)(-2\sigma_0(\boldsymbol{w} - \boldsymbol{p}))}{D^2} \\
&= \frac{2(\boldsymbol{w} - \boldsymbol{p})}{SD^2} \left[DS + \sigma_0 NB\right] \\
&= \frac{2(\boldsymbol{w} - \boldsymbol{p})}{SD^2} \left[\sigma_0 S^2 + \sigma_0 NB\right] \\
&= \frac{2(\boldsymbol{w} - \boldsymbol{p})}{\sigma_0 S^3} \left[A^2 + B^2 + (A - B)B\right] \\
&= \frac{2(\boldsymbol{w} - \boldsymbol{p})}{\sigma_0 S^3} \left[A(A + B)\right].
\end{aligned}$$

Thus, we have

$$\nabla_{\boldsymbol{p}} f = \frac{2(\boldsymbol{w} - \boldsymbol{p})}{\sigma_0 S^3} \left[A(A + B)\right]. \tag{15}$$

By symmetry, the gradient with respect to $\boldsymbol{q}$ is

$$\nabla_{\boldsymbol{q}} f = \frac{-2(\boldsymbol{w} - \boldsymbol{q})}{\sigma_0 S^3} \left[B(A + B)\right]. \tag{16}$$

FINAL EXPRESSION

Combining the results, we obtain

$$\nabla_{\boldsymbol{p}} I(\boldsymbol{p}, \boldsymbol{q}) = \log\left(\frac{1 - \pi}{\pi}\right) \mathbb{E}_W \left[\Phi'(f) \nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right] + \mathbb{E}_W \left[\Phi'(f) \log\left(\frac{\Phi(f)}{\Phi(-f)}\right) \nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right]$$

$$\nabla_{\boldsymbol{p}} I(\boldsymbol{p}, \boldsymbol{q}) = \mathbb{E}_W \left[\left(\log\left(\frac{1 - \pi}{\pi}\right) + \log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\right) \Phi'(f) \nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right] \tag{17}$$

$$\nabla_{\boldsymbol{q}} I(\boldsymbol{p}, \boldsymbol{q}) = \mathbb{E}_W \left[\left(\log\left(\frac{1 - \pi}{\pi}\right) + \log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\right) \Phi'(f) \nabla_{\boldsymbol{q}} f(\boldsymbol{w})\right]. \tag{18}$$

D.3 HESSIAN DERIVATION

The full Hessian of the mutual information $I(\boldsymbol{z})$ with respect to the parameter vector $\boldsymbol{z} = (\boldsymbol{p}, \boldsymbol{q})$ is a $2D \times 2D$ block matrix

$$\boldsymbol{H}(\boldsymbol{z}) = \nabla_{\boldsymbol{z}}^2 I = \begin{pmatrix} \nabla_{\boldsymbol{p}}^2 I & \nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I) \\ \nabla_{\boldsymbol{p}}(\nabla_{\boldsymbol{q}} I) & \nabla_{\boldsymbol{q}}^2 I \end{pmatrix}.$$

DIAGONAL BLOCKS

We first derive the diagonal blocks - $\nabla_{\boldsymbol{p}}^2 I$ and $\nabla_{\boldsymbol{q}}^2 I$.

We have

$$\begin{aligned}
\nabla_{\boldsymbol{p}} I(\boldsymbol{p}, \boldsymbol{q}) &= \log\left(\frac{1 - \pi}{\pi}\right) \mathbb{E}_W \left[\Phi'(f) \nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right] + \mathbb{E}_W \left[\Phi'(f) \log\left(\frac{\Phi(f)}{\Phi(-f)}\right) \nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right] \\
&= \log\left(\frac{1 - \pi}{\pi}\right) \nabla_{\boldsymbol{p}} \pi + \nabla_{\boldsymbol{p}} \mathbb{E}_W \left[g(f(\boldsymbol{w}))\right] \\
&= G_{\boldsymbol{p}1} + G_{\boldsymbol{p}2}.
\end{aligned}$$

The Hessian with respect to $\boldsymbol{p}$ is found by differentiating the gradient.

Using the product rule for vector calculus, we obtain

$$\nabla_{\boldsymbol{p}} G_{\boldsymbol{p}1} = (\nabla_{\boldsymbol{p}}\pi)\left(\nabla_{\boldsymbol{p}} \log\left(\frac{1-\pi}{\pi}\right)\right)^{\top} + \log\left(\frac{1-\pi}{\pi}\right)\nabla^2_{\boldsymbol{p}}\pi$$

where

$$\nabla_{\boldsymbol{p}} \log\left(\frac{1-\pi}{\pi}\right) = -\frac{1}{\pi(1-\pi)}\nabla_{\boldsymbol{p}}\pi$$

and

$$\nabla^2_{\boldsymbol{p}}\pi = \mathbb{E}_W\left[\Phi''(f)\nabla_{\boldsymbol{p}} f\left(\nabla_{\boldsymbol{p}} f\right)^{\top} + \Phi'(f)\nabla^2_{\boldsymbol{p}} f\right].$$

Thus, we have

$$\nabla_{\boldsymbol{p}} G_{\boldsymbol{p}1} = -\frac{1}{\pi(1-\pi)}(\nabla_{\boldsymbol{p}}\pi)(\nabla_{\boldsymbol{p}}\pi)^{\top} + \log\left(\frac{1-\pi}{\pi}\right)\mathbb{E}_W\left[\Phi''(f)\nabla_{\boldsymbol{p}} f\left(\nabla_{\boldsymbol{p}} f\right)^{\top} + \Phi'(f)\nabla^2_{\boldsymbol{p}} f\right] \tag{19}$$

and

$$\begin{aligned}
\nabla_{\boldsymbol{p}} G_{\boldsymbol{p}2} &= \nabla^2_{\boldsymbol{p}}\,\mathbb{E}_W\left[g(f(\boldsymbol{w}))\right] \\
&= \nabla_{\boldsymbol{p}}\,\mathbb{E}_W\left[\frac{dg}{df}\nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right] \\
&= \mathbb{E}_W\left[\nabla_{\boldsymbol{p}}\left(\frac{dg}{df}\right)(\nabla_{\boldsymbol{p}} f)^T + \frac{dg}{df}\nabla^2_{\boldsymbol{p}} f\right] \\
&= \mathbb{E}_W\left[\frac{d^2 g}{df^2}(\nabla_{\boldsymbol{p}} f)(\nabla_{\boldsymbol{p}} f)^T + \frac{dg}{df}\nabla^2_{\boldsymbol{p}} f\right]
\end{aligned}$$

DERIVATION OF $\frac{d^2 g}{df^2}$

We differentiate $\frac{dg}{df} = \log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\cdot\Phi'(f)$ using the product rule as below.

$$\begin{aligned}
\frac{d^2 g}{df^2} &= \frac{d}{df}\left(\log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\cdot\Phi'(f)\right) \\
&= \frac{d}{df}\left(\Phi'(f)\cdot[\log(\Phi(f)) - \log(\Phi(-f))]\right) \\
&= \Phi'(f)\left[\frac{\Phi'(f)}{\Phi(f)} - \frac{-\Phi'(f)}{\Phi(-f)}\right] + \log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\cdot\Phi''(f) \\
&= \left(\Phi'(f)\right)^2\left[\frac{\Phi(-f)+\Phi(f)}{\Phi(f)\Phi(-f)}\right] + \log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\cdot\Phi''(f).
\end{aligned}$$

Thus, we have

$$\frac{d^2 g}{df^2} = \frac{\left(\Phi'(f)\right)^2}{\Phi(f)\Phi(-f)} + \log\left(\frac{\Phi(f)}{\Phi(-f)}\right)\cdot\Phi''(f). \tag{20}$$

DERIVATION OF $\nabla^2_{\boldsymbol{p}} f(\boldsymbol{w})$

Let $C_{\boldsymbol{p}} = \frac{2A(A+B)}{\sigma_0 S^3}$, so that $\nabla_{\boldsymbol{p}} f = C_{\boldsymbol{p}}(\boldsymbol{w}-\boldsymbol{p})$.

We use the product rule $\nabla^2_{\boldsymbol{p}} f = (\boldsymbol{w}-\boldsymbol{p})(\nabla_{\boldsymbol{p}} C_{\boldsymbol{p}})^T + C_{\boldsymbol{p}}\nabla_{\boldsymbol{p}}(\boldsymbol{w}-\boldsymbol{p})$.

First, $\nabla_{\boldsymbol{p}}(\boldsymbol{w}-\boldsymbol{p}) = -\mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

Next, we find the gradient of the scalar $C_{\boldsymbol{p}}$ with respect to $\boldsymbol{p}$. We have

$$\nabla_{\boldsymbol{p}} C_{\boldsymbol{p}} = \nabla_{\boldsymbol{p}} \left(\frac{2A(A+B)}{\sigma_0 S^3} \right) = \nabla_{\boldsymbol{p}} \left(\frac{U}{V} \right).$$

Further

$$\nabla_{\boldsymbol{p}} U = 2A \nabla_{\boldsymbol{p}} B = -4A(\boldsymbol{w}-\boldsymbol{p})$$

and

$$\nabla_{\boldsymbol{p}} V = 3\sigma_0 S^2 \nabla_{\boldsymbol{p}} S.$$

Substituting $\nabla_{\boldsymbol{p}} S$ from (14), we get

$$\nabla_{\boldsymbol{p}} V = 3\sigma_0 S^2 \frac{B}{S}(-2(\boldsymbol{w}-\boldsymbol{p})) = -6\sigma_0 S B(\boldsymbol{w}-\boldsymbol{p}).$$

Now, we have

$$\begin{aligned}
\nabla_{\boldsymbol{p}} C_{\boldsymbol{p}} &= \frac{\sigma_0 S^3(-4A(\boldsymbol{w}-\boldsymbol{p})) - 2A(A+B)(-6\sigma_0 S B(\boldsymbol{w}-\boldsymbol{p}))}{(\sigma_0 S^3)^2} \\
&= \frac{4\sigma_0 S A(\boldsymbol{w}-\boldsymbol{p})\left[-S^2 + 3B(A+B)\right]}{\sigma_0^2 S^6} \\
&= \frac{4A(\boldsymbol{w}-\boldsymbol{p})\left[3B(A+B) - S^2\right]}{\sigma_0 S^5}
\end{aligned}$$

Combining the terms gives the Hessian of f

$$\nabla^2_{\boldsymbol{p}} f = (\boldsymbol{w}-\boldsymbol{p}) \left(\frac{4A(\boldsymbol{w}-\boldsymbol{p})\left[3B(A+B) - S^2\right]}{\sigma_0 S^5} \right)^\top - \frac{2A(A+B)}{\sigma_0 S^3}\mathbf{I}$$

$$= \frac{2A}{\sigma_0 S^3}\left[\frac{2(3B(A+B) - S^2)}{S^2}(\boldsymbol{w}-\boldsymbol{p})(\boldsymbol{w}-\boldsymbol{p})^\top - (A+B)\mathbf{I} \right]$$

$$= \frac{2A}{\sigma_0 S^3}\left[\frac{2T_{\boldsymbol{p}}}{S^2}(\boldsymbol{w}-\boldsymbol{p})(\boldsymbol{w}-\boldsymbol{p})^\top - (A+B)\mathbf{I} \right]$$

where $T_{\boldsymbol{p}} = (3B(A+B) - S^2)$.

Finally, we get

$$\nabla_{\boldsymbol{p}} G_{\boldsymbol{p}2} = \mathbb{E}_W \left[\left(\frac{(\Phi'(f))^2}{\Phi(f)\Phi(-f)} + \log\left(\frac{\Phi(f)}{\Phi(-f)} \right) \cdot \Phi''(f) \right) (\nabla_{\boldsymbol{p}} f)(\nabla_{\boldsymbol{p}} f)^T + \left(\Phi'(f) \log\left(\frac{\Phi(f)}{\Phi(-f)} \right) \right) \nabla^2_{\boldsymbol{p}} f \right].$$

DERIVATION OF $\nabla_{\boldsymbol{q}} G_{\boldsymbol{q}2}$

Similarly, we have

$$\nabla_{\boldsymbol{q}} G_{\boldsymbol{q}2} = \mathbb{E}_W \left[\frac{d^2 g}{df^2}(\nabla_{\boldsymbol{q}} f)(\nabla_{\boldsymbol{q}} f)^T + \frac{dg}{df}\nabla^2_{\boldsymbol{q}} f \right] \tag{21}$$

with

$$\begin{aligned}
\nabla^2_{\boldsymbol{q}} f &= (-(\boldsymbol{w}-\boldsymbol{q}))\left(\nabla_{\boldsymbol{q}} C_{\boldsymbol{q}}\right)^T + C_{\boldsymbol{q}} \nabla_{\boldsymbol{q}} \left(-(\boldsymbol{w}-\boldsymbol{q})\right) \\
&= (-(\boldsymbol{w}-\boldsymbol{q}))\left(\nabla_{\boldsymbol{q}} C_{\boldsymbol{q}}\right)^T + C_{\boldsymbol{q}}\mathbf{I}
\end{aligned}$$

where

$$C_{\boldsymbol{q}} = \frac{2B(A+B)}{\sigma_0 S^3}$$

and

$$\nabla_{\boldsymbol{q}} C_{\boldsymbol{q}} = \frac{4B(\boldsymbol{w}-\boldsymbol{q})\left[3A(A+B)-S^2\right]}{\sigma_0 S^5}.$$

Thus, we have

$$\nabla^2_{\boldsymbol{q}} f = (-(\boldsymbol{w}-\boldsymbol{q}))\left(\frac{4B(\boldsymbol{w}-\boldsymbol{q})\left[3A(A+B)-S^2\right]}{\sigma_0 S^5}\right)^\top + \frac{2B(A+B)}{\sigma_0 S^3}\mathbf{I}$$

$$= \frac{2B}{\sigma_0 S^3}\left[(A+B)\mathbf{I} - \frac{2(3A(A+B)-S^2)}{S^2}(\boldsymbol{w}-\boldsymbol{q})(\boldsymbol{w}-\boldsymbol{q})^\top\right]$$

$$= \frac{2B}{\sigma_0 S^3}\left[(A+B)\mathbf{I} - \frac{2T_{\boldsymbol{q}}}{S^2}(\boldsymbol{w}-\boldsymbol{q})(\boldsymbol{w}-\boldsymbol{q})^\top\right]$$

where $T_{\boldsymbol{q}} = (3A(A+B)-S^2)$.

Final Expression

Assembling all components gives the final expression for the Hessian

$$\nabla^2_{\boldsymbol{p}} I = \nabla_{\boldsymbol{p}} G_{\boldsymbol{p}1} + \nabla_{\boldsymbol{p}} G_{\boldsymbol{p}2} \tag{22}$$

$$= -\frac{1}{\pi(1-\pi)}\left(\nabla_{\boldsymbol{p}}\pi\right)\left(\nabla_{\boldsymbol{p}}\pi\right)^\top + \mathbb{E}_W\left[\Psi(\boldsymbol{w})\left(\nabla_{\boldsymbol{p}} f\right)\left(\nabla_{\boldsymbol{p}} f\right)^\top + \Omega(\boldsymbol{w})\nabla^2_{\boldsymbol{p}} f\right] \tag{23}$$

and

$$\nabla^2_{\boldsymbol{q}} I = -\frac{1}{\pi(1-\pi)}\left(\nabla_{\boldsymbol{q}}\pi\right)\left(\nabla_{\boldsymbol{q}}\pi\right)^\top + \mathbb{E}_W\left[\Psi(\boldsymbol{w})\left(\nabla_{\boldsymbol{q}} f\right)\left(\nabla_{\boldsymbol{q}} f\right)^\top + \Omega(\boldsymbol{w})\nabla^2_{\boldsymbol{q}} f\right] \tag{24}$$

where

$$\Psi(\boldsymbol{w}) = \frac{\left(\Phi'(f)\right)^2}{\Phi(f)\Phi(-f)} + \log\left(\frac{(1-\pi)\Phi(f)}{\pi\Phi(-f)}\right)\cdot\Phi''(f),$$

$$\Omega(\boldsymbol{w}) = \log\left(\frac{(1-\pi)\Phi(f)}{\pi\Phi(-f)}\right)\Phi'(f)$$

and $\nabla_{\boldsymbol{p}}\pi$, $\nabla_{\boldsymbol{q}}\pi$, $\nabla_{\boldsymbol{p}} f$, $\nabla_{\boldsymbol{q}} f$, $\nabla^2_{\boldsymbol{p}} f$ and $\nabla^2_{\boldsymbol{q}} f$ are the expressions derived above.

Off-Diagonal Hessian Blocks

We will now derive the off-diagonal block $\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I)$. We have

$$\nabla_{\boldsymbol{q}}\left(\nabla_{\boldsymbol{p}} I\right) = \nabla_{\boldsymbol{q}}\left(G_{\boldsymbol{p}1} + G_{\boldsymbol{p}2}\right).$$

Further

$$\nabla_{\boldsymbol{q}} G_{\boldsymbol{p}1} = \left(\nabla_{\boldsymbol{p}}\pi\right)\left(\nabla_{\boldsymbol{q}}\log\left(\frac{1-\pi}{\pi}\right)\right)^\top + \log\left(\frac{1-\pi}{\pi}\right)\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}}\pi).$$

where

$$\nabla_{\boldsymbol{q}}\log\left(\frac{1-\pi}{\pi}\right) = -\frac{1}{\pi(1-\pi)}\nabla_{\boldsymbol{q}}\pi$$

and

$$\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}}\pi) = \mathbb{E}_W\left[\Phi''(f)\nabla_{\boldsymbol{p}} f\left(\nabla_{\boldsymbol{q}} f\right)^\top + \Phi'(f)\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f)\right]$$

Next,

$$\nabla_{\boldsymbol{q}} G_{\boldsymbol{p}2} = \nabla_{\boldsymbol{q}}\left(\mathbb{E}_W\left[\frac{dg}{df}\nabla_{\boldsymbol{p}} f\right]\right)$$

$$= \mathbb{E}_W\left[\nabla_{\boldsymbol{q}}\left(\frac{dg}{df}\cdot\nabla_{\boldsymbol{p}} f\right)\right]$$

$$= \mathbb{E}_W\left[\left(\nabla_{\boldsymbol{p}} f\right)\left(\nabla_{\boldsymbol{q}}\frac{dg}{df}\right)^T + \left(\frac{dg}{df}\right)\left(\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f)\right)\right]$$

DERIVATION OF $\nabla_{\boldsymbol{q}}\left(\frac{dg}{df}\right)$

We use the chain rule. $\frac{dg}{df}$ is a scalar function of f, which is a function of $\boldsymbol{q}$.

$$\nabla_{\boldsymbol{q}}\left(\frac{dg}{df}\right) = \frac{d}{df}\left(\frac{dg}{df}\right)\cdot\nabla_{\boldsymbol{q}} f = \left(\frac{d^2g}{df^2}\right)\nabla_{\boldsymbol{q}} f \tag{25}$$

We have already derived $\frac{d^2g}{df^2}$ and $\nabla_{\boldsymbol{q}} f$.

DERIVATION OF $\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f)$

This is the most complex term. We start with the expression for $\nabla_{\boldsymbol{p}} f$:

$$\nabla_{\boldsymbol{p}} f = C_{\boldsymbol{p}}(\boldsymbol{w}-\boldsymbol{p}), \quad \text{where} \quad C_{\boldsymbol{p}} = \frac{2A(A+B)}{\sigma_0 S^3}$$

We differentiate $\nabla_{\boldsymbol{p}} f$ with respect to $\boldsymbol{q}$ using the product rule:

$$\begin{aligned}\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f) &= \nabla_{\boldsymbol{q}}\left(C_{\boldsymbol{p}}(\boldsymbol{w}-\boldsymbol{p})\right)\\ &= (\boldsymbol{w}-\boldsymbol{p})(\nabla_{\boldsymbol{q}} C_{\boldsymbol{p}})^T + C_{\boldsymbol{p}}\underbrace{\left(\nabla_{\boldsymbol{q}}(\boldsymbol{w}-\boldsymbol{p})\right)}_{=\mathbf{0}}\\ &= (\boldsymbol{w}-\boldsymbol{p})(\nabla_{\boldsymbol{q}} C_{\boldsymbol{p}})^T\end{aligned}$$

The second term is the zero matrix because $(\boldsymbol{w}-\boldsymbol{p})$ is not a function of $\boldsymbol{q}$. Next, we find the gradient of the scalar $C_{\boldsymbol{p}}$ with respect to $\boldsymbol{q}$. We have

$$\nabla_{\boldsymbol{q}} C_{\boldsymbol{p}} = \nabla_{\boldsymbol{q}}\left(\frac{2A(A+B)}{\sigma_0 S^3}\right) = \nabla_{\boldsymbol{q}}\left(\frac{U}{V}\right).$$

Further

$$\begin{aligned}\nabla_{\boldsymbol{q}} U &= 2\nabla_{\boldsymbol{q}}(A^2+AB)\\ &= 2(2A+B)\nabla_{\boldsymbol{q}} A = -4(2A+B)(\boldsymbol{w}-\boldsymbol{q})\end{aligned}$$

and

$$\nabla_{\boldsymbol{q}} V = 3\sigma_0 S^2\nabla_{\boldsymbol{q}} S.$$

Substituting $\nabla_{\boldsymbol{q}} S$, we get

$$\nabla_{\boldsymbol{q}} V = 3\sigma_0 S^2\frac{A}{S}(-2(\boldsymbol{w}-\boldsymbol{q})) = -6\sigma_0 SA(\boldsymbol{w}-\boldsymbol{q}).$$

Now, we have

$$\begin{aligned}\nabla_{\boldsymbol{q}} C_{\boldsymbol{p}} &= \frac{\sigma_0 S^3(-4(2A+B)(\boldsymbol{w}-\boldsymbol{q})) - 2A(A+B)(-6\sigma_0 SA(\boldsymbol{w}-\boldsymbol{q}))}{\left(\sigma_0 S^3\right)^2}\\ &= \frac{4\sigma_0 S(\boldsymbol{w}-\boldsymbol{q})\left[-S^2(2A+B)+3A^2(A+B)\right]}{\sigma_0^2 S^6}\\ &= \frac{4(\boldsymbol{w}-\boldsymbol{q})\left[3A^2(A+B)-S^2(2A+B)\right]}{\sigma_0 S^5}\end{aligned}$$

Substituting this back, the mixed-partial Hessian of f is

$$\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f) = \left(\frac{4\left[3A^2(A+B)-S^2(2A+B)\right]}{\sigma_0 S^5}\right)(\boldsymbol{w}-\boldsymbol{p})(\boldsymbol{w}-\boldsymbol{q})^\top$$

$$= \frac{4T_{\boldsymbol{pq}}}{\sigma_0 S^5}(\boldsymbol{w}-\boldsymbol{p})(\boldsymbol{w}-\boldsymbol{q})^\top$$

where $T_{\boldsymbol{pq}} = 3A^2(A+B)-S^2(2A+B)$. This is a scalar multiplied by a $D\times D$ outer product matrix.

We plug the expressions back into the main equation for $\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I)$

$$\nabla_{\boldsymbol{q}} G_{\boldsymbol{p}^2} = \mathbb{E}_W\left[\left(\frac{d^2g}{df^2}\right)(\nabla_{\boldsymbol{p}} f)(\nabla_{\boldsymbol{q}} f)^T + \left(\frac{dg}{df}\right)(\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f))\right] \tag{26}$$

Final Expression

We plug the expressions back into the main equation for $\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I)$

$$\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I) = -\frac{1}{\pi(1-\pi)} \left(\nabla_{\boldsymbol{p}}\pi\right) \left(\nabla_{\boldsymbol{q}}\pi\right)^\top + \mathbb{E}_W \left[\Psi(\boldsymbol{w}) \left(\nabla_{\boldsymbol{p}} f\right) \left(\nabla_{\boldsymbol{q}} f\right)^\top + \Omega(\boldsymbol{w}) \nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f)\right] \quad (27)$$

The other off-diagonal block is its transpose: $\nabla_{\boldsymbol{p}}(\nabla_{\boldsymbol{q}} I) = (\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I))^T$, i.e.

$$\nabla_{\boldsymbol{p}}(\nabla_{\boldsymbol{q}} I) = \nabla = -\frac{1}{\pi(1-\pi)} \left(\nabla_{\boldsymbol{q}}\pi\right) \left(\nabla_{\boldsymbol{p}}\pi\right)^\top + \mathbb{E}_W \left[\Psi(\boldsymbol{w}) \left(\nabla_{\boldsymbol{q}} f\right) \left(\nabla_{\boldsymbol{p}} f\right)^\top + \Omega(\boldsymbol{w}) \nabla_{\boldsymbol{p}}(\nabla_{\boldsymbol{q}} f)\right] \quad (28)$$

D.4 Summary

The terms of the Hessian are as below

$$\nabla^2_{\boldsymbol{z}} I = \begin{pmatrix} \nabla^2_{\boldsymbol{p}} I & \nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I) \\ \nabla_{\boldsymbol{p}}(\nabla_{\boldsymbol{q}} I) & \nabla^2_{\boldsymbol{q}} I \end{pmatrix}.$$

Let $A = \|\boldsymbol{w} - \boldsymbol{q}\|^2$, $B = \|\boldsymbol{w} - \boldsymbol{p}\|^2$, and $S = \sqrt{A^2 + B^2}$. We have

$$\nabla^2_{\boldsymbol{p}} I(\boldsymbol{z}^*) = -\frac{1}{\pi(1-\pi)} \left(\nabla_{\boldsymbol{p}}\pi\right) \left(\nabla_{\boldsymbol{p}}\pi\right)^\top + \mathbb{E}_W \left[\Psi(\boldsymbol{w}) \left(\nabla_{\boldsymbol{p}} f\right) \left(\nabla_{\boldsymbol{p}} f\right)^\top + \Omega(\boldsymbol{w}) \nabla^2_{\boldsymbol{p}} f\right]$$

where $f = f(\boldsymbol{w})$ with

$$f(\boldsymbol{w}) = \frac{\|\boldsymbol{w} - \boldsymbol{q}\|_2^2 - \|\boldsymbol{w} - \boldsymbol{p}\|_2^2}{\sigma_0 \sqrt{\|\boldsymbol{w} - \boldsymbol{q}\|_2^4 + \|\boldsymbol{w} - \boldsymbol{p}\|_2^4}}. \quad (29)$$

$$\pi = \mathbb{E}_W \left[\mathbb{P}\left(Y = 1 | W, (\boldsymbol{p}, \boldsymbol{q})\right)\right], \quad \nabla_{\boldsymbol{p}}\pi = \mathbb{E}_W \left[\Phi'(f) \nabla_{\boldsymbol{p}} f(\boldsymbol{w})\right],$$

$$\Psi(\boldsymbol{w}) = \frac{\left(\Phi'(f)\right)^2}{\Phi(f)\Phi(-f)} + \log\left(\frac{(1-\pi)\Phi(f)}{\pi\Phi(-f)}\right) \cdot \Phi''(f),$$

$$\Omega(\boldsymbol{w}) = \log\left(\frac{(1-\pi)\Phi(f)}{\pi\Phi(-f)}\right) \Phi'(f), \quad \nabla_{\boldsymbol{p}} f = \frac{2(\boldsymbol{w} - \boldsymbol{p})}{\sigma_0 S^3} \left[A(A+B)\right],$$

and

$$\nabla^2_{\boldsymbol{p}} f = \frac{2A}{\sigma_0 S^3} \left[\frac{2T_{\boldsymbol{p}}}{S^2} (\boldsymbol{w} - \boldsymbol{p})(\boldsymbol{w} - \boldsymbol{p})^\top - (A+B)\mathbf{I}\right]$$

where $T_{\boldsymbol{p}} = (3B(A+B) - S^2)$. Next, we have

$$\nabla^2_{\boldsymbol{q}} I(\boldsymbol{z}^*) = -\frac{1}{\pi(1-\pi)} \left(\nabla_{\boldsymbol{q}}\pi\right) \left(\nabla_{\boldsymbol{q}}\pi\right)^\top + \mathbb{E}_W \left[\Psi(\boldsymbol{w}) \left(\nabla_{\boldsymbol{q}} f\right) \left(\nabla_{\boldsymbol{q}} f\right)^\top + \Omega(\boldsymbol{w}) \nabla^2_{\boldsymbol{q}} f\right]$$

with

$$\nabla_{\boldsymbol{q}} f = \frac{-2(\boldsymbol{w} - \boldsymbol{q})}{\sigma_0 S^3} \left[B(A+B)\right],$$

and

$$\nabla^2_{\boldsymbol{q}} f = \frac{2B}{\sigma_0 S^3} \left[(A+B)\mathbf{I} - \frac{2T_{\boldsymbol{q}}}{S^2} (\boldsymbol{w} - \boldsymbol{q})(\boldsymbol{w} - \boldsymbol{q})^\top\right]$$

where $T_{\boldsymbol{q}} = (3A(A+B) - S^2)$. Then

$$\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I)(\boldsymbol{z}^*) = -\frac{1}{\pi(1-\pi)} \left(\nabla_{\boldsymbol{p}}\pi\right) \left(\nabla_{\boldsymbol{q}}\pi\right)^\top + \mathbb{E}_W \left[\Psi(\boldsymbol{w}) \left(\nabla_{\boldsymbol{p}} f\right) \left(\nabla_{\boldsymbol{q}} f\right)^\top + \Omega(\boldsymbol{w}) \nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f)\right]$$

with

$$\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} f) = \frac{4T_{\boldsymbol{pq}}}{\sigma_0 S^5} (\boldsymbol{w} - \boldsymbol{p})(\boldsymbol{w} - \boldsymbol{q})^\top$$

where $T_{\boldsymbol{pq}} = 3A^2(A+B) - S^2(2A+B)$, and by symmetry

$$\nabla_{\boldsymbol{p}}(\nabla_{\boldsymbol{q}} I) = (\nabla_{\boldsymbol{q}}(\nabla_{\boldsymbol{p}} I))^\top.$$

E PAIR OPT-DIST HEURISTIC

We have

$$\mathcal{L}_2(\boldsymbol{p}, \boldsymbol{q}, \eta) = \frac{1}{d} \sum_{i=1}^{d} \frac{\left(\boldsymbol{v}_i^\top (\boldsymbol{b} - \boldsymbol{\mu})\right)^2}{\lambda_i} + \eta \frac{\|(\boldsymbol{p} - \boldsymbol{q}) - 2r^* \boldsymbol{v}_1\|_2^2}{4{r^*}^2}.$$

The first term computes the squared distance between the query midpoint and the posterior mean along each eigenvector, normalized by its corresponding eigenvalue. These projections are summed and divided by d to yield the mean standardized error. The second term measures the geometric misalignment between the normal vector of the query hyperplane and the principal eigenvector, normalized by the target magnitude.

F EXPERIMENT DETAILS

In this section, we provide the experimental details and highlight additional experimental results. The MCMC sampling is implemented using the Stan probabilistic programming language (Carpenter et al., 2017). The MCMC sampling is initialized with the posterior mean of the previous iteration to help it converge faster.

F.1 USER STUDY

Here, we provide an overview of the user study.

DOMAIN AND COLOR SPACE

The experiment operates entirely within the continuous 3-dimensional CIELAB ($L^*a^*b^*$) color space, which is designed to be perceptually uniform.

- L^* (Lightness): $[0, 100]$ (Black to White)
- a^* (Green–Red): $[-128, 127]$
- b^* (Blue–Yellow): $[-128, 127]$

The target color is represented as a true coordinate W within these bounds. We used 3 target colors for the study and for each user, one of them was selected.

At each step, the user is presented with two color swatches, $\boldsymbol{p}$ and $\boldsymbol{q}$. The system predicts the probability that the user prefers $\boldsymbol{p}$ (meaning $\boldsymbol{p}$ is perceptually closer to W than $\boldsymbol{q}$).

ACTIVE LEARNING (QUERY SYNTHESIS) LOOP

The experiment uses Bayesian optimization to sequentially select the most informative color pairs. The system starts with a uniform probability distribution over the entire CIELAB bounding box. After each choice, the posterior distribution of W is updated by draws MCMC samples from the posterior. A pair of colors, providing maximum information about W, is selected at every iteration based on the following synthesis methods. The optimal hyperplane selected by all the methods is the same and they differ in the exact points being selected.

- Confidence-Aware Synthesis: Queries are selected to satisfy the optimality criteria derived in the paper, ensuring that the two points are at an optimal distance apart, neither too close nor too far. as explained in Sec. 4.
- Plain Model Synthesis: The farthest queries that still lie within the embedding space while having the optimal hyperplane are selected.
- Normalized Response Synthesis (Variant A): Queries are selected at a random distance apart, still ensuring that they have the optimal hyperplane and lie within the embedding space.
- Normalized Response Synthesis (Variant B): Queries are selected based on the method in (Chu & Ghahramani, 2005), still ensuring that they have the optimal hyperplane and lie within the embedding space.

NOISE CALIBRATION ACROSS MODELS

We set a value of $\sigma_0 = 0.3$ for the confidence aware model and $\sigma_0 = 2.0$ for the plain model. These values fit the user study participants' responses well on an average. The value for the normalized model was calibrated based on that for the confidence aware model, ensuring that both models assume the same underlying human perceptual error.

EVALUATION METRICS

To evaluate both the convergence speed and how well each generative model captures the user's underlying decision-making process, the experiment tracks the following metrics throughout the active learning loop:

1. Estimation Error (ΔE): The Euclidean distance between the estimated mean of the posterior ($\hat{W}$) and the true hidden target (W).
2. Cross Entropy : Measures cross entropy loss of the predicted probabilities using samples from its current posterior distribution.
3. Choice Prediction Accuracy: The percentage of the user's past choices that the model correctly predicts using samples from its current posterior distribution.
4. Confidence Rating: Participants are asked to rate the confidence of their responses on a scale of 0 to 2, with 0 being the lowest and 2 being the highest.

Note: Both *confidence aware* and *normalized* models were modified slightly to better predict the response probabilities as the colors get closer. For example, the *confidence aware* was modified as

$$P(Y|\boldsymbol{w}) = \Phi\left(\frac{\|\boldsymbol{w}-\boldsymbol{q}\|_2^2 - \|\boldsymbol{w}-\boldsymbol{p}\|_2^2}{\sigma_0\sqrt{\|\boldsymbol{w}-\boldsymbol{q}\|_2^4 + \|\boldsymbol{w}-\boldsymbol{p}\|_2^4 + \epsilon}}\right).$$

This amounts to the noises associated with the distances d_1 and d_2, e_1 and e_2 having variances $\sigma_0^2\left(\|\boldsymbol{w}-\boldsymbol{p}\|_2^4 + \frac{\epsilon}{2}\right)$ and $\sigma_0^2\left(\|\boldsymbol{w}-\boldsymbol{q}\|_2^4 + \frac{\epsilon}{2}\right)$ respectively. This modification doesn't affect the optimality criterion derived in Sec. 4. It affects only the optimal magnitude which is handled by the 1D optimization. This enables the model to predict probabilities closer to 0.5 when the colors are very close. A similar modification is applied to the *normalized* model. The *plain* model doesn't need any modification since the probability naturally goes to 0.5 as the closers get closer.

F.2 SYNTHETIC DATASET EXPERIMENTS

F.2.1 EXPERIMENTAL SETUP

The synthetic dataset is constructed by creating an embedding of N items in $\mathbb{R}^D$ by drawing samples from $U[-4,4]^D$ and a user point was drawn from $U[-1,1]^D$. Responses were obtained for the paired comparison queries selected according to the corresponding AL method and the estimate of the user point was updated. The experiments were conducted for 5 trials and for every trial, a new user point was sampled. Results for the user point estimation error were averaged over trials for 100 queries.

F.2.2 k-NN APPROX METHOD

Considering k nearest neighbors of the points gives us access to all the surrounding points that may approximate the optimal pair of points' geometry more closely than the immediate neighbors. In this method, we perform a k-NN search for the individual points and then evaluate each of the combination of points for optimality.

Let $\mathbb{P} = k\text{-NN}(\boldsymbol{p}^*)$ and $\mathbb{Q} = k\text{-NN}(\boldsymbol{q}^*)$. The points are selected as below:

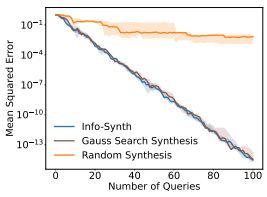
$$(\boldsymbol{p}, \boldsymbol{q}) = \underset{\boldsymbol{p}^*\in\mathbb{P},\ \boldsymbol{q}^*\in\mathbb{Q}}{\arg\max}\ I(W; Y|(\tilde{\boldsymbol{p}}, \tilde{\boldsymbol{q}})).$$

This method again uses a filter and refine strategy. The top k nearest points are computed using Euclidean distances and the top α fraction of the pairs undergo MI evaluation to select the best approximation.

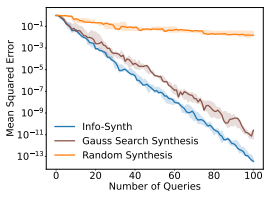
F.2.3 Additional Results

Variation of Synthesis performance with σ_0 values

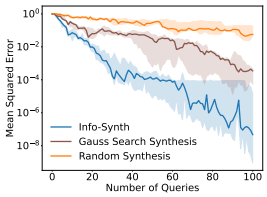
We ran experiments considering more noise levels $\sigma_0 = [0.01, 0.05, 0.2, 0.3, 0.5]$. As noise level increases, the overall performance is worse, with *Gauss Search Synthesis* worsening more than our method, *Info-Synth*. All results are for data in 4D.



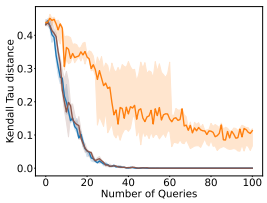
(a) $\sigma = 0.01$



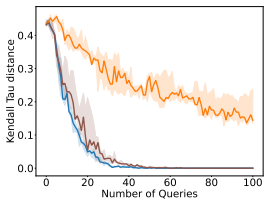
(b) $\sigma = 0.05$



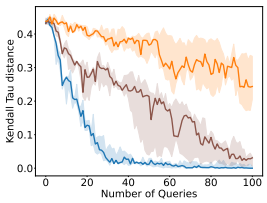
(c) $\sigma = 0.2$



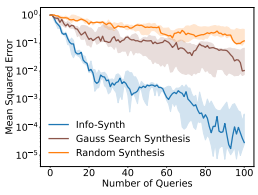
(a) $\sigma = 0.01$



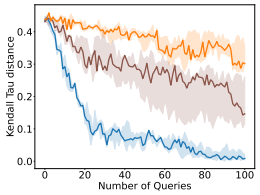
(b) $\sigma = 0.05$



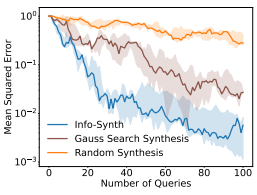
(c) $\sigma = 0.2$



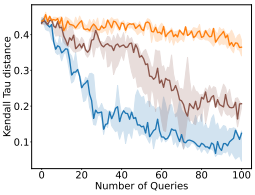
(a) $\sigma = 0.3$



(b) $\sigma = 0.3$



(c) $\sigma = 0.5$



(d) $\sigma = 0.5$

Synthesis comparison with discrete methods

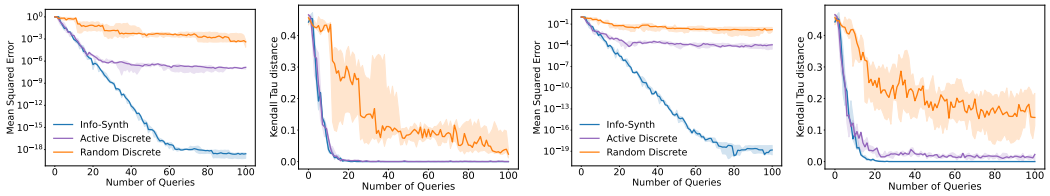


Figure 11: Results for $D = 2$ and $N = 500$ for $\sigma_0 = 0.001$ (left) and $\sigma_0 = 0.1$ (right).

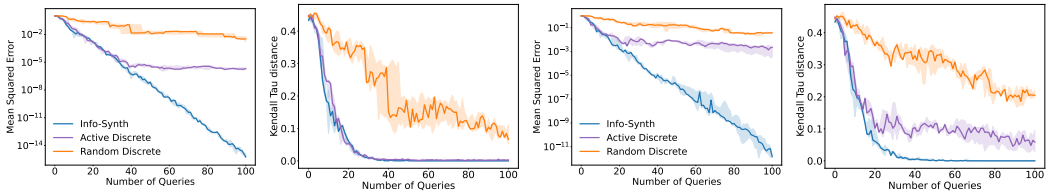


Figure 12: Results for $D = 4$ and $N = 500$ for $\sigma_0 = 0.001$ (left) and $\sigma_0 = 0.1$ (right).

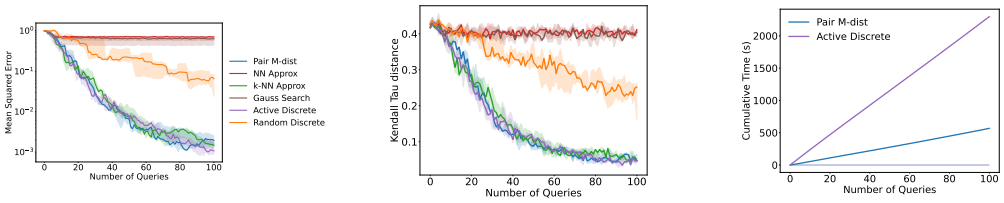


Figure 13: Results for $D = 10$, $N = 500$ and $\sigma_0 = 0.01$.

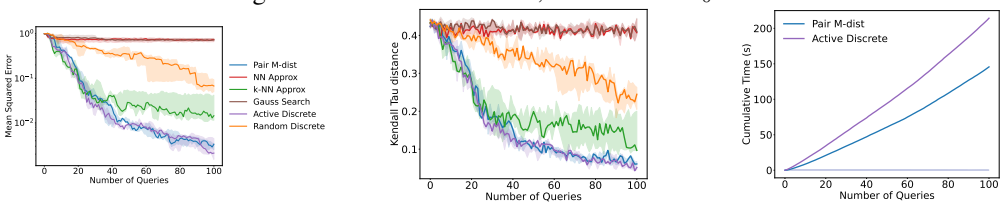


Figure 14: Results for $D = 10$, $N = 100$ and $\sigma_0 = 0.01$.

Discrete Comparison

Ablation studies

Synthesis Methods

The synthetic dataset experiments with synthesis performance comparison themselves serve as an ablation separating the gains of our response model from query geometry. Both, *Info-Synth* and *Gauss Search Synthesis* optimize for queries with similar geometries. The only difference being the former also considers the query distances but the latter doesn't. This specifically isolates and demonstrates the performance gain derived directly from our confidence-aware response model and the optimized distances, rather than just the query geometry.

Discrete Methods

The finite dataset approximations, Pair M-dist and Pair Opt-dist follow a filter and refine strategy as explained in Sec. 5. The refine part is similar to Active Discrete and hence this also serves as an ablation. We also conducted experiments on Pair M-dist using only the filtering part (selecting the top query based on this). We refer to this method as *Nearest M-dist*. This method performs poorly compared to Active Discrete since the filtering is only an approximate criterion. Results illustrated in Fig. 15.



Figure 15: Results for $D = 10$, $N = 100$ and $\sigma_0 = 0.01$.

User Mismatch

We ran experiments to test user mismatch for both (i) *plain* probability model (where queries with larger gap are preferred) and (ii) *normalized* model (the preference is independent of the gap between the items, only the query geometry is considered). The results observed for different experiments are as below:

Synthesis Comparison

In both scenarios (i) and (ii), both *Gauss Search Synthesis* and *Info-Synth* achieve a similar performance. *Info-Synth* differs from *Gauss Search Synthesis* only in the part where queries with optimal

gap are selected and both methods select queries with similar geometries. Thus when the optimal gap assumption in the user is removed, *Info-Synth* no longer has an advantage. In *Gauss Search Synthesis*, the gap between the selected queries can be random so it has no advantage with user mismatch either. At a higher noise level, both the active methods lose advantage over *Random* beyond a certain point. Results are illustrated in Figs. 16 and 17.

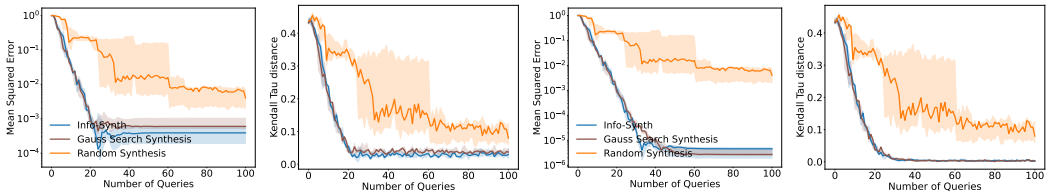


Figure 16: Results for $D = 4$ and $\sigma_0 = 0.001$ with the user following the *Plain* (left) and *Normalized* (right).

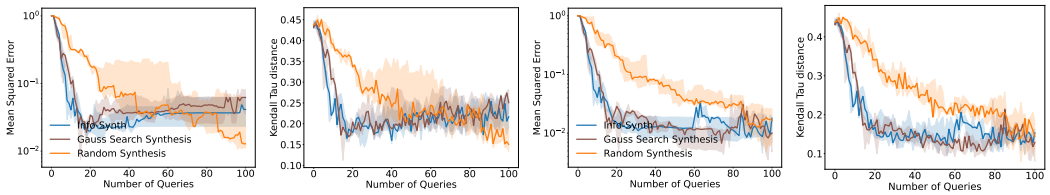


Figure 17: Results for $D = 4$ and $\sigma_0 = 0.1$ with the user following the *Plain* (left) and *Normalized* (right).

Synthesis v/s Discrete Comparison

In both scenarios, *Info-Synth* performs worse than *Active Discrete*, with the difference being larger for (i) than (ii). Since the continuous synthesis is driven by a fundamentally incorrect objective function, its primary advantage disappears and querying a continuous pair no longer results in better results than a finite dataset. Even though *Active Discrete* is also mismatched, it might occasionally sample highly informative points (e.g., points extremely far apart in (i)) purely by chance within the pool, whereas *Info-Synth* would systematically and incorrectly zero in on a suboptimal moderate distance. Results are illustrated in Figs. 18, 19 and 20.

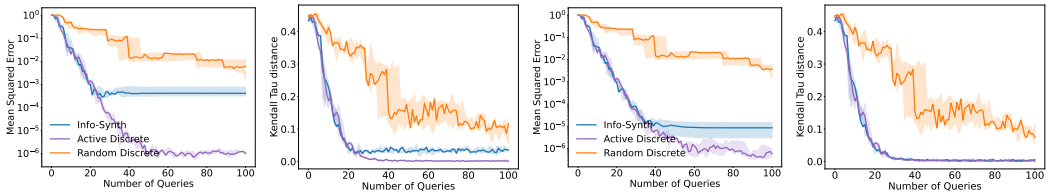


Figure 18: Results for $D = 4$ and $\sigma_0 = 0.001$ with the user following the *Plain*(left) and *Normalized*(right).

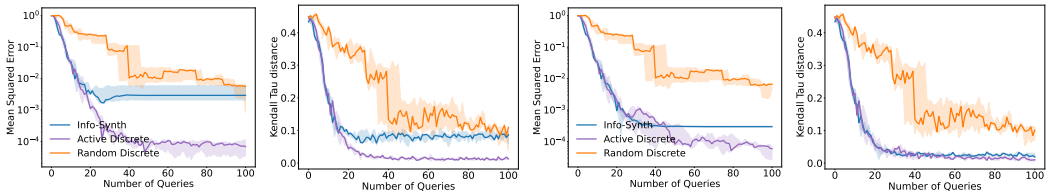


Figure 19: Results for $D = 4$ and $\sigma_0 = 0.01$ with the user following the *Plain*(left) and *Normalized*(right).

Discrete Comparison

In both scenarios, the overall performance is worse but the relative performance of the methods remains the same. The filter and refine strategy employed in *Pair M-dist* makes it robust for model mismatch. The refine part is essentially similar to the strategy employed in *Active Discrete* but only

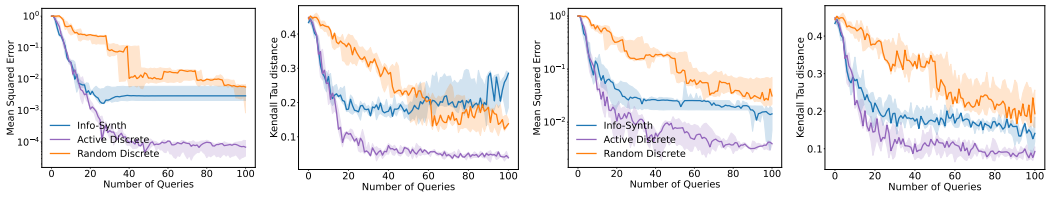


Figure 20: Results for $D = 4$ and $\sigma_0 = 0.1$ with the user following the *Plain*(left) and *Normalized*(right).

the top queries filtered in the first part are evaluated. This makes it computationally more efficient than *Active Discrete* while retaining the robustness of *Active Discrete*. Results are illustrated in Figs. 21 and 22.



Figure 21: Results for $D = 10$ and $N = 100$ for $\sigma_0 = 0.01$ with the user following the *Plain* model.



Figure 22: Results for $D = 10$ and $N = 100$ for $\sigma_0 = 0.01$ with the user following the *Normalized* model.

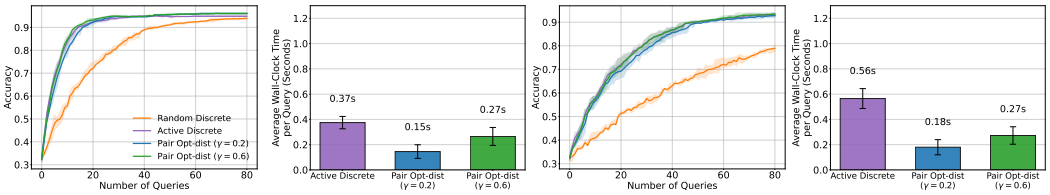
F.3 Reddit Summary Dataset Experiments

F.3.1 Experimental setup

For the results shown in Figure 6, our chosen user (**User 1**) provided 1,874 paired query responses. Compared to other users in this dataset, this specific user had a relatively larger number of paired query responses. We used 70% of these queries as the active learning pool for learning the user response, and the remaining 30% for evaluating accuracy. The original embedding size for these representations was 512. To speed up these active learning experiments, particularly for the *Active Discrete* baselines, we used PCA to project embedding dimensions to 128. We provide results in the raw embedding dimension without PCA later in Figure 25.

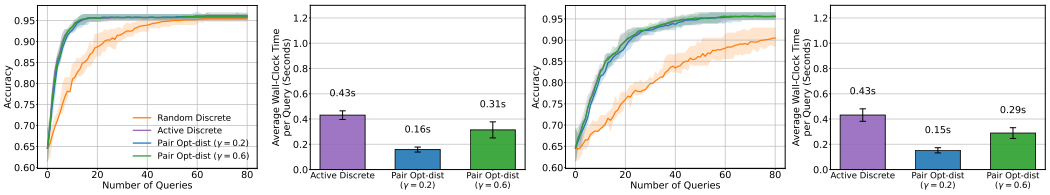
F.3.2 Results for additional users

In addition to the results for User 1 in Figure 6, we show the results for two additional users in Figures 23 and 24. The total number of queries available for these two users are 1,010 and 837 respectively.



(a) $\sigma = 0.1$ (b) $\sigma = 0.1$ (c) $\sigma = 0.3$ (d) $\sigma = 0.3$

Figure 23: **User 2** in Reddit Summary TL;DR dataset. This user's dataset provided 1010 paired comparisons for querying. We used 70 % of these queries for approximating the response, and the rest of the queries for evaluation. Our approximation method, *Pair Opt-dist* at two different gamma values can be seen to provide competitive active query performance with a much smaller query response time. We show results for both $\sigma = 0.1$ and $\sigma = 0.3$

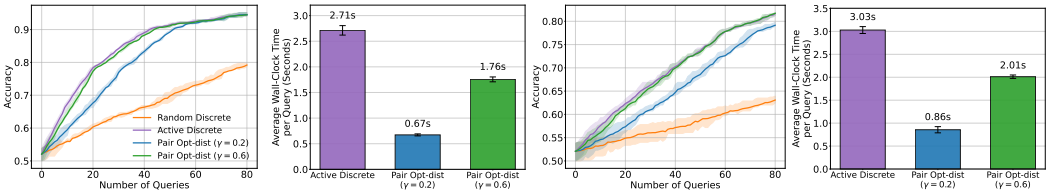


(a) $\sigma = 0.1$ (b) $\sigma = 0.1$ (c) $\sigma = 0.3$ (d) $\sigma = 0.3$

Figure 24: **User 3** in Reddit Summary TL;DR dataset. This user's dataset provided 837 paired comparisons for querying. We used 70 % of these queries for approximating the response, and the rest of the queries for evaluation. Our approximation method, *Pair Opt-dist* at two different gamma values can be again seen to provide competitive active query performance with a much smaller query response time. We show results for both $\sigma = 0.1$ and $\sigma = 0.3$

F.3.3 Results in raw embedding space without PCA

Our experiments for User 1 in Figure 6 were conducted after reducing the 512 dimensional original embedding space to 128 dimensions using PCA transformation. In Figure 25 we provide results on the 512 dimensional embedding space without any dimensionality reduction. As compared to the 128 dimensional case, all methods require more queries to converge. We also observe that our *Pair Opt-dist* method with $\gamma = 0.2$ lags $\gamma = 0.6$ by a wider margin. *Pair Opt-dist* with $\gamma = 0.6$ remains competitive with Active Discrete with lesser query approximation cost.



(a) $\sigma = 0.1$ (b) $\sigma = 0.1$ (c) $\sigma = 0.3$ (d) $\sigma = 0.3$

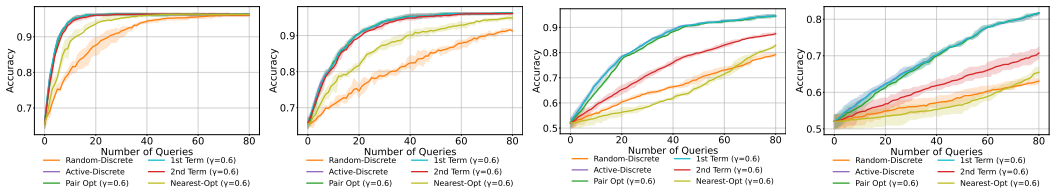
Figure 25: **User 1** in Reddit Summary TL;DR dataset on full 512 dimensional embedding space. As compared to results in Figure 6 that apply dimensionality reduction on the same dataset, all AL methods require more queries to converge. *Pair Opt-dist* at $\gamma = 0.6$ remains competitive with Active Discrete while providing a smaller query approximation cost. Though the performance for $\gamma = 0.2$ lags further, the query approximation cost is much cheaper. We show results for both $\sigma = 0.1$ and $\sigma = 0.3$

F.3.4 Ablation of Pair-Opt terms

To better analyze the contribution of each term in the heuristic score (7), we conduct ablations by including only one of these two terms at a time. We provide results for these in Figure 26 for User 1 in both 128 dimensional and the original 512 dimensional settings. In the higher dimensional setting, we can see that the gap between the first term (cyan) and the second term(red) is much larger than it is for the 128 dimensional case where they are both close. With larger dimensionality, the informative direction for query selection might align to a lesser degree with the first eigenvector

of the data space which is prioritized by the second term. Since it is a smaller dataset, it might not contain pairs that satisfy this query direction and imposing it as a criterion might be too strict.

We also provide results for *Nearest-Opt* ablation in yellow. This method just picks the top query corresponding to the *Pair Opt-dist* heuristic score. This thus requires no mutual information computation, resulting in a very fast query response time of about 0.06 seconds. Though its performance deteriorates severely when dimensionality increases.



(a) 128 dim. ($\sigma = 0.1$) (b) 128 dim. ($\sigma = 0.3$) (c) 512 dim. ($\sigma = 0.1$) (d) 512 dim. ($\sigma = 0.3$)

Figure 26: Ablations on **User 1** in Reddit Summary TL;DR dataset for both 128 dimensional 512 dimensional settings. We analyze how the performance of *Pair Opt-dist* changes when we include only the first term (1st Term label in cyan) and the second term (2nd Term label in red) when computing the heuristic score in (7). We see that the gap between these increases, and the second term hurts performance of the combined term in green in the higher dimensional space.

F.4 Gain Tuning

The objective of the experiments is for the unicycle to smoothly track the trajectory. We seek to identify the optimal control parameters that accomplish this objective in as few experiments as possible, relying only on comparative feedback "A is better than B") rather than numerical gradient values.

We simulate a human-in-the-loop tuning process where the learner maintains a probabilistic belief about where the optimal parameters lie in the log-parameter space. We use different acquisition functions to select the next query pair - two different sets of gains (K_A, K_B) that will yield the most information. For each query pair, the system simulates a unicycle robot tracking a specific path starting from various initial states. Instead of a real human, we use an automated oracle that evaluates the simulated trajectories based on the trajectory tracking error (computed using Hausdorff distance). To simulate real human inconsistency, the oracle uses a sigmoid function to generate a preference probability: if Trajectory A is much better than B, the probability of picking A is high (near 1.0); if they are similar, it approaches a coin flip (0.5). To ensure the learned gains are robust, the oracle averages the error across multiple trajectories or multiple initial states before deciding which set of gains is superior.

The reference paths are generated using Bézier curves, providing smooth, continuous trajectories defined by a set of control points. A Bézier curve is defined by $n + 1$ control points, and the robot's target position at any phase $t \in [0, 1]$ is computed using the De Casteljau algorithm Farouki (2012).