PREVAIL: Disentangling Preference Distributions for Generative Ranking

Zhangkai Wu
The University of Sydney
zhangkai.wu@sydney.edu.au

Kaize Shi
University of Southern Queensland
Kaize.Shi@unisq.edu.au

Xu Zhang
Macquarie University
xu.zhang12@hdr.mq.edu.au

Zhihong Cui
University of Oslo
zhihongc@ifi.uio.no

Longbing Cao*
Macquarie University
longbing.cao@mq.edu.au

Abstract

Preference ranking modeling is important for selecting and adapting language-model responses to human preferences. Compared with deterministic ranking methods, generative preference ranking models the prompt-response data distribution, providing richer uncertainty characterization and sample diversity for learning a more expressive ranking boundary. Recent work improves modeling efficiency by shifting generative preference ranking from textual space to embedding space, but without explicit preference-conditioned separation, the learned embedding distribution may still entangle preferred and rejected responses, producing ambiguous synthetic samples that weaken the ranking boundary. To address this issue, we propose PREVAIL, a **PRE**ference **VA**riational d**I**sentang**L**ement framework for generative ranking. PREVAIL formulates embedding-level generative preference ranking with a unified class-conditional ELBO and decomposes this optimization objective to expose the encoded latent entanglement between preferred and rejected embeddings. This derivation motivates two practical preference-aware generation variants: PREVAIL-MI, involving a deep class-separation objective, and PREVAIL-MMD, featuring a probabilistic aggregate-matching objective. Both variants learn a preference-disentangled embedding data distribution and synthesize data pairs that preserve ranking semantics for the generative ranking boundary. Experiments show that PREVAIL improves preference ranking performance and produces useful generated embeddings for downstream response selection. Code is available at https://github.com/amasawa/PREVAIL.

1 Introduction

Aligning large language model outputs with human preferences is essential for building reliable AI systems [1–3]. A preference-ranking function that captures linguistic semantics plays a central role in comparing and selecting candidate responses [4, 5] and in evaluating response quality across tasks [6–8]; it also provides the optimization signal for reinforcement-learning-based alignment [9] and preference-optimization methods for model adaptation [10–12].

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

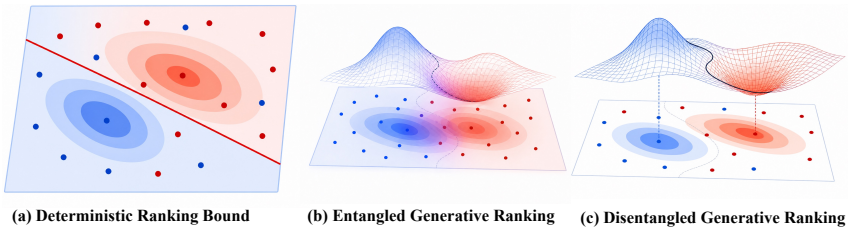


Figure 1: Comparison of preference ranking paradigms. **(a) Deterministic Ranking** learns a conditional decision boundary $p(y \mid \mathbf{u})$ that separates preferred (blue) from rejected (red), but does not model the underlying data distribution. **(b) Entangled Generative Ranking** models the joint distribution $p(\mathbf{u}, y)$ and enables preference-consistent synthesis, but preferred and rejected samples may remain entangled in the learned distribution. **(c) Disentangled Generative Ranking** explicitly separates preferred and rejected embedding distributions, yielding cleaner synthetic pairs for learning a more reliable and expressive ranking boundary.

Preference ranking [13, 14] is commonly formulated as learning a conditional boundary $p(y \mid \mathbf{u})$ over a prompt–response pair $\mathbf{u} = (\mathbf{x}, \mathbf{y})$ and a binary preference label y, where $\mathbf{x}$ denotes the prompt and $\mathbf{y}$ denotes the response. Recent advances strengthen this boundary with richer LLM-assisted supervision, including critiques [15], natural-language feedback [16], self-labeled comparisons [17], and fine-grained criteria [18]. Despite the effectiveness of LLM-assisted signals in strengthening the ranking boundary, these methods still focus on conditional discrimination over $p(y|\mathbf{u})$, as illustrated in Fig. 1(a), leaving the underlying data generating structure unmodeled and providing no structural basis for disentangling preferred from rejected responses. Moreover, this discriminative boundary can degrade due to data scarcity and high annotation costs arising from safety and privacy issues [19, 20].

In contrast, generative preference ranking models [21–25] the joint distribution $p(\mathbf{u}, y)$, enabling synthesis of more preference pairs that alleviates data scarcity and supports learning a more informative ranking boundary [26, 27]. LENS [24] shows that generative preference ranking at the prompt–response embedding level can improve efficiency. However, without a disentanglement guarantee, preferred and rejected embeddings may become entangled in the learned data distribution, leading to an ambiguous ranking boundary, as illustrated in Fig. 1(b). This gap calls for a unified variational learning framework that can model prompt–response embedding distributions and expose the preference entanglement that affects generative ranking. We summarize deterministic and generative preference-ranking learning in App. A.

This motivates a disentangled generative objective that models preference-conditioned embeddings while explicitly separating preferred and rejected latent embeddings, as shown in Fig. 1(c). To keep the efficiency of embedding-space generation while enforcing preference-level disentanglement in the latent distribution, we develop PREVAIL, a **PRE**ference **VA**riational d**I**sentang**L**ement framework for generative ranking. Our main contributions are:

- We introduce a class-conditional ELBO (cELBO) as a unified learning framework for generative embedding ranking models, and use its decomposition to motivate preference-aware latent generation. Unlike discriminative ranking, which learns only the boundary $p(y \mid \mathbf{u})$, and LENS, which appends a heuristic penalty, the decomposition derives the separation forces from the variational objective itself, as a within-class information term and a class-wise aggregate matching term.
- We develop two practical variants of PREVAIL: PREVAIL-MI uses a deep mutual-information formulation, while PREVAIL-MMD uses a probabilistic MMD formulation for preference-disentangled latent generation.
- We generate preference-disentangled embedding pairs from PREVAIL and improve preference ranking, with benefits for downstream response selection.

2 Prompt-Response Embedding Distribution Modeling

In preference-based LLM alignment [9, 10], preference-consistent generation provides a practical way to alleviate the scarcity of human-labelled prompts, thereby improving the alignment between model outputs and human feedback. We provide a generative joint distribution modeling over

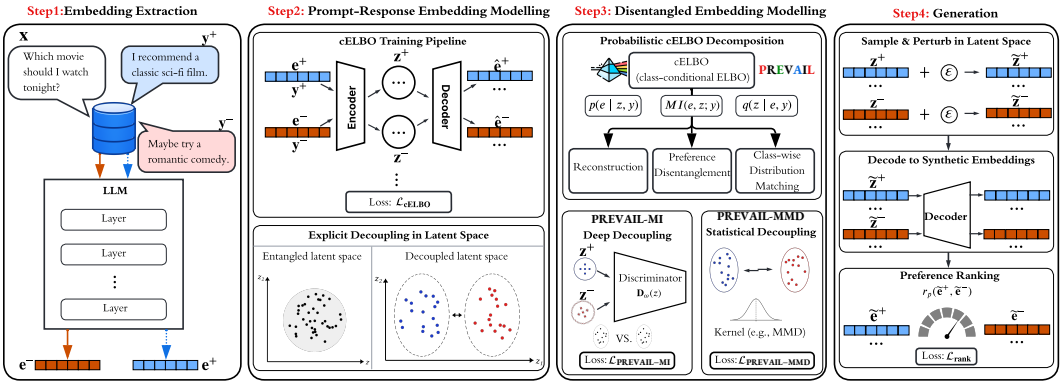


Figure 2: Starting from prompt–response pairs, we first extract the embedding $\mathbf{e}$ and model its preference-conditioned distribution with the class-conditional ELBO in Eq. (1). The ELBO decomposition then motivates two practical PREVAIL variants: a deep class-separation formulation in Eq. (4) and a statistical aggregate-matching formulation in Eq. (5). After learning the preference-disentangled latent structure, the generator synthesizes additional embedding pairs.

embedding level for preference ranking. The overall PREVAIL workflow is summarized in Fig. 2. Let $\mathbf{x}$ denote a prompt, $\mathbf{y}^+$ and $\mathbf{y}^-$ denote the preferred and rejected responses, $\mathbf{u}^\pm = (\mathbf{x}, \mathbf{y}^\pm)$ denote prompt–response pairs, and $y \in \{0, 1\}$ denote the binary preference label. A frozen LLM encoder maps each prompt–response pair to its embedding, $\mathbf{e}^\pm = \mathrm{LLMembed}(\mathbf{x}, \mathbf{y}^\pm)$[2]. For a generic labeled sample, $\mathbf{e}$ denotes either $\mathbf{e}^+$ with $y = 1$ or $\mathbf{e}^-$ with $y = 0$. This deterministic map pushes the raw joint distribution $p(\mathbf{u}, y)$ to an embedding-level joint distribution $p(\mathbf{e}, y)$. Therefore, modeling $p(\mathbf{e}, y) = p(y \mid \mathbf{e})p(\mathbf{e})$ is our embedding-level formulation of the generative ranking objective $p(\mathbf{u}, y)$, with the marginal term $p(\mathbf{e})$ enabling embedding-space generation. Given the embedding-level supervised dataset $\mathcal{E} = \{(\mathbf{e}_i, y_i)\}_{i=1}^N$, preference ranking learns a scalar $s_\psi(\mathbf{e})$ such that preference is captured by $\sigma\big(s_\psi(\mathbf{e}^+) - s_\psi(\mathbf{e}^-)\big)$. To mitigate limited preference data and the cost of human annotation, Tao et al. [24] first train a response-embedding VAE, i.e., $\mathcal{E}((\mathbf{e}, y)^n; \theta, \phi)$ with encoder $q_\phi(\mathbf{z} \mid \mathbf{e})$ and decoder $g_\theta(\mathbf{z})$, and then perform latent-space synthesis by sampling $\widetilde{\mathbf{z}} = \mathbf{z} + \epsilon$ and decoding $\widetilde{\mathbf{e}} = g_\theta(\widetilde{\mathbf{z}})$ to construct generated embeddings for post-hoc training of s_ψ.

During the two-stage preference-ranking pipeline, it is crucial to learn an embedding VAE that separates positive and negative embeddings well in latent space, which enables subsequent preference ranking and data synthesis. We formalize this step as learning a prompt-response disentangled embedding VAE, concentrating on the prompt-response marginal embedding likelihood as a two-component mixture, $p_\theta(\mathbf{e}) = \sum_{y\in\{0,1\}} \pi_y \int p_\theta(\mathbf{e} \mid \mathbf{z})\, p(\mathbf{z} \mid y)\, \mathrm{d}\mathbf{z}$, where π_y is the mixture weight and $p(\mathbf{z} \mid y)$ is the component-specific latent distribution.

During the inference stage, Tao et al. [24] introduce a hierarchical prior, i.e., $p(\mathbf{z}) = \sum_{y\in\{0,1\}} p(\mathbf{z} \mid y)\, p(y)$, to incorporate binary label information. Specifically, they employ a structured latent variable $(y, \mathbf{z})$, where $y \in \{0, 1\}$ is a discrete indicator with $p(y) = \mathrm{Cat}(\pi)$ and $\mathbf{z}$ is a continuous gaussian variable with $p(\mathbf{z} \mid y) = \mathcal{N}(0, I)$. The aggregated posterior is parameterized by the encoder as $q_\phi(y, \mathbf{z} \mid \mathbf{e}) = q_\phi(y \mid \mathbf{e})\, q_\phi(\mathbf{z} \mid \mathbf{e}, y)$, where $q_\phi(y \mid \mathbf{e})$ performs amortized assignment and $q_\phi(\mathbf{z} \mid \mathbf{e}, y)$ defines class-specific posteriors. During the generation stage, the decoder $p_\theta(\mathbf{e} \mid \mathbf{z})$ induces the mixture marginal $p_\theta(\mathbf{e}) = \sum_{y\in\{0,1\}} \pi_y \int p_\theta(\mathbf{e} \mid \mathbf{z})\, p(\mathbf{z} \mid y)\, \mathrm{d}\mathbf{z}$.

Specifically, maximizing the prompt-response marginal embedding likelihood equals to optimize a class-conditional ELBO that contains *reconstruction* under $p_\theta(\mathbf{e} \mid \mathbf{z})$ and a class-wise KL term; the decomposition below further separates the latter into *within-class information* and *class-wise aggregate matching*.

$$\mathcal{L}_{\texttt{cELBO}}(\theta, \phi) := \mathbb{E}_{q(\mathbf{e})}\Big[\mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{e})}\big[\log p_\theta(\mathbf{e} \mid \mathbf{z})\big] - \mathbb{E}_{q_\phi(y|\mathbf{e})}\mathrm{D}_{\mathrm{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{e}, y) \,\|\, p(\mathbf{z} \mid y)\big)\Big], \tag{1}$$

where $q(\mathbf{e}) = \frac{1}{B}\sum_{n=1}^B \delta_{\mathbf{e}^n}(\mathbf{e})$ denotes the empirical embedding distribution within a mini-batch, and the normalized class-conditional aggregated posteriors are $\bar{q}_\phi^{(y)}(\mathbf{z}) := \frac{1}{B_y}\sum_{n=1}^B q_\phi(y \mid \mathbf{e}_n)\, q_\phi(\mathbf{z} \mid$

[2]Processes each prompt–response pair through the language model and returns the final hidden-state representation at the last layer.

$\mathbf{e}_n, y)$, where $B_y = \sum_{n=1}^{B} q_\phi(y \mid \mathbf{e}_n)$ and $y \in \{0, 1\}$. In our setting, we use observed labels and set $q_\phi(y \mid \mathbf{e}_i) = \delta_{y=y_i}$, similar to [28, 29]; thus, the label-assignment KL term is omitted from the optimized objective.

3 Disentangled Prompt-Response Embedding Modeling

To acquire disentangled prompt–response embeddings, [24] integrate a divergence term, i.e.,

$$\mathcal{L}_{\texttt{Lens}}(\theta, \phi) \;:=\; \mathcal{L}_{\texttt{cELBO}} - \lambda\, W_1\big(\bar{q}_\phi^{(1)},\, \bar{q}_\phi^{(0)}\big), \tag{2}$$

where $W_1(\cdot)$ denotes Wasserstein distance. It can be interpreted as pursuing two objectives: (i) maximizing a class-conditional ELBO over prompt–response embeddings, and (ii) minimizing a mini-batch-based, biased distance surrogate that indirectly encourages preferred and unpreferred embeddings to separate in latent space.

In contrast, we treat the cELBO as a unified learning framework for generative embedding ranking models, where distribution modeling, class-conditioned latent structure, and synthetic embedding generation are connected through the same variational objective. The decomposition identifies two distinct forces: a within-class information term and a class-wise aggregate matching term. Based on this decomposition, we design practical surrogate objectives that add class-separation signals while retaining the reconstruction and instance-level VAE regularization used in training.

Decomposed cELBO. Let $\mathcal{D}_B = \{(\mathbf{e}_i, y_i)\}_{i=1}^{B}$ denote a mini-batch of prompt–response embeddings with binary preference labels $y \in \{0, 1\}$. Ignoring the Wasserstein separation term in Eq. (2), the remaining class-conditional ELBO admits a decomposition that makes within-class latent information and class-wise aggregate matching explicit. This decomposition motivates practical objectives for learning a generative preference-ranking boundary, but the conditional mutual-information term itself should not be interpreted as a direct preferred/rejected separation penalty. Specifically, by introducing the class-conditional aggregated posterior $\bar{q}_\phi^{(y)}(\mathbf{z})$ and collecting terms by class, we obtain the following decomposed objective:

$$\begin{aligned}\mathcal{L}_{\mathrm{cELBO}}(\theta, \phi) = &\underbrace{\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y, \mathbf{z} \mid \mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big]}_{\textit{reconstruction}} \\ &- \underbrace{\sum_{y \in \{0,1\}} B_y\, I_\phi(\mathbf{e}; \mathbf{z} \mid Y = y)}_{\textit{within-class information}} - \underbrace{\sum_{y \in \{0,1\}} B_y\, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z}) \,\|\, p(\mathbf{z} \mid y)\big)}_{\textit{class-wise aggregate matching}},\end{aligned} \tag{3}$$

where $B_y = \sum_{i=1}^{B} q_\phi(y \mid \mathbf{e}_i)$ is the (soft) class count within the mini-batch. The three terms respectively perform *reconstruction*, control how much instance-specific information is retained within each preference class through *within-class information*, and match each class-conditional aggregated posterior to its class-specific latent distribution through *class-wise aggregate matching*. Refer to App. C for the full derivation. In the next subsection, we show how to use the observed preference class to build practical class-separation surrogates from samples.

3.1 Deep Mutual-Information Variant (PREVAIL-MI)

A direct implementation of Eq. (3) still requires evaluating the *class-wise aggregate matching* term $\sum_y B_y\, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z}) \,\|\, p(\mathbf{z} \mid y)\big)$, which is intractable when $\bar{q}_\phi^{(y)}(\mathbf{z})$ is only accessible through samples. To obtain an explicit, sample-based class-separation signal, we use a plug-in surrogate that replaces the component prior $p(\mathbf{z} \mid y)$ by the mini-batch mixture $\bar{q}_\phi(\mathbf{z}) = \sum_y \hat{\pi}_y\, \bar{q}_\phi^{(y)}(\mathbf{z})$. This yields the class-label mutual information $I_\phi(Y; \mathbf{z})$ under the aggregated joint $\bar{q}_\phi(y, \mathbf{z})$ (full derivation in App. D), which measures whether the latent code preserves preference identity.The resulting PREVAIL-MI objective is therefore a cELBO-motivated practical surrogate: it retains instance-level VAE regularization and adds a discriminator-based class-separation regularizer rather than exactly optimizing the decomposed cELBO.

Concretely, we estimate the required density ratio by training a discriminator $D_\omega(\mathbf{z}) \approx \bar{q}_\phi(y{=}1 \mid \mathbf{z})$ as in density-ratio estimation [30, 31], and obtain the surrogate loss in a deep manner:

$$\begin{aligned}\mathcal{L}^{\mathrm{MI}}_{\mathrm{PREVAIL}}(\theta,\phi,\omega) \;:=\; & \sum_{i=1}^{B} \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{e}_i,y_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \beta \sum_{i=1}^{B} \mathrm{D}_{KL}\big(q_\phi(\mathbf{z}\mid\mathbf{e}_i,y_i)\,\|\,p(\mathbf{z}\mid y_i)\big) \\ & - \lambda B\,\widehat{\mathcal{C}}_\omega, \\ \widehat{\mathcal{C}}_\omega \;:=\; & -\Big(\hat{\pi}_1\,\mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(1)}}[\log D_\omega(\mathbf{z})] + \hat{\pi}_0\,\mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(0)}}[\log(1-D_\omega(\mathbf{z}))]\Big),\end{aligned} \tag{4}$$

where $\hat{\pi}_y := B_y/B$ and $\bar{q}_\phi^{(y)}(\mathbf{z})$ is the class-conditional aggregated posterior estimated within the mini-batch. In our setting, we use the observed preference labels and take $q_\phi(y \mid \mathbf{e}) = \delta_{y=y_i}$, hence B_y is the hard class count. Unlike continuous MI estimators [30] that rely on sample shuffling to approximate marginal distributions, our discriminator is trained directly on samples from the observed class-conditional posteriors $\bar{q}_\phi^{(1)}$ and $\bar{q}_\phi^{(0)}$, yielding a direct density-ratio estimate without the approximation error introduced by shuffling. Here β controls instance-level prior regularization, which can contract latent codes toward the prior, while λ controls the discriminator-based preference-identification signal.

3.2 Probabilistic MMD Variant (PREVAIL-MMD)

As an alternative that avoids density-ratio estimation, we further rewrite the class-wise aggregate matching part of the decomposed cELBO to expose an explicit trade-off between (i) *aggregated distribution matching* and (ii) *per-sample* latent-information pressure, following the InfoVAE/InfoDiffusion reformulation [32, 33] (see App. E for details). We then replace the *aggregated distribution matching* term by a kernel two-sample distance (MMD), which is directly estimable from samples and constrains the mixture $\bar{q}_\phi(\mathbf{z}) = \sum_y \hat{\pi}_y\,\bar{q}_\phi^{(y)}(\mathbf{z})$ to remain close to the prior. This MMD term is a global aggregate-matching regularizer, not a direct class-separation penalty. This yields the discriminator-free surrogate in a statistical manner:

$$\begin{aligned}\mathcal{L}^{\mathrm{MMD}}_{\mathrm{PREVAIL}}(\theta,\phi) \;:=\; & \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - (\alpha+\gamma-1)\,B\cdot \mathrm{MMD}^2_k\big(\bar{q}_\phi(\mathbf{z}),\,p(\mathbf{z})\big) \\ & - (1-\gamma)\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y\mid\mathbf{e}_i)}\mathrm{D}_{KL}\big(q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)\,\|\,p(\mathbf{z}\mid y)\big),\end{aligned} \tag{5}$$

where α, γ regulate the trade-off between global aggregate matching and per-sample latent-information pressure. When $\gamma > 1$, the instance-level KL term is no longer a standard VAE contraction term; it encourages latent codes to retain sample-specific variation under the global MMD constraint, while class-conditioned posterior heads make this variation preference-aware.

3.3 Preference Ranking Modeling

Optimization. Both strategies are trained on mini-batches of prompt–response embeddings with observed preference labels. For the deep MI variant Eq. (4), we alternately update the discriminator D_ω using the binary cross-entropy on $\mathbf{z} \sim \bar{q}_\phi^{(y)}$ and update (θ, ϕ) to maximize $\mathcal{L}^{\mathrm{MI}}_{\mathrm{PREVAIL}}$. The alternation is required by the learned density-ratio estimator: each encoder update shifts the class-conditional latent distributions on which D_ω is trained, so D_ω is refreshed to track the current ratio before it supplies the preference-identification signal to (θ, ϕ). This block-coordinate schedule is more stable than jointly updating an under-trained discriminator and a moving representation. For the statistical variant Eq. (5), we directly optimize (θ, ϕ) by maximizing $\mathcal{L}^{\mathrm{MMD}}_{\mathrm{PREVAIL}}$, where the MMD term is computed from sampled latent codes in the mini-batch; as an empirical kernel statistic it needs no learned estimator and hence no alternating update. In both cases, the learned latent structure is shaped by a balance between instance-level regularization, class-conditioned posterior heads, and either a discriminator-based class-separation signal (PREVAIL-MI) or a global aggregate-matching constraint with per-sample information pressure (PREVAIL-MMD). This balance supports preference-aware latent synthesis for downstream ranking.

Preference ranking with disentangled embedding generation. After learning a preference-disentangled generator (either Eq. (4) or Eq. (5)), we train the preference ranker with additional generated embedding pairs. For each labeled embedding $(\mathbf{e}_i, y_i)$, we sample $\widetilde{\mathbf{z}}_i \sim q_\phi(\mathbf{z} \mid \mathbf{e}_i, y_i)$ by applying a small perturbation $\widetilde{\mathbf{z}}_i = \mathbf{z}_i + \epsilon$, then decode $\widetilde{\mathbf{e}}_i \sim p_\theta(\mathbf{e} \mid \widetilde{\mathbf{z}}_i)$. We construct extra preference pairs $(\widetilde{\mathbf{e}}^+, \widetilde{\mathbf{e}}^-)$ and train the preference ranker $s_\psi(\mathbf{e})$ by:

$$\mathcal{L}_{\text{rank}}(\psi) = -\mathbb{E}\big[\log \sigma(s_\psi(\mathbf{e}^+) - s_\psi(\mathbf{e}^-))\big] - \eta\, \mathbb{E}\big[\log \sigma(s_\psi(\widetilde{\mathbf{e}}^+) - s_\psi(\widetilde{\mathbf{e}}^-))\big], \tag{6}$$

where η controls the weight of preference-disentangled synthetic pairs.

Risk-control motivation under distribution closeness. Let P_y denote the true embedding distribution conditioned on preference label $y \in \{0, 1\}$, and $\hat{P}_y$ denote the class-conditional distribution induced by our disentangled generator. Consider preference ranking with a pairwise loss $\ell_\psi(\mathbf{e}^+, \mathbf{e}^-)$ (e.g., ranking loss), and let P (resp. $\hat{P}$) denote the induced pair distribution obtained by sampling $\mathbf{e}^+ \sim P_1, \mathbf{e}^- \sim P_0$ ($\mathbf{e}^+ \sim \hat{P}_1, \mathbf{e}^- \sim \hat{P}_0$, respectively). We train on the mixture $P_{\text{mix}} = (1-\eta)P + \eta\hat{P}$. We denote $\text{MMD}_k^{\text{pair}}(P, \hat{P})$ for the MMD measured over induced preference-pair distributions, distinguishing it from the latent-space MMD regularizer used in PREVAIL-MMD. The following result should be read as an RKHS surrogate analysis rather than a strict characterization of the MLP ranker used in our experiments. It does not claim that synthetic data always improves ranking. Instead, it motivates why keeping the generated pair distribution close to the true pair distribution under $\text{MMD}_k^{\text{pair}}$ can control the bias introduced by synthetic generation.

Proposition 1 (Risk transfer for an RKHS-bounded surrogate)**.** *For analytical clarity, consider a surrogate pairwise ranking loss ℓ_ψ that belongs to an RKHS ball induced by kernel k with $\|\ell_\psi\|_{\mathcal{H}_k} \le C$. Then the population risks satisfy:*

$$\big|\, \mathbb{E}_{(\mathbf{e}^+,\mathbf{e}^-)\sim P}\big[\ell_\psi(\mathbf{e}^+, \mathbf{e}^-)\big] - \mathbb{E}_{(\mathbf{e}^+,\mathbf{e}^-)\sim P_{\text{mix}}}\big[\ell_\psi(\mathbf{e}^+, \mathbf{e}^-)\big] \,\big| \;\le\; \eta\, C \cdot \text{MMD}_k^{\text{pair}}(P, \hat{P}). \tag{7}$$

In particular, if the generator is trained so that $\hat{P}_y$ is close to P_y (hence P close to $\hat{P}$) under $\text{MMD}_k^{\text{pair}}$, then training preference ranking with $\hat{P}$ introduces a controlled bias at the level of this RKHS surrogate. The result is therefore used only as a qualitative motivation for distribution-closeness regularization. The proof is given in App. F, and a bound-level condition for useful synthesis is discussed in App. F.1.

The proof follows from the standard RKHS property of MMD, i.e., $|\mathbb{E}_P f - \mathbb{E}_Q f| \le \|f\|_{\mathcal{H}_k} \text{MMD}_k(P, Q)$.

4 Experiments

In this section, we evaluate PREVAIL on preference ranking and downstream response selection. Due to space limitations, the statistics of the data set are provided in App. B.1. Details of embedding extraction and the evaluation setup are given in App. B.2. The VAE training configs are described in App. B.3, the reward model training configs are provided in App. B.4, and the downstream SFT evaluating configs are provided in App. B.5.

4.1 Does PREVAIL Produce a Preference-Separable Latent Geometry?

We first evaluate whether the proposed disentanglement mechanism produces the intended latent geometry, rather than only reporting downstream ranking scores. For each preference pair, we extract the LLM embeddings of the preferred and rejected responses and map them into the latent space using the trained encoder. We use the posterior mean as the latent representation and project all samples into two dimensions with t-SNE, using different colors for the two preference labels. A clearer separation indicates that the latent space provides a more useful preference boundary for ranking, while a compact class structure suggests that latent perturbation is less likely to produce off-distribution synthetic embeddings. As shown in Fig. 3, PREVAIL-MI and PREVAIL-MMD produce more distinguishable preferred and rejected regions than the original embedding space and LENS.

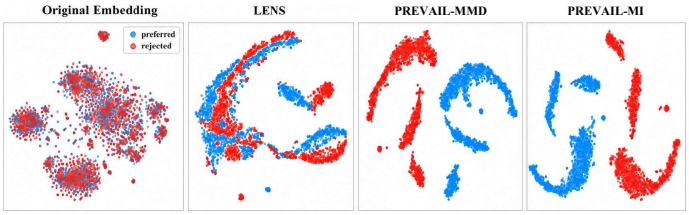


Figure 3: t-SNE visualization of preferred and rejected response embedding latents. Blue points indicate preferred responses, and red points indicate rejected responses. PREVAIL-MI and PREVAIL-MMD show clearer separation than the original embedding space and LENS, suggesting a more distinct preference boundary for ranking.

Table 1: Best-of-16 gold reward (BoN-16) on HH-RLHF and TL;DR. Textual-synthesis and latent-synthesis baseline results are sourced from Tao et al. [24]. Our results (PREVAIL-MMD, PREVAIL-MI) are reported as mean±std over 3 random seeds under the Skywork BoN-16 protocol. Higher is better; **bold** and $\star$ denote first place, † second place, and ‡ third place.

	HH-RLHF				TL;DR			
Method	Orig.	2×	4×	8×	Orig.	2×	4×	8×
Textual Synthesis								
Fully Fine-tune	1.49	1.57	1.78	1.93	0.69	0.84	0.97	1.23
Low-Rank Adaptation	1.28	1.48	1.52	1.61	0.57	0.87	0.92	1.15
Embedding MLP	1.43	1.51	1.62	1.73	0.78	0.94	1.02	1.11
Self-Rewarding [34]	1.49	1.48	1.59	1.77	0.69	0.78	0.92	0.95
Self-Evolved [17]	1.49	1.42	1.54	1.63	0.69	0.72	0.79	0.75
IPO [35]	1.49	1.37	1.25	1.32	0.69	0.63	0.67	0.62
Latent Space Synthesis								
Direct Perturbation [24]	1.43	1.46	1.32	1.46	0.78	0.81	0.84	0.79
Gaussian Sampling [24]	1.43	1.23	1.12	0.94	0.78	0.64	0.53	0.43
LENS [24]	1.43	1.86‡	1.94‡	2.20‡	0.78	1.25‡	1.44‡	1.48‡
PREVAIL-MI (ours)	1.32±0.07	**1.92**±0.11*	**2.11**±0.09*	**2.37**±0.05*	0.82±0.12	1.29±0.10†	**1.87**±0.09*	**1.92**±0.06*
PREVAIL-MMD (ours)	1.32±0.14	1.90±0.12†	2.07±0.10†	2.27±0.06†	0.82±0.08	**1.31**±0.09*	1.77±0.11†	1.89±0.07†

4.2 Why Does Preference-Conditioned VAE Design Matter?

We next analyze the design mechanism behind PREVAIL. The goal is to verify whether the gains come from preference-conditioned latent modeling and the MI/MMD objectives, rather than from simply adding a larger or different VAE architecture. Different from LENS [24], which relies on separately trained VAEs to model preferred and rejected response embeddings, PREVAIL uses a single class-conditional VAE under a unified cELBO objective. The preference-specific latent heads are not introduced as an additional separation trick; instead, they implement the class-conditional posterior $q_\phi(\mathbf{z} \mid \mathbf{e}, y)$ derived from the cELBO. Thus, preferred and rejected embeddings share the same generative model, while PREVAIL-MI uses a discriminator-based class-separation surrogate and PREVAIL-MMD uses a global MMD constraint together with per-sample information pressure. The label-conditioned architecture is therefore part of the variational formulation, while the final latent separation is produced by its interaction with these training objectives. This formulation makes the preference-dependent entanglement in latent space explicit and enables label-aware modeling of preferred and rejected embeddings under a unified variational framework. Concretely, PREVAIL models the prompt–response embedding distribution as a two-component preference mixture and uses the observed binary preference labels during training, i.e., $q_\phi(y \mid \mathbf{e}) = \delta_{y=y_i}$. Accordingly, our implementation does not treat preferred and rejected samples as implicitly mixed in a single latent space. Instead, we adopt a SHAREDVAE architecture with a shared encoder and preference-specific latent heads, such that chosen and rejected embeddings are mapped into separate latent variables before being decoded back to the embedding space. This design is consistent with our class-conditional formulation, where positive and negative embeddings are modeled through separate posteriors and then used for synthetic pairs.

Under this framework, we implement two variants of PREVAIL. PREVAIL-MMD is a statistical variant based on the MMD-regularized objective in Eq. (5), while PREVAIL-MI is a deep variant based on the discriminator-regularized objective in Eq. (4). Full architectural and optimization details are provided in App. B.3. We further control for this backbone factor in App. B.6, showing that the class-conditional posterior parameterization alone does not account for the gains of PREVAIL.

Table 2: Downstream validation via rejection-sampling-based SFT. Avg. Reward is computed by SKYWORK-REWARD-LLAMA-3.1-8B-V0.2 on 1,000 held-out test prompts. GPT-4 Win Rate reports the fraction of prompts where GPT-4 prefers the output of the given SFT model in pairwise evaluation; higher values indicate stronger downstream response quality. Higher is better for both metrics.

	HH-RLHF		TL;DR	
Method	**Avg. Reward**	**GPT-4 Win Rate (%)**	**Avg. Reward**	**GPT-4 Win Rate (%)**
LENS [24]	1.87	64	1.36	67
PREVAIL-MI (ours)	2.07	70	1.74	73
PREVAIL-MMD (ours)	2.03	68	1.62	70

4.3 Preference Ranking Modeling

We follow the overall embedding synthesis pipeline in [24], but adapt it to our preferred/rejected disentangled PREVAIL. Starting from the embedding pairs $\{(\mathbf{e}_i^+, \mathbf{e}_i^-)\}_{i=1}^N$, we first encode preferred and rejected responses separately through the trained class-conditional generator. For each original preferred embedding $\mathbf{e}_i^+$ and rejected embedding $\mathbf{e}_i^-$, we obtain their corresponding latent variables $\mathbf{z}_i^+ \sim q_\phi(\mathbf{z} \mid \mathbf{e}_i^+, y{=}1)$ and $\mathbf{z}_i^- \sim q_\phi(\mathbf{z} \mid \mathbf{e}_i^-, y{=}0)$. We then generate synthetic latent variants by injecting Gaussian perturbations: $\widetilde{\mathbf{z}}_{i,j}^+ = \mathbf{z}_i^+ + \epsilon_{i,j}^+, \widetilde{\mathbf{z}}_{i,j}^- = \mathbf{z}_i^- + \epsilon_{i,j}^-$, where $\epsilon_{i,j}^\pm \sim \mathcal{N}(0, \sigma_{\text{noise}}^2 I)$ and j indexes multiple synthetic samples per original embedding. These perturbed latent codes are decoded back into the embedding space as: $\widetilde{\mathbf{e}}_{i,j}^+ = p_\theta(\mathbf{e} \mid \widetilde{\mathbf{z}}_{i,j}^+), \widetilde{\mathbf{e}}_{i,j}^- = p_\theta(\mathbf{e} \mid \widetilde{\mathbf{z}}_{i,j}^-)$.

To construct generated preference embeddings, we preserve the prompt-conditioned pairing structure instead of forming a Cartesian product across preference classes. In particular, for each original prompt index i, the generated preferred and rejected embeddings are paired only with their counterpart generated from the same prompt: $\mathcal{E}_{\text{train}} = \{(\mathbf{e}_i^+, \mathbf{e}_i^-)\}_{i=1}^N \cup \{(\widetilde{\mathbf{e}}_{i,j}^+, \widetilde{\mathbf{e}}_{i,j}^-) \mid i \in [N],\ j \in [K]\}$. Here K is the number of synthetic samples per original pair. This construction keeps each Bradley–Terry comparison conditioned on the same prompt, while adding synthetic preferred/rejected embedding pairs for reward-model training. Evaluation follows the Best-of-16 protocol with SKYWORK-REWARD-LLAMA-3.1-8B as the reward model. Tab. 1 presents the full comparison. Specifically, we compare against the methods including textual-space synthesis approaches (Fully Fine-tune, Low-Rank Adaptation, Embedding MLP, Self-Rewarding [34], Self-Evolved [17], and IPO [35]) and latent-space baselines (Direct Perturbation, Gaussian Sampling, and the original LENS with contrastive VAE). On **HH-RLHF**, the two PREVAIL variants occupy the top two positions at every generation ratio. PREVAIL-MI achieves the best score at $2\times$ (1.92), $4\times$ (2.11), and $8\times$ (2.37), while PREVAIL-MMD ranks second in all three settings. The scores increase with the generation ratio, showing that stronger generation is beneficial on this dataset. On **TL;DR**, PREVAIL also remains strongest overall. PREVAIL-MMD is slightly better at $2\times$ (1.31 vs. 1.29), while PREVAIL-MI becomes best at $4\times$ (1.87) and $8\times$ (1.92). Overall, PREVAIL-MI achieves the best results in five of the six generation settings, suggesting that the MI-based estimator provides more useful synthetic pairs for ranking.

The gains widen with the generation ratio because higher ratios amplify the effect of synthetic-pair quality. At $2\times$, original pairs still form half of the training set and bound the influence of the synthesis method, whereas at $8\times$ synthetic pairs account for $7/8$ of it, so synthesis quality dominates the learned boundary. Prop. 1 gives the mechanism: the surrogate risk shift is bounded by $\eta C \, \mathrm{MMD}_k^{\mathrm{pair}}(P, \hat{P})$, so distribution mismatch becomes more consequential as synthetic pairs contribute more. Tab. 1 shows both directions on TL;DR: Gaussian Sampling degrades from 0.64 to 0.53 and 0.43 as the ratio grows, while PREVAIL-MI rises from 1.29 to 1.87 and 1.92, and its margin over LENS grows from $+0.04$ at $2\times$ to $+0.43$ and $+0.44$ at $4\times$ and $8\times$. The LENS scores are those reported by its authors under the same datasets, ratios, and Best-of-16 protocol, at their selected KL weight $\beta = 1.0$ and separation weight $\gamma_W = 0.1$ of LENS; we do not re-tune LENS.

4.4 Downstream Validation of the Learned Reward Model

To further verify the effectiveness of our method in downstream tasks and examine whether the learned reward model can be naturally incorporated into a standard preference-guided adaptation pipeline, we adopt rejection sampling based [36] supervised fine-tuning (SFT) for downstream validation. We use

Table 3: Generalization across embedding backbone models. All methods use 4× generation on HH-RLHF and are evaluated with the Skywork BoN-16 protocol.

Backbone	PREVAIL-MI	PREVAIL-MMD
GEMMA-2B	0.86	0.72
LLAMA-3.2-3B	0.91	0.87
MISTRAL-7B	2.10	1.86
QWEN-2.5-7B	2.05	1.97
LLAMA-3.1-8B	2.14	2.11
QWEN-2.5-14B	2.15	2.08

Table 4: Sensitivity analysis for regularization coefficients, synthesis noise, and sample number. ♣ and ♢ denote PREVAIL-MI and PREVAIL-MMD, respectively. Bold marks the best value within each variant block.

	Noise Level σ^2_{noise}				Sample Number K			
Coeff. Setting	**0.001**	**0.01**	**0.1**	**1.0**	**100**	**200**	**1,000**	**10,000**
$(\beta = 0.1, \lambda = 1)^{\clubsuit}$	**2.03**	2.09	1.99	1.90	1.65	1.72	2.09	2.10
$(\beta = 1, \lambda = 0.1)^{\clubsuit}$	1.89	**2.10**	**2.04**	**2.06**	**1.79**	**1.92**	**2.10**	**2.11**
$(\beta = 1, \lambda = 10)^{\clubsuit}$	2.01	2.07	2.03	1.91	1.71	1.86	2.07	2.10
$(\alpha = 0.1, \gamma = 1)^{\diamondsuit}$	1.82	1.92	1.87	1.80	1.69	1.79	1.92	2.01
$(\alpha = 0.1, \gamma = 0.1)^{\diamondsuit}$	1.87	2.01	**1.93**	1.82	1.72	1.82	2.01	2.05
$(\alpha = 0.1, \gamma = 10)^{\diamondsuit}$	**1.90**	**2.04**	1.92	**1.90**	**1.77**	**1.85**	**2.04**	**2.07**

rejection sampling instead of PPO or DPO, since PPO introduces substantially higher computational cost and optimization variance, while DPO focuses on direct policy optimization from explicit chosen–rejected pairs rather than evaluating the quality of reward-selected supervision targets. Concretely, for each prompt, we sample multiple candidate responses from LLAMA-3.1-8B-INSTRUCT, rank them with the trained reward model, and retain the highest-scoring response as the target to construct the SFT dataset. We evaluate the resulting SFT models on held-out test prompts using the average reward from SKYWORK-REWARD-LLAMA-3.1-8B-V0.2 and GPT-4 pairwise win rate; full experimental details are provided in App. B.5. Tab. 2 summarizes the results. PREVAIL-MI achieves the best Avg. Reward and GPT-4 Win Rate on both datasets, with the largest reward gain appearing on TL;DR ($+0.38$ over LENS) and the same $+6\%$ win-rate gain on both datasets. PREVAIL-MMD also outperforms LENS on both datasets, with a larger gain on TL;DR (1.62 vs. 1.36) than on HH-RLHF (2.03 vs. 1.87). The trend shows disentangled reward modeling translates into stronger downstream policies.

4.5 General Implication: Robustness Across Embedding Backbones

To examine whether preference-disentangled generation is tied to a specific embedding encoder or represents a more general mechanism, we evaluate PREVAIL across multiple backbone models. We report that the method generalizes across several open-source model families, including GEMMA-2B, LLAMA-3.2-3B, MISTRAL-7B, QWEN-2.5-7B, and LLAMA-3.1-8B, all evaluated under the 4× generation setting. The results in Tab. 3 show that both PREVAIL variants remain effective across embedding backbones, and PREVAIL-MI is consistently higher than PREVAIL-MMD under the same reward-model training and evaluation protocol. This suggests that the benefit of our method does not come from a backbone-specific artifact, but from the generation mechanism itself. At the same time, the gap between PREVAIL-MI and PREVAIL-MMD varies across backbones, suggesting that the benefit of each disentanglement estimator depends on the representation space while still preserving the same overall ordering. Tab. 3 reports the BoN-16 scores under 4× generation across backbone models. PREVAIL-MI is higher than PREVAIL-MMD for every backbone, with the best score reaching 2.15 on QWEN-2.5-14B. Both variants improve substantially once the backbone reaches the 7B scale, while PREVAIL-MI keeps a consistent advantage across all tested model families.

4.6 Sensitivity Analysis

We conduct sensitivity analysis from three aspects: the coefficient combinations controlling class-separation surrogates, within-class information, and aggregate matching; the number of original preference samples used for reward modeling; and the latent noise level in Tab. 4.

Sensitivity to regularization coefficients. We first study the effect of the key regularization coefficients in our preferred disentangled VAE training. For PREVAIL-MI, we vary the instance-level prior coefficient β and the discriminator-based disentanglement coefficient λ in Eq. (4). For PREVAIL-MMD, we vary the *aggregated distribution matching* coefficient α and the instance-level KL coefficient γ in Eq. (5). These coefficients control the trade-off among *reconstruction* fidelity, aggregate matching, per-sample information pressure, and class-separation signals. This analysis examines the same three-way trade-off and identifies coefficient combinations that yield

stable synthetic preference generation for downstream reward modeling. Across this grid, the best PREVAIL-MI score is higher than the best PREVAIL-MMD score in each column. The setting $\beta = 1, \lambda = 0.1$ performs best for PREVAIL-MI in most columns and reaches 2.11, while $\beta = 0.1, \lambda = 1$ is best only at $\sigma^2_{\text{noise}} = 0.001$. For PREVAIL-MMD, $\alpha = 0.1, \gamma = 10$ is strongest in most columns, but $\alpha = 0.1, \gamma = 0.1$ gives the best MMD score at $\sigma^2_{\text{noise}} = 0.1$ (1.93). Overall, the MMD scores range from 1.69 to 2.07.

Sensitivity to sample number. We next examine how the effectiveness of latent-space generation varies with the number of original preference samples. Specifically, we subsample K original preference pairs from the training split and train the reward model with and without latent generation under the same protocol. This setting is intended to evaluate whether the proposed generation strategy is mainly useful when only a small number of labeled preference pairs are available. By controlling K, this analysis also allows us to assess the robustness of the learned generator and reward model across different levels of supervision. All configurations improve monotonically as K grows from 100 to 10,000. PREVAIL-MI with $\beta = 1, \lambda = 0.1$ scales best, rising from 1.79 to 2.11, while the strongest PREVAIL-MMD setting reaches 2.07. This indicates that PREVAIL-MI benefits more from additional preference supervision, although PREVAIL-MMD also improves steadily with more samples.

Sensitivity to synthesis noise level. Finally, we study the effect of the latent perturbation level used during synthetic-sample generation. Recall that our generation pipeline perturbs the latent code as $\widetilde{\mathbf{z}} = \mathbf{z} + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2_{\text{noise}} I)$. We vary σ^2_{noise} to examine how the magnitude of latent perturbation influences the quality of decoded synthetic embeddings and the resulting reward-model performance. This analysis is motivated by the trade-off between diversity and preference preservation: if the noise level is too small, the generated samples may remain too close to the originals and provide limited generation benefit; if it is too large, the decoded embeddings may drift away from the underlying preference structure and introduce noisy supervision. Therefore, this experiment is used to identify an appropriate noise range for stable and effective latent-space synthesis. For all PREVAIL-MI variants, performance peaks around $\sigma^2_{\text{noise}} \in [0.01, 0.1]$ (best 2.10 at 0.01) and is lower when the noise becomes too small or too large, confirming the diversity–fidelity trade-off. PREVAIL-MMD shows a flatter response across noise levels, with row-wise ranges up to 0.19 across the four settings, suggesting it is less sensitive to the perturbation magnitude.

5 Conclusion

This paper proposes PREVAIL, a generative preference-ranking framework that models prompt–response embeddings with preference-aware latent generation. At the core of PREVAIL is cELBO, a unified variational framework for learning generative embedding ranking models. By decomposing a class-conditional ELBO, PREVAIL separates *reconstruction*, *within-class information*, and *class-wise aggregate matching*, which motivates two practical variants, PREVAIL-MI and PREVAIL-MMD. Experiments show that PREVAIL generates more useful synthetic preference pairs and improves reward-model performance across datasets, generation settings, and embedding backbones. These results highlight the value of disentangled embedding generation for efficient generative preference ranking.

Limitations. PREVAIL assumes clean binary preference labels, the standard setting shared by all compared methods, and its robustness to noisy annotations is not evaluated here. Both separation surrogates in Eqs. (4) and (5) act on aggregated posteriors $\bar{q}_\phi^{(y)}$ rather than on individual pairs, which we expect to dampen sparse label noise, and the cELBO admits soft labels in place of the hard assignment $q_\phi(y \mid \mathbf{e}_i) = \delta_{y=y_i}$; both remain untested here. Extending PREVAIL beyond response ranking to broader settings such as LLM agent tasks is a further direction.

Acknowledgements and Disclosure of Funding

The authors received no third-party funding for this work and declare no competing interests.

References

[1] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, 2017.

[2] 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. In *Advances in Neural Information Processing Systems*, 2020.

[3] 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. In *Advances in Neural Information Processing Systems*, 2022.

[4] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's Verify Step by Step. In *International Conference on Learning Representations*, 2023.

[5] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E Gonzalez, et al. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. *International Conference on Machine Learning*, 2024.

[6] Abhijnan Nath, Andrey Volozin, Saumajit Saha, Albert Aristotle Nanda, Galina Grunin, Rahul Bhotika, and Nikhil Krishnaswamy. DPL: Diverse Preference Learning without a Reference Model. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025.

[7] Nitay Calderon, Liat Ein Dor, and Roi Reichart. Multi-Domain Explainability of Preferences. In *Conference on Empirical Methods in Natural Language Processing*, 2025.

[8] Feng Luo, Rui Yang, Hao Sun, Chunyuan Deng, Jiarui Yao, Jingyan Shen, Huan Zhang, and Hanjie Chen. Rethinking Diverse Human Preference Learning through Principal Component Analysis. In *Findings of the Association for Computational Linguistics*, 2025.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[10] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Advances in Neural Information Processing Systems*, 2023.

[11] Ning Yang, Hai Lin, Yibo Liu, Baoliang Tian, Guoqing Liu, and Haijun Zhang. Token-Importance Guided Direct Preference Optimization. In *International Conference on Learning Representations*, 2026.

[12] Philipp Becker, Niklas Freymuth, Serge Thilges, Fabian Otto, and Gerhard Neumann. TROLL: Trust Regions Improve Reinforcement Learning for Large Language Models. In *International Conference on Learning Representations*, 2026.

[13] Hao Sun, Yunyi Shen, and Jean-Francois Ton. Rethinking Reward Modeling in Preference-Based Large Language Model Alignment. In *International Conference on Learning Representations*, 2025.

[14] Kexin Huang, Junkang Wu, Ziqian Chen, Xue Wang, Jinyang Gao, Bolin Ding, Jiancan Wu, Xiangnan He, and Xiang Wang. Larger or Smaller Reward Margins to Select Preferences for LLM Alignment? In *International Conference on Machine Learning*, 2025.

[15] Yue Yu, Zhengxing Chen, Aston Zhang, Liang Tan, Chenguang Zhu, Richard Yuanzhe Pang, Yundi Qian, Xuewei Wang, Suchin Gururangan, Chao Zhang, et al. Self-Generated Critiques Boost Reward Modeling for Language Models. In *Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2025.

[16] Zongqi Wang, Rui Wang, Yuchuan Wu, Yiyao Yu, Pinyi Zhang, Shaoning Sun, Yujiu Yang, and Yongbin Li. Reward Modeling from Natural Language Human Feedback. *International Conference on Machine Learning*, 2026.

[17] Chenghua Huang, Zhizhen Fan, Lu Wang, Fangkai Yang, Pu Zhao, Zeqi Lin, Qingwei Lin, Dongmei Zhang, Saravan Rajmohan, and Qi Zhang. Self-evolved Reward Learning for LLMs. In *International Conference on Learning Representations*, 2025.

[18] Xiaobo Liang, Haoke Zhang, Juntao Li, Kehai Chen, Qiaoming Zhu, and Min Zhang. Generative Reward Modeling via Synthetic Criteria Preference Learning. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

[19] Taiwei Shi, Kai Chen, and Jieyu Zhao. Safer-Instruct: Aligning Language Models with Automated Preference Data. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

[20] Zhaoxuan Tan, Zheng Li, Tianyi Liu, Haodong Wang, Hyokun Yun, Ming Zeng, Pei Chen, Zhihan Zhang, Yifan Gao, Ruijie Wang, et al. Aligning Large Language Models with Implicit Preferences from User-Generated Content. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

[21] Anand Siththaranjan, Cassidy Laidlaw, and Dylan Hadfield-Menell. Distributional Preference Learning: Understanding and Accounting for Hidden Context in RLHF. *Advances in Neural Information Processing Systems*, 2023.

[22] Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing Reinforcement Learning from Human Feedback with Variational Preference Learning. *Advances in Neural Information Processing Systems*, 2024.

[23] Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward Reasoning Models. In *Advances in Neural Information Processing System*, 2025.

[24] Leitian Tao, Xuefeng Du, and Sharon Li. Limited Preference Data? Learning Better Reward Model with Latent Space Synthesis. In *Advances in Neural Information Processing Systems*, 2025.

[25] Gihoon Kim and Euntai Kim. Swap-Guided Preference Learning for Personalized Reinforcement Learning from Human Feedback. *International Conference on Learning Representations*, 2026.

[26] Yunyi Shen, Hao Sun, and Jean-Francois Ton. Active Reward Modeling: Adaptive Preference Labeling for Large Language Model Alignment. In *International Conference on Machine Learning*, 2025.

[27] Xun Deng, Han Zhong, Rui Ai, Fuli Feng, Zheng Wang, and Xiangnan He. Less is More: Improving LLM Alignment via Preference Data Selection. *Advances in Neural Information Processing Systems*, 2025.

[28] Mike Wu and Noah Goodman. Multimodal Generative Models for Scalable Weakly-Supervised Learning. *Advances in neural information processing systems*, 2018.

[29] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem Meent. Structured Disentangled Representations. In *International Conference on Artificial Intelligence and Statistics*, 2019.

[30] Hyunjik Kim and Andriy Mnih. Disentangling by Factorising. In *International Conference on Machine Learning*, 2018.

[31] Pan Wang, Qiang Zhou, Yawen Wu, Tianlong Chen, and Jingtong Hu. DLF: Disentangled-Language-Focused Multimodal Sentiment Analysis. In *AAAI Conference on Artificial Intelligence*, 2025.

[32] Shengjia Zhao, Jiaming Song, and Stefano Ermon. InfoVAE: Balancing Learning and Inference in Variational Autoencoders. In *AAAI Conference on Artificial Intelligence*, 2019.

[33] Yingheng Wang, Yair Schiff, Aaron Gokaslan, Weishen Pan, Fei Wang, Christopher De Sa, and Volodymyr Kuleshov. InfoDiffusion: Representation Learning using Information Maximizing Diffusion Models. In *International Conference on Machine Learning*, 2023.

[34] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-Rewarding Language Models. In *International Conference on Machine Learning*, 2024.

[35] Shivank Garg, Ayush Singh, Shweta Singh, and Paras Chopra. IPO: Your Language Model is Secretly a Preference Classifier. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

[36] Leo Gao, John Schulman, and Jacob Hilton. Scaling Laws for Reward Model Overoptimization. In *International Conference on Machine Learning*, 2023.

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

[38] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback. *arXiv preprint arXiv:2309.00267*, 2023.

[39] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. In *International Conference on Learning Representations*, 2017.

[40] Zhangkai Wu and Longbing Cao. C^2VAE: Gaussian copula-based VAE differing disentangled from coupled representations with contrastive posterior. *arXiv preprint arXiv:2309.13303*, 2023.

[41] Chenrui Ma, Xi Xiao, Tianyang Wang, Xiao Wang, and Yanning Shen. CAD-VAE: Leveraging Correlation-Aware Latents for Comprehensive Fair Disentanglement. In *AAAI Conference on Artificial Intelligence*, 2026.

[42] Zhangkai Wu, Xuhui Fan, Jin Li, Zhilin Zhao, Hui Chen, and Longbing Cao. SepDiff: Self-encoding parameter diffusion for learning latent semantics. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3273–3284, 2025. doi: 10.1145/3711896.3737125.

[43] Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to Learn Automatic Summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, 2017.

[44] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[45] Zhangkai Wu, Longbing Cao, and Lei Qi. eVAE: Evolutionary variational autoencoder. *IEEE Transactions on Neural Networks and Learning Systems*, 2024. doi: 10.1109/TNNLS.2024.3359275.

[46] Zhangkai Wu, Xuhui Fan, and Longbing Cao. ProgDiffusion: Progressively Self-encoding Diffusion Models. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2025.

Appendix Contents

- A. Preference Ranking Learning
  - A.1 Deterministic Preference Ranking
  - A.2 Generative Preference Ranking
- B. Experimental Details
  - B.1 Dataset Statistics
  - B.2 Embedding Extraction and Evaluation Setup
    - B.2.1 Training Embeddings
    - B.2.2 Multi-response Evaluation Set
    - B.2.3 Gold Reward Scoring
    - B.2.4 Best-of-16 Evaluation
  - B.3 VAE Training Configs
  - B.4 Preference Ranking Model Training Configs
  - B.5 Experimental Settings for Rejection-Sampling-Based SFT
  - B.6 Architecture-Controlled Ablation
- C. Decomposed class-conditional ELBO
- D. Disentangled prompt-response embedding cELBO in Deep Manner
  - D.1 Mixture-posterior Surrogate
  - D.2 Density-ratio Estimation
  - D.3 Practical Deep Objective
- E. Disentangled prompt-response embedding cELBO in Statistical Manner
- F. RKHS Surrogate Analysis for Preference Ranking under Disentangled Generation
  - F.1 Bound-Level Condition for Useful Synthesis
  - F.2 Squared MMD Convention
- G. Qualitative Examples

A Preference Ranking Learning

We organize prior preference-ranking studies by the distributional object they learn. Given a prompt $\mathbf{x}$, a response $\mathbf{y}$, a prompt–response pair $\mathbf{u} = (\mathbf{x}, \mathbf{y})$, and a binary preference label y, deterministic methods focus on the conditional ranking rule $p(y \mid \mathbf{u})$, while generative methods model the joint structure $p(\mathbf{u}, y)$. his distinction separates methods that only refine the decision boundary from methods that also model the data distribution for preference-consistent synthesis.

A.1 Deterministic Preference Ranking

Many recent studies improve preference ranking by strengthening the conditional decision boundary $p(y \mid \mathbf{u})$, where $\mathbf{u} = (\mathbf{x}, \mathbf{y})$ denotes the prompt–response pair and y denotes the preference label. RLAIF reduces the dependence on costly human preference labels by using LLMs to generate AI feedback for preference learning [37, 38]. Critic-RM enriches preference prediction with self-generated critiques rather than a single scalar score [15]. RM-NLHF introduces natural-language feedback as process-level supervision for preference ranking [16]. SER expands limited preference data by self-labeling unlabeled examples and selecting high-confidence comparisons for iterative preference-ranker training [17]. Criteria-based preference modeling further decomposes preference judgment into generated criteria, rubrics, and decisions [18]. Although these methods may generate critiques, feedback, or criteria, their main learning target remains the conditional preference rule $p(y \mid \mathbf{u})$. They directly exploit supervised preference signals and are easy to integrate into downstream preference-ranking or policy-optimization pipelines, but they usually treat the representation distribution of prompt–response pairs as fixed.

A.2 Generative Preference Ranking

Generative preference ranking instead models the joint distribution $p(\mathbf{u}, y) = p(y \mid \mathbf{u})p(\mathbf{u})$, aiming to capture both the preference boundary and the underlying data distribution. With latent-variable assumptions, this objective can be optimized through a variational surrogate, allowing the model to learn not only $p(y \mid \mathbf{u})$ but also the marginal structure $p(\mathbf{u})$ that supports preference-consistent sample generation. Recent work improves generative preference modeling by expanding the sources of preference uncertainty, including hidden contexts behind preference judgments [21], personalized latent preferences across users [22, 25], and reasoning-based reward estimation [23]. These probabilistic designs make the preference boundary more expressive, but modeling and sampling in the textual space remain high-dimensional and computationally expensive. LENS [24] reduces this cost by moving generative modeling to the prompt–response embedding space. However, the preferred and rejected embedding data distributions can still be entangled, leaving the generated preference pairs ambiguous. Our method continues this generative view, but focuses on explicitly disentangling preferred and rejected embedding distributions through a class-conditional variational objective, so that the synthesized preference pairs better support preference-ranker training.

A.3 Positioning against Disentangled VAEs

Disentangled VAEs define their disentanglement term over latent dimensions, driven by no label at all, as the β-weighted KL in β-VAE [39] and the total correlation in FactorVAE [30] and C^2VAE [40], or, when labels are available, used to isolate a sensitive attribute in the representation, as in CAD-VAE [41]. Self-encoding diffusion models likewise learn disentangled latent semantics without labels, as in SepDiff [42]. Generative preference modeling instead needs the two label-conditional distributions $q_\phi(\mathbf{z} \mid \mathbf{e}, y{=}1)$ and $q_\phi(\mathbf{z} \mid \mathbf{e}, y{=}0)$ to be separated, so that latent perturbation does not cross the preference boundary and produce ambiguous pairs. The two goals are not equivalent: a dimension-wise penalty can be satisfied while the preferred and rejected components still overlap. PREVAIL closes this gap by estimating the separation term directly from the observed preference labels, as $I(Y; \mathbf{z})$ through a label discriminator in PREVAIL-MI or as an MMD between class-conditioned aggregated posteriors in PREVAIL-MMD. Tab. 5 summarizes the comparison along the surrogate objective and the source of the disentanglement signal.

Table 5: Positioning of PREVAIL against disentangled VAEs.

Method	Surrogate objective	Disentanglement comes from
β-VAE [39]	ELBO	$\beta\, D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z}))$
FactorVAE [30]	ELBO + TC	$D_{\mathrm{KL}}(q(\mathbf{z}) \,\|\, \prod_j q(z_j))$, density-ratio trick
CAD-VAE [41]	ELBO + CMI	$I(\mathbf{z}_t; \mathbf{z}_s \mid \mathbf{z}_c)$, auxiliary predictors
PREVAIL-MI	cELBO + preference MI	$I(Y; \mathbf{z})$, preference labels and density-ratio trick
PREVAIL-MMD	cELBO + aggregate matching	$\mathrm{MMD}^2(\bar{q}_\phi(\mathbf{z}), p(\mathbf{z}))$, class-conditioned posteriors

B Experimental Details

B.1 Dataset Statistics

We evaluate on two widely used human-preference datasets, following the experimental setup of [17, 34, 24].

TL;DR [3] is based on the Reddit TL;DR summarization corpus, where Reddit posts are paired with short author-provided summaries [43]. Building on this corpus, Stiennon et al. [2] further collected human preference comparisons between model-generated summaries, forming a summarization preference dataset for reward modeling and preference-based adaptation. Each example contains a Reddit post as the prompt, two candidate summaries, and a human preference label indicating which summary is better. We follow the experimental setting of Huang et al. [17]. The data are organized as comparison records, with fields such as prompt metadata, two candidate responses, and an annotator preference label. This processed version contains 178,944 rows in total, including 92,858 training examples and 86,086 validation examples.

HH-RLHF [4] is a large-scale dialogue preference dataset introduced by Anthropic [44] for training and evaluating helpful and harmless assistants. Each example consists of a multi-turn dialogue context, two candidate responses produced by language models, and a human preference label indicating the preferred response. The annotations were collected by asking human labelers to compare the two responses under criteria such as helpfulness and harmlessness. The public version of the dataset contains 169,352 samples. For both datasets, we use the official training splits to train reward models and evaluate on held-out test prompts. Dataset statistics are summarized in Table 6.

Table 6: Dataset statistics for the two benchmarks used in our experiments.

Dataset	# Train Pairs	Task Type	Response Type	Feedback Type
TL;DR	92,858	Summarization	Text summary	Human pairwise preference
HH-RLHF	161,000	Dialogue	Conversational response	Human pairwise preference

B.2 Embedding Extraction and Evaluation Setup

B.2.1 Training Embeddings

For each prompt $\mathbf{x}$ with a preferred response $\mathbf{y}^+$ and a rejected response $\mathbf{y}^-$ in the training set, we extract fixed embeddings using LLAMA-3.1-8B-INSTRUCT[5] as the backbone encoder similar to [17, 24]. Specifically, we concatenate the prompt with its corresponding response and perform a single forward pass through the model with `output_hidden_states=True`. The embedding of each text is taken as the hidden state of the *last token* at the *final transformer layer*, yielding a $4{,}096$-dimensional vector. Sequences are truncated to a maximum of 512 tokens. The resulting training corpus is stored as a matrix of shape $[N, 2, 4096]$, where the second dimension indexes the chosen prompt–response pair $(\mathbf{x}, \mathbf{y}^+)$ and the rejected prompt–response pair $(\mathbf{x}, \mathbf{y}^-)$ respectively.

For format consistency between training and evaluation, HH-RLHF inputs are formatted as ‘‘`\n\nHuman: {prompt}\n\nAssistant: {response}`’’, while TL;DR inputs are formatted as ‘‘`{prompt} {response}`’’.

[3] `https://huggingface.co/datasets/HuggingFaceH4/summarize-from-feedback`

[4] `https://huggingface.co/datasets/Anthropic/hh-rlhf`

[5] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

B.2.2 Multi-response Evaluation Set

To construct the evaluation set for Best-of-N scoring, we sample $3{,}000$ prompts from the held-out test split of each dataset. For each prompt, we generate 16 independent responses using LLAMA-3.1-8B-INSTRUCT with nucleus sampling (`do_sample=True`). Embeddings for all $3{,}000 \times 16$ responses are extracted using the same procedure described above, resulting in arrays of shape $[3000, 16, 4096]$.

B.2.3 Gold Reward Scoring

Each of the 16 responses per prompt is assigned a gold reward score using SKYWORK-REWARD-LLAMA-3.1-8B-V0.2 [6] as the oracle reward model. For HH-RLHF, the full multi-turn conversation is formatted using the model's chat template with interleaved `Human/Assistant` turns. For TL;DR, the prompt and summary are formatted as a single-turn exchange. The resulting reward scores are stored as a matrix of shape $[3000, 16]$ and serve as ground-truth labels for evaluating the trained reward model.

B.2.4 Best-of-16 Evaluation

At evaluation time, the trained reward model scores all 16 candidate responses for each prompt and selects the one with the highest predicted reward. The quality of this selection is measured by the gold reward assigned to the chosen response by SKYWORK-REWARD-LLAMA-3.1-8B-V0.2, averaged over all $3{,}000$ test prompts: BoN-16 $= \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} r^* \Big(\arg\max_{i \in [16]} \hat{r}(\mathbf{e}_{p,i}) \Big)$,, where $\mathcal{P}$ is the set of test prompts, $\mathbf{e}_{p,i}$ is the embedding of the i-th response to prompt p, $\hat{r}$ is the trained reward model, and r^* is the gold reward from SKYWORK.

B.3 VAE Training Configs

We train a embedding VAE on the selected preference embedding pairs to learn a structured latent space for data generation. The model adopts a *shared-encoder dual-head* architecture (SHAREDVAE): a single encoder maps the $4{,}096$-dimensional input embeddings to a hidden representation of size 256 via a linear layer followed by ReLU activation, after which two independent sets of projection heads (μ^+, σ^+) and (μ^-, σ^-) produce separate latent distributions of dimension 128 for chosen and rejected samples respectively. A shared decoder mirrors this structure with a linear layer ($128 \to 256$) with ReLU activation followed by a final linear projection ($256 \to 4{,}096$). The VAE is optimized with Adam (learning rate 10^{-4}) for 30 epochs with batch size 128, trained on $3{,}000$ randomly sampled preference pairs from the training set. The KL term is held at a target level (about 0.02 on HH-RLHF) throughout training, so the reported coefficients correspond to a controlled regularization regime rather than a bare grid search, in line with adaptive KL-weight schedules for VAE training [45]. All experiments are conducted on a single NVIDIA Tesla H100 GPU.

B.4 Preference Ranking Model Training Configs

We train a Bradley–Terry reward model as preference ranking model implemented as a lightweight MLP that takes a $4{,}096$-dimensional preference embedding as input and outputs a scalar reward score. The network consists of a single hidden layer of size 512 with ReLU activation followed by a linear output layer, giving an architecture of $4{,}096 \to 512 \to 1$. The model is trained with the Bradley–Terry pairwise ranking loss using the Adam optimizer (learning rate 10^{-4}) for 100 epochs with batch size 128.

For the baseline, the model is trained on the original $3{,}000$ preference pairs. For LENS, generated pairs expand the training set to $2\times$, $4\times$, or $8\times$ the original size using VAE-generated synthetic pairs, resulting in $6{,}000$, $12{,}000$, and $24{,}000$ training pairs respectively. All experiments are repeated over three random seeds (42, 123, 456), each independently sampling the $3{,}000$ training pairs from the full training corpus, and results are reported as mean $\pm$ standard deviation.

[6] `https://huggingface.co/Skywork/Skywork-Reward-Llama-3.1-8B-v0.2`

B.5 Experimental Settings for Rejection-Sampling-Based SFT

We follow a two-stage pipeline to evaluate whether improved reward modeling leads to better downstream policy adaptation. The compared reward models are those trained with the original LENS objective, **PREVAIL-MMD**, and **PREVAIL-MI**. Following Tao et al. [24], downstream validation is conducted through rejection-sampling-based supervised fine-tuning (SFT). In this subsection, we write $\mathbf{x}$ for a prompt and $\mathbf{y}$ for a response, while y remains reserved for the binary preference label in the reward-modeling sections.

To construct the downstream SFT dataset, we first subsample $1{,}000$ prompts from the training split. For each prompt $\mathbf{x}$, we use LLAMA-3.1-8B-INSTRUCT as the policy model to generate 16 candidate responses, denoted by $\mathcal{Y}(\mathbf{x}) = \{\mathbf{y}_1, \ldots, \mathbf{y}_{16}\}$. Each candidate prompt–response pair $(\mathbf{x}, \mathbf{y})$ is then converted into its final-layer embedding $\mathbf{e} = \mathrm{LLMembed}(\mathbf{x}, \mathbf{y})$ and scored by a trained reward model $\hat{r}(\mathbf{e})$. For each prompt, we retain the highest-scoring response as the supervision target, i.e., $\mathbf{y}^*(\mathbf{x}) = \arg\max_{\mathbf{y} \in \mathcal{Y}(\mathbf{x})} \hat{r}\big(\mathrm{LLMembed}(\mathbf{x}, \mathbf{y})\big)$. Applying this procedure separately to the reward models trained with LENS, PREVAIL-MMD, and PREVAIL-MI yields three reward-selected SFT datasets, denoted by $\mathcal{D}_{\text{LENS}}$, $\mathcal{D}_{\text{MMD}}$, and $\mathcal{D}_{\text{MI}}$, respectively. Since all three datasets are constructed from the same prompt subset and the same candidate pools, the only difference among them is the reward model used to select the target response.

We then train one policy model on each reward-selected dataset using the same initialization and optimization setup. All SFT runs start from the same LLAMA-3.1-8B-INSTRUCT checkpoint and are trained on prompt–response pairs $(\mathbf{x}, \mathbf{y}^*)$ with the standard autoregressive language-modeling objective under teacher forcing, i.e., $\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{|\mathbf{y}^*|} \log p_\theta(\mathbf{y}_t^* \mid \mathbf{x}, \mathbf{y}_{<t}^*)$. We perform full-parameter fine-tuning for all model parameters rather than parameter-efficient adaptation. The training configuration follows Tao et al. [24]: we train for 1 epoch with learning rate 1×10^{-5}, batch size 32, gradient accumulation step 1, and maximum sequence length 512. We employ DeepSpeed ZeRO Stage 2 as the training framework for memory-efficient distributed full fine-tuning.

After SFT, we evaluate the resulting models on a held-out set of $1{,}000$ test prompts sampled from the test split. For each test prompt, each SFT model generates one response under the same inference configuration. We report two metrics. The first is the average reward assigned by the held-out oracle reward model SKYWORK-REWARD-LLAMA-3.1-8B-V0.2, defined as $\mathrm{AvgReward} = \frac{1}{|\mathcal{P}_{\text{test}}|} \sum_{\mathbf{x} \in \mathcal{P}_{\text{test}}} r^*\big(\mathrm{LLMembed}(\mathbf{x}, \hat{\mathbf{y}}(\mathbf{x}))\big)$, where $\hat{\mathbf{y}}(\mathbf{x})$ denotes the response generated by the fine-tuned policy for prompt $\mathbf{x}$. The second is GPT-4 pairwise win rate, where outputs from two competing SFT models are compared prompt by prompt and judged by GPT-4. All compared SFT models use the same base policy, the same training and test prompt subsets, the same rejection-sampling procedure, and the same SFT hyperparameters.

B.6 Architecture-Controlled Ablation

Effect of cELBO-derived disentanglement. Since PREVAIL uses a single class-conditional VAE, we want to know whether its improvement can be explained by label conditioning alone. All variants are run with the same three random seeds (42, 123, 456) under identical data, architecture, augmentation ratio, and reward-model settings, and Tab. 7 reports mean$\pm$std. To isolate this factor, we compare PREVAIL with a class-conditional VAE that uses the same shared encoder, decoder, and label-conditioned latent heads, but removes the MI/MMD disentanglement regularizer. This ablation keeps the architecture fixed and only removes the distribution-level separation term. As shown in Tab. 7, the fully shared-head VAE achieves 0.83 on HH-RLHF under $4\times$ generation, while the class-conditional VAE without disentanglement obtains 0.80. Thus, label-conditioned latent heads alone do not account for the gain. Adding a Wasserstein separation term improves the score to 1.62, indicating the benefit of explicit distribution-level separation, but it still falls below PREVAIL-MMD (2.07) and PREVAIL-MI (2.11). These results show the main improvement comes from the cELBO-derived disentanglement regularization, rather than from the class-conditioned architecture itself. The gap between the undisentangled variants and the two PREVAIL objectives is far larger than the seed-level variation, so the ordering is not an artifact of single-run noise. Since the number of synthetic pairs is fixed at $4\times$ for every variant, the gains cannot be attributed to synthetic-data quantity either: neither class conditioning alone nor the presence of a separation term alone accounts for them.

Table 7: Architecture-controlled ablation on HH-RLHF under 4× generation. All variants use a single VAE framework and the same reward-model training protocol; mean±std over three seeds.

Variant	**Disentanglement Loss**	**BoN-16**
Single shared-head VAE	None	0.83±0.11
Single class-conditional VAE	None	0.80±0.09
Single class-conditional VAE + W_1	Wasserstein	1.62±0.12
PREVAIL-MMD	MMD	2.07±0.10
PREVAIL-MI	MI	2.11±0.09

C Decomposed class-conditional ELBO

We provide the derivation of the decomposed form of the class-conditional ELBO used in the main text. For a mini-batch $\{(\mathbf{e}_i, y_i)\}_{i=1}^B$, define the (soft) class count $B_y := \sum_{i=1}^B q_\phi(y \mid \mathbf{e}_i)$ and the class-conditional aggregated posterior $\bar{q}_\phi^{(y)}(\mathbf{z}) := \frac{1}{B_y} \sum_{i=1}^B q_\phi(y \mid \mathbf{e}_i)\, q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)$.[7] The key step is adding and subtracting $\log \bar{q}_\phi^{(y)}(\mathbf{z})$ to split the KL term, and then grouping terms by y.[8]

[7] Here $\bar{q}_\phi^{(y)}(\mathbf{z})$ is an *aggregated (empirical) posterior* induced by the mini-batch and the variational assignments, rather than a "traditional" Bayesian posterior of the form $p(\mathbf{z} \mid y, \mathcal{D})$ derived from a generative model and a dataset-level likelihood. In particular, the normalization B_y is a *soft class count* over the mini-batch, and $\bar{q}_\phi^{(y)}(\mathbf{z})$ is a convex combination of per-sample posteriors $q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)$ weighted by $q_\phi(y \mid \mathbf{e}_i)$; it should be understood as a variational mixture distribution (and can vary across batches), not as an exact posterior obtained via Bayes' rule.

[8] Although we present the general soft-assignment derivation with $q_\phi(y \mid \mathbf{e}_i)$, our experiments use a known component assignment (hard labels), i.e., $q_\phi(y \mid \mathbf{e}_i) = \delta_{y=y_i}$, under which B_y reduces to the hard class count and $\bar{q}_\phi^{(y)}(\mathbf{z})$ becomes the empirical class-conditional aggregated posterior.

$$
\begin{aligned}
&\mathcal{L}_{\mathrm{cELBO}}(\theta,\phi) \\
&= \underbrace{\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big]}_{\text{(A) reconstruction term}} \;-\; \underbrace{\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y\mid\mathbf{e}_i)} \mathrm{D}_{KL}\big(q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)\,\|\,p(\mathbf{z}\mid y)\big)}_{\text{(B) class-conditional KL term}} \\
&= \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y\mid\mathbf{e}_i)} \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)}\Big[\log q_\phi(\mathbf{z}\mid\mathbf{e}_i,y) - \log p(\mathbf{z}\mid y)\Big] \\
&= \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y\mid\mathbf{e}_i)} \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)}\Big[\log q_\phi(\mathbf{z}\mid\mathbf{e}_i,y) - \log \bar{q}_\phi^{(y)}(\mathbf{z})\Big] \\
&\quad - \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y\mid\mathbf{e}_i)} \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)}\Big[\log \bar{q}_\phi^{(y)}(\mathbf{z}) - \log p(\mathbf{z}\mid y)\Big] \\
&= \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \sum_{y\in\{0,1\}} \sum_{i=1}^{B} q_\phi(y\mid\mathbf{e}_i)\, \mathrm{D}_{KL}\big(q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)\,\|\,\bar{q}_\phi^{(y)}(\mathbf{z})\big) \\
&\quad - \sum_{y\in\{0,1\}} B_y\, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z})\,\|\,p(\mathbf{z}\mid y)\big) \\
&= \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \sum_{y\in\{0,1\}} \underbrace{\left(\sum_{i=1}^{B} q_\phi(y\mid\mathbf{e}_i)\, \mathrm{D}_{KL}\big(q_\phi(\mathbf{z}\mid\mathbf{e}_i,y)\,\|\,\bar{q}_\phi^{(y)}(\mathbf{z})\big)\right)}_{\text{(C)}} \\
&\quad - \sum_{y\in\{0,1\}} B_y\, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z})\,\|\,p(\mathbf{z}\mid y)\big) \\
&= \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}\mid\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \sum_{y\in\{0,1\}} \underbrace{\Big(B_y\, I_\phi(\mathbf{e};\mathbf{z}\mid Y{=}y)\Big)}_{\text{(D)}} \\
&\quad - \sum_{y\in\{0,1\}} B_y\, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z})\,\|\,p(\mathbf{z}\mid y)\big).
\end{aligned}
\tag{8}
$$

In the above, (A) is the reconstruction term and (B) is the original class-conditional KL regularizer. The only nontrivial step is the conversion from **(C)** to **(D)**: namely, showing that the within-class KL aggregation equals the conditional mutual information term:

$$
\text{(C)} \;=\; B_y\, I_\phi(\mathbf{e};\mathbf{z}\mid Y{=}y).
$$

We provide this (C)→(D) derivation in detail below:

$$
\begin{aligned}
&\sum_{i=1}^{B} q_\phi(y \mid \mathbf{e}_i)\, \mathrm{D}_{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) \,\|\, \bar{q}_\phi^{(y)}(\mathbf{z})\big) \\
&= \sum_{i=1}^{B} q_\phi(y \mid \mathbf{e}_i)\, \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{e}_i,y)}\Big[\log q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) - \log \bar{q}_\phi^{(y)}(\mathbf{z})\Big] \\
&= \sum_{i=1}^{B} q_\phi(y \mid \mathbf{e}_i) \int q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)\Big[\log q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) - \log \bar{q}_\phi^{(y)}(\mathbf{z})\Big]\, \mathrm{d}\mathbf{z} \\
&= \sum_{i=1}^{B} q_\phi(y \mid \mathbf{e}_i) \int q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})}\, \mathrm{d}\mathbf{z} \\
&= B_y \sum_{i=1}^{B} \frac{q_\phi(y \mid \mathbf{e}_i)}{B_y} \int q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})}\, \mathrm{d}\mathbf{z} \\
&= B_y \sum_{i=1}^{B} q_\phi(\mathbf{e}_i \mid y) \int q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})}\, \mathrm{d}\mathbf{z} \\
&= B_y \int \sum_{i=1}^{B} q_\phi(\mathbf{e}_i \mid y)\, q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})}\, \mathrm{d}\mathbf{z} \\
&= B_y \int \sum_{i=1}^{B} q_\phi(\mathbf{e}_i, \mathbf{z} \mid y) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})}\, \mathrm{d}\mathbf{z} \\
&= B_y \int \sum_{i=1}^{B} q_\phi(\mathbf{e}_i, \mathbf{z} \mid y) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{q_\phi(\mathbf{z} \mid y)}\, \mathrm{d}\mathbf{z} \\
&= B_y \mathbb{E}_{q_\phi(\mathbf{e},\mathbf{z}|y)}\Big[\log q_\phi(\mathbf{z} \mid \mathbf{e}, y) - \log q_\phi(\mathbf{z} \mid y)\Big] \\
&= B_y\, I_\phi(\mathbf{e}; \mathbf{z} \mid Y = y).
\end{aligned}
\tag{9}
$$

The identity $q_\phi(\mathbf{e}_i \mid y) = \frac{q_\phi(y|\mathbf{e}_i)}{B_y}$ used above follows directly from a mini-batch normalization. Specifically, given the nonnegative weights $\{q_\phi(y \mid \mathbf{e}_i)\}_{i=1}^{B}$ and the (soft) class count $B_y := \sum_{j=1}^{B} q_\phi(y \mid \mathbf{e}_j)$, we define a discrete conditional distribution over mini-batch indices by:

$$
q_\phi(\mathbf{e}_i \mid y) := \frac{q_\phi(y \mid \mathbf{e}_i)}{\sum_{j=1}^{B} q_\phi(y \mid \mathbf{e}_j)} = \frac{q_\phi(y \mid \mathbf{e}_i)}{B_y}, \tag{10}
$$

which guarantees $\sum_{i=1}^{B} q_\phi(\mathbf{e}_i \mid y) = 1$. Then define the induced joint distribution $q_\phi(\mathbf{e}_i, \mathbf{z} \mid y) := q_\phi(\mathbf{e}_i \mid y)\, q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)$ and the corresponding marginal:

$$
q_\phi(\mathbf{z} \mid y) := \sum_{i=1}^{B} q_\phi(\mathbf{e}_i \mid y)\, q_\phi(\mathbf{z} \mid \mathbf{e}_i, y), \tag{11}
$$

which coincides with $\bar{q}_\phi^{(y)}(\mathbf{z})$ by construction. With these induced distributions, the last equality follows from the definition of conditional mutual information: $I_\phi(\mathbf{e}; \mathbf{z} \mid Y = y) = \mathbb{E}_{q_\phi(\mathbf{e},\mathbf{z}|y)}\Big[\log \frac{q_\phi(\mathbf{z}|\mathbf{e},y)}{q_\phi(\mathbf{z}|y)}\Big]$.

D Disentangled prompt-response embedding cELBO in Deep Manner

This appendix turns the decomposed cELBO in App. C into the deep mutual-information surrogate used by PREVAIL-MI. The decomposed objective isolates *reconstruction*, *within-class information*, and *class-wise aggregate matching*. We then use the observed preference class to construct a discriminator-based class-separation signal from samples. This gives the deep variant in Eq. (4), while preserving the same *reconstruction* and instance-level KL structure as the cELBO.

D.1 Mixture-posterior Surrogate

From Eq. (3), the term that is difficult to evaluate is:

$$\sum_{y\in\{0,1\}} B_y \, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z}) \,\|\, p(\mathbf{z} \mid y)\big), \tag{12}$$

because the class-conditional aggregated posterior $\bar{q}_\phi^{(y)}(\mathbf{z})$ is only represented by samples. To obtain a sample-based surrogate, we replace the component prior by the mini-batch mixture posterior:

$$\bar{q}_\phi(\mathbf{z}) := \sum_{y\in\{0,1\}} \hat{\pi}_y \, \bar{q}_\phi^{(y)}(\mathbf{z}), \qquad \hat{\pi}_y := \frac{B_y}{B}. \tag{13}$$

Then the class-wise aggregate matching surrogate can be written as:

$$\begin{aligned}\sum_{y\in\{0,1\}} \hat{\pi}_y \, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z}) \,\|\, \bar{q}_\phi(\mathbf{z})\big) &= \sum_{y\in\{0,1\}} \hat{\pi}_y \, \mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(y)}}\left[\log \frac{\bar{q}_\phi^{(y)}(\mathbf{z})}{\bar{q}_\phi(\mathbf{z})}\right] \\ &= I_{\bar{q}_\phi}(Y;\mathbf{z}).\end{aligned} \tag{14}$$

This quantity measures how much the latent code preserves the preference identity. It therefore provides a class-separation signal in the latent space, distinct from the within-class information term in Eq. (3).

D.2 Density-ratio Estimation

The density ratio in Eq. (14) can be estimated by a binary discriminator. Let the aggregated joint be:

$$\bar{q}_\phi(y,\mathbf{z}) = \hat{\pi}_y \, \bar{q}_\phi^{(y)}(\mathbf{z}). \tag{15}$$

The optimal discriminator for predicting the preference class from $\mathbf{z}$ is:

$$D_{\omega^\star}(\mathbf{z}) = \bar{q}_\phi(y=1 \mid \mathbf{z}) = \frac{\hat{\pi}_1 \, \bar{q}_\phi^{(1)}(\mathbf{z})}{\hat{\pi}_1 \, \bar{q}_\phi^{(1)}(\mathbf{z}) + \hat{\pi}_0 \, \bar{q}_\phi^{(0)}(\mathbf{z})}. \tag{16}$$

Thus the discriminator gives the class-posterior density ratio directly. Equivalently, its binary cross-entropy estimates the conditional entropy term in $I_{\bar{q}_\phi}(Y;\mathbf{z}) = H_{\bar{q}_\phi}(Y) - H_{\bar{q}_\phi}(Y \mid \mathbf{z})$. Since $H_{\bar{q}_\phi}(Y)$ is fixed by the mini-batch class proportions, minimizing the discriminator cross-entropy increases the preference information retained in $\mathbf{z}$.

D.3 Practical Deep Objective

Using the discriminator surrogate, the deep variant keeps the *reconstruction* term and the instance-level KL regularization, and adds a class-separation penalty:

$$\begin{aligned}\mathcal{L}^{\mathrm{MI}}_{\mathrm{PREVAIL}}(\theta,\phi,\omega) \;:=\; & \sum_{i=1}^{B} \mathbb{E}_{q_\phi(\mathbf{z}\mid\mathbf{e}_i,y_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] - \beta \sum_{i=1}^{B} \mathrm{D}_{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{e}_i, y_i) \,\|\, p(\mathbf{z} \mid y_i)\big) \\ & - \lambda B \, \widehat{\mathcal{C}}_\omega, \\ \widehat{\mathcal{C}}_\omega \;:=\; & -\Big(\hat{\pi}_1 \, \mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(1)}}[\log D_\omega(\mathbf{z})] + \hat{\pi}_0 \, \mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(0)}}[\log(1 - D_\omega(\mathbf{z}))]\Big).\end{aligned} \tag{17}$$

This is the implementation-ready objective used in Eq. (4). The coefficient β controls instance-level prior regularization, while λ controls the strength of the discriminator-based preference-identification signal. Because these two terms have different scales and can exert competing pressures on latent separation, PREVAIL-MI should be interpreted as a practical cELBO-motivated surrogate whose final separation also depends on the class-conditioned posterior heads. The next appendix gives the complementary statistical variant, where aggregate matching is handled by an MMD penalty rather than a discriminator.

E Disentangled prompt-response embedding cELBO in Statistical Manner

Following the InfoVAE/InfoDiffusion reformulation [32, 33, 46], we further rewrite the class-wise aggregate matching part of our decomposed cELBO into an equivalent form that makes the trade-off between (i) *aggregated distribution matching* and (ii) *per-sample* latent-information pressure explicit. Throughout this appendix, $Y \in \{0,1\}$ denotes the preference label, $\mathbf{e}$ is the prompt–response embedding, $\mathbf{z}$ is the continuous latent code, $B_y := \sum_{i=1}^{B} q_\phi(y \mid \mathbf{e}_i)$ and $\hat{\pi}_y := B_y/B$. We use $\bar{q}_\phi^{(y)}(\mathbf{z})$ for the class-conditional aggregated posterior and $\bar{q}_\phi(\mathbf{z}) := \sum_y \hat{\pi}_y \, \bar{q}_\phi^{(y)}(\mathbf{z})$.

$$
\begin{aligned}
& -\alpha \sum_{y\in\{0,1\}} B_y \, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z}) \,\|\, p(\mathbf{z} \mid y)\big) - \sum_{y\in\{0,1\}} B_y \, I_\phi(\mathbf{e}; \mathbf{z} \mid Y = y) \\
& \quad + \gamma \sum_{y\in\{0,1\}} B_y \, I_\phi(\mathbf{e}; \mathbf{z} \mid Y = y) \\
= \; & -\alpha \sum_{y\in\{0,1\}} B_y \, \mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(y)}} \left[\log \frac{\bar{q}_\phi^{(y)}(\mathbf{z})}{p(\mathbf{z} \mid y)} \right] \\
& \quad - (1-\gamma) \sum_{y\in\{0,1\}} B_y \, \mathbb{E}_{\mathbf{e}\sim\mathcal{D}_y} \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{e},y)} \left[\log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})} \right] \\
= \; & -\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{e}_i)} \left[\alpha \log \frac{\bar{q}_\phi^{(y)}(\mathbf{z})}{p(\mathbf{z} \mid y)} + (1-\gamma) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{\bar{q}_\phi^{(y)}(\mathbf{z})} \right] \\
= \; & -\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{e}_i)} \left[\alpha \log \frac{\bar{q}_\phi^{(y)}(\mathbf{z})}{p(\mathbf{z} \mid y)} + (1-\gamma) \Big(\log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{p(\mathbf{z} \mid y)} - \log \frac{\bar{q}_\phi^{(y)}(\mathbf{z})}{p(\mathbf{z} \mid y)} \Big) \right] \\
= \; & -\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{e}_i)} \left[(\alpha+\gamma-1) \log \frac{\bar{q}_\phi^{(y)}(\mathbf{z})}{p(\mathbf{z} \mid y)} + (1-\gamma) \log \frac{q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)}{p(\mathbf{z} \mid y)} \right] \\
= \; & -(\alpha+\gamma-1) \sum_{y\in\{0,1\}} B_y \, \mathrm{D}_{KL}\big(\bar{q}_\phi^{(y)}(\mathbf{z}) \,\|\, p(\mathbf{z} \mid y)\big) \\
& \quad - (1-\gamma) \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y|\mathbf{e}_i)} \mathrm{D}_{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{e}_i, y) \,\|\, p(\mathbf{z} \mid y)\big).
\end{aligned}
\tag{18}
$$

As in InfoVAE, the *aggregated distribution matching* term can be replaced by a kernel two-sample distance, avoiding explicit density (ratio) estimation [32, 33]. We adopt maximum mean discrepancy (MMD), defined as:

$$
\begin{aligned}
\mathrm{MMD}_k^2\big(\bar{q}_\phi(\mathbf{z}), p(\mathbf{z})\big) = \; & \mathbb{E}_{\mathbf{z},\mathbf{z}'\sim\bar{q}_\phi(\mathbf{z})}\big[k(\mathbf{z},\mathbf{z}')\big] + \mathbb{E}_{\mathbf{z},\mathbf{z}'\sim p(\mathbf{z})}\big[k(\mathbf{z},\mathbf{z}')\big] \\
& - 2\,\mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi(\mathbf{z}),\, \mathbf{z}'\sim p(\mathbf{z})}\big[k(\mathbf{z},\mathbf{z}')\big],
\end{aligned}
\tag{19}
$$

where k is a positive definite kernel. Since $\bar{q}_\phi(\mathbf{z})$ is a two-component mixture, the first expectation expands to:

$$
\begin{aligned}
\mathbb{E}_{\mathbf{z},\mathbf{z}'\sim\bar{q}_\phi(\mathbf{z})}[k(\mathbf{z},\mathbf{z}')] = \; & \hat{\pi}_1^2 \, \mathbb{E}_{\mathbf{z},\mathbf{z}'\sim\bar{q}_\phi^{(1)}}[k(\mathbf{z},\mathbf{z}')] + \hat{\pi}_0^2 \, \mathbb{E}_{\mathbf{z},\mathbf{z}'\sim\bar{q}_\phi^{(0)}}[k(\mathbf{z},\mathbf{z}')] \\
& + 2\hat{\pi}_1\hat{\pi}_0 \, \mathbb{E}_{\mathbf{z}\sim\bar{q}_\phi^{(1)},\, \mathbf{z}'\sim\bar{q}_\phi^{(0)}}[k(\mathbf{z},\mathbf{z}')].
\end{aligned}
\tag{20}
$$

Following InfoVAE [32] (and its usage in [33]), we replace the *aggregated distribution matching* term by an MMD penalty, yielding the following implementation-ready surrogate. The MMD penalty matches the aggregated mixture to the prior; it does not directly maximize the distance between preferred and rejected class-conditional distributions. Preference-aware separation arises from the class-conditioned posterior heads and the per-sample information pressure controlled by γ under this global aggregate constraint.

$$\begin{aligned}\mathcal{L}^{\mathrm{MMD}}_{\mathrm{PREVAIL}}(\theta,\phi) \;:=\; & \sum_{i=1}^{B} \mathbb{E}_{q_\phi(y,\mathbf{z}|\mathbf{e}_i)}\big[\log p_\theta(\mathbf{e}_i \mid \mathbf{z})\big] \\ & - (1-\gamma)\sum_{i=1}^{B} \mathbb{E}_{q_\phi(y|\mathbf{e}_i)} \mathrm{D}_{KL}\big(q_\phi(\mathbf{z} \mid \mathbf{e}_i, y)\,\|\,p(\mathbf{z} \mid y)\big) \\ & - (\alpha+\gamma-1)\, B \cdot \mathrm{MMD}_k^2\big(\bar{q}_\phi(\mathbf{z}),\, p(\mathbf{z})\big).\end{aligned} \tag{21}$$

F RKHS Surrogate Analysis for Preference Ranking under Disentangled Generation

This appendix provides a short RKHS-surrogate analysis for preference-ranker training with disentangled synthetic preference pairs. The analysis does not exactly characterize the MLP reward model used in the experiments; it only formalizes why distribution closeness between real and generated preference pairs is a useful motivation for synthetic generation. Let $u := (\mathbf{e}^+, \mathbf{e}^-)$ denote a preference pair in the pair space $\mathcal{U}$. Let P be the true pair distribution induced by sampling $\mathbf{e}^+ \sim P_1$ and $\mathbf{e}^- \sim P_0$, and let $\hat{P}$ be the synthetic pair distribution induced by our generator. We train the preference ranker on the mixture:

$$P_{\mathrm{mix}} \;:=\; (1-\eta)P + \eta\hat{P}, \qquad \eta \in [0,1].$$

Assume the pairwise ranking loss $\ell_\psi(u)$ belongs to an RKHS $\mathcal{H}_k$ induced by a positive definite kernel $k : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ and satisfies $\|\ell_\psi\|_{\mathcal{H}_k} \le C$.

Risk-transfer statement. Under the RKHS surrogate assumption above, the population risks satisfy:

$$\Big|\, \mathbb{E}_{u\sim P}\big[\ell_\psi(u)\big] - \mathbb{E}_{u\sim P_{\mathrm{mix}}}\big[\ell_\psi(u)\big] \,\Big| \;\le\; \eta\, C \cdot \mathrm{MMD}_k^{\mathrm{pair}}(P, \hat{P}),$$

where $\mathrm{MMD}_k^{\mathrm{pair}}$ denotes the (unsquared) maximum mean discrepancy induced by k over the preference-pair space $\mathcal{U}$.

Proof. Using the mixture definition and the RKHS form of MMD, the deviation between the original and generated-pair risks is bounded as:

$$\begin{aligned}&\Big|\mathbb{E}_{u\sim P}[\ell_\psi(u)] - \mathbb{E}_{u\sim P_{\mathrm{mix}}}[\ell_\psi(u)]\Big| \\ &\quad= \Big|\mathbb{E}_{u\sim P}[\ell_\psi(u)] - (1-\eta)\mathbb{E}_{u\sim P}[\ell_\psi(u)] - \eta\mathbb{E}_{u\sim\hat{P}}[\ell_\psi(u)]\Big| \\ &\quad= \eta\Big|\mathbb{E}_{u\sim P}[\ell_\psi(u)] - \mathbb{E}_{u\sim\hat{P}}[\ell_\psi(u)]\Big| \\ &\quad\le \eta\,\|\ell_\psi\|_{\mathcal{H}_k} \sup_{\|f\|_{\mathcal{H}_k}\le 1}\Big|\mathbb{E}_{u\sim P}[f(u)] - \mathbb{E}_{u\sim\hat{P}}[f(u)]\Big| \\ &\quad= \eta\,\|\ell_\psi\|_{\mathcal{H}_k}\,\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}) \;\le\; \eta\, C\,\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}).\end{aligned}$$

The first equality expands the generated-pair mixture distribution P_{mix} by linearity of expectation. The only inequality is the standard MMD/RKHS scaling step: for any nonzero $f \in \mathcal{H}_k$, normalize $f/\|f\|_{\mathcal{H}_k}$ inside the unit ball in the MMD definition, which gives $|\mathbb{E}_P f - \mathbb{E}_{\hat{P}} f| \le \|f\|_{\mathcal{H}_k}\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P})$. Applying this with $f = \ell_\psi$ and $\|\ell_\psi\|_{\mathcal{H}_k} \le C$ gives the stated bound.

□

F.1 Bound-Level Condition for Useful Synthesis

The bound above only controls the distributional bias introduced by synthetic pairs. We now state a simple condition under which the finite-sample upper bound can become tighter after adding synthetic pairs. This is a bound-level statement, not a guarantee that the learned neural ranker achieves lower expected risk. Let $\mathcal{B}_N$ denote the finite-sample risk upper bound when training on N real preference pairs, and let $\mathcal{B}_{\mathrm{mix}}$ denote the corresponding upper bound when training on N real pairs and M synthetic pairs drawn from $\hat{P}$. Assume the stochastic estimation term follows the standard $O(n^{-1/2})$ rate with constant $A > 0$, while the synthetic distribution bias is controlled by Prop. 1. Then, the upper-bound difference satisfies:

$$\begin{aligned}\mathcal{B}_N - \mathcal{B}_{\mathrm{mix}} &\geq \frac{A}{\sqrt{N}} - \left(\frac{A}{\sqrt{N+M}} + \eta C\,\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P})\right) \\ &= A\left(\frac{1}{\sqrt{N}} - \frac{1}{\sqrt{N+M}}\right) - \eta C\,\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}) \\ &\geq 0,\end{aligned}$$

provided that:

$$\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}) \leq \frac{A}{\eta C}\left(\frac{1}{\sqrt{N}} - \frac{1}{\sqrt{N+M}}\right).$$

If $M = \rho N$ for a fixed synthesis ratio $\rho > 0$, this condition becomes:

$$\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}) \leq \underbrace{\frac{A}{\eta C}\left(1 - \frac{1}{\sqrt{1+\rho}}\right)}_{C'} \frac{1}{\sqrt{N}}.$$

Thus, under the standard concentration assumption, the surrogate upper bound can be tightened when the generated pair distribution is sufficiently close to the true pair distribution, with the required closeness scaling as $O(N^{-1/2})$. This condition should be interpreted as qualitative support for controlling generated-pair distribution shift, rather than as a strict guarantee that synthetic data improves the learned MLP ranker.

F.2 Squared MMD Convention

If the main text uses the squared pair-distribution MMD, $(\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}))^2$, then:

$$\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}) \;=\; \sqrt{(\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}))^2}.$$

In that case, the bound above can be equivalently written using $\sqrt{(\mathrm{MMD}_k^{\mathrm{pair}}(P,\hat{P}))^2}$.

G Qualitative Examples

We present qualitative case studies to illustrate the response selections made by the two PREVAIL variants. For each example, we sample 16 candidate responses from LLAMA-3.1-8B-INSTRUCT for a given prompt from the TL;DR test set, rank them with each reward model, and show the top-ranked response along with its SKYWORK gold reward score.

Example I

Subreddit: r/AskReddit **Post title:** I want to quit the band I'm in, what's the best way to handle this situation?

Post: So I'm a guitar player in a metal band. I've been in the band for about a year now and it's been really fun, but lately I just don't see it going anywhere and I am getting increasingly annoyed by my bandmates. The reason I joined was because the old guitarist who was an original member quit and they needed a new guitarist/songwriter. Since then, we've gone through 2 drummers and our current one is still not quite at the right skill level.

PREVAIL-MMD RM selected response:

> *I'm a guitar player in a metal band that's not going anywhere, and my bandmates are annoying me. How do I get out of this situation without hurting anyone's feelings?*

⇒ SKYWORK score: **1.08**

PREVAIL-MI RM selected response:

> *Band is not going well and I want to leave. I'm not sure how to tell them.*

⇒ SKYWORK score:**2.15**

Example II

Subreddit: r/jobs **Post title:** Question about non-compete clauses...

Post: So I just received an offer from a company I am thrilled to have the chance to work for. The position is right, the pay is right, and the work/life balance looks to be promising. They sent over a formal offer for me to look at, in addition to an additional agreement that needs signed with a non-compete agreement. I completely understand the need for such an agreement, as the company will be extensively investing their time and resources in training me for the job.

PREVAIL-MMD RM selected response:

> *What are your thoughts on non-compete agreements? Are they enforceable? Can I still pursue other career opportunities if I sign one?*

⇒ SKYWORK score: **0.76**

PREVAIL-MI RM — selected response:

> *Received a job offer with a non-compete clause and was wondering if it's worth considering the opportunity, or if I should negotiate or try to get out of it?*

⇒ SKYWORK score: **2.53**

Example III

Subreddit: r/tifu **Post title:** TIFU by rescuing a feral kitten

Post: So I just came back from studying abroad and have been staying at my parents house until I move into my college apartment on Monday. The house is in a quiet town, kind of suburb-ish. The backyard ends with woods and hiking trails. Anyways, my 80 year old aunt said she had a litter of kittens living underneath her porch.

PREVAIL-MMD RM selected response:

> *I decided to rescue the feral kittens. Now I'm stuck with one of them and have no idea what to do.*

⇒ SKYWORK score: **1.80**

PREVAIL-MI RM selected response:

> *Found feral kitten under aunt's porch, rescued it, now can't sleep due to kitten's constant meowing.*

⇒ SKYWORK score: **1.88**

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the proposed cELBO framework, the two PREVAIL variants, and the empirical scope. The risk-control claim is stated conservatively and tied to the distribution-closeness assumption in Sec. 3.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated Limitations paragraph in Sec. 5 states the clean binary-label assumption, the structural argument for robustness to label noise, and the scope of the evaluated tasks.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: The main theoretical result states the RKHS/IPM assumption in Prop. 1, and the proof is provided in App. F. The appendix also gives the sufficient condition for beneficial augmentation and the MMD convention.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Sec. 4 and App. B describe the datasets, embedding extraction, VAE training, reward-model training, SFT validation, evaluation metrics, and random seeds used for the main experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The abstract provides a public code repository, and the experiments use publicly available datasets and models with URLs listed in App. B. The appendix describes the main reproduction settings.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: App. B specifies the dataset splits, embedding backbone, sequence length, optimizer, learning rates, batch sizes, epochs, model dimensions, seeds, and evaluation protocols.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The main BoN-16 results in Tab. 1 and the architecture-controlled ablation table in App. B.6 are reported as mean and standard deviation over three random seeds, and the seed values are specified in App. B.4 and App. B.6.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The compute resources are included in Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses public benchmark datasets and open or cited models for preference-modeling research, and no new human-subject data collection is conducted.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

Answer: [No]

Justification: The paper motivates improved alignment with human preferences as a positive impact, but it does not include a dedicated discussion of negative societal impacts or misuse risks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This work does not release high-risk models or scraped datasets. Any synthetic preference samples are derived from existing public benchmarks and are intended only for research evaluation under the corresponding dataset terms.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

Answer: [No]

Justification: The paper credits and cites the datasets, models, and baselines used in the experiments, but it does not explicitly list all licenses and terms of use for each existing asset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces and releases code for PREVAIL, with the repository linked in the abstract. The method and experimental configurations are documented in the main text and appendix.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

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

    Answer: [N/A]

    Justification: The paper does not conduct new crowdsourcing experiments or new human-subject studies; it uses existing public preference datasets.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

Answer: [N/A]

Justification: No new human-subject research or crowdsourcing was performed for this work, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLMs were used only for grammar checking and formula-format consistency, and did not affect the core methodology, scientific rigor, or originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.