Limited Preference Data? Learning Better Reward Model with Latent Space Synthesis

Leitian Tao[1] **Xuefeng Du**[2] **Sharon Li**[1]
[1]Department of Computer Sciences, University of Wisconsin-Madison
[2]College of Computing and Data Science, Nanyang Technological University
leitiantao@cs.wisc.edu, xuefeng.du@ntu.edu.sg, sharonli@cs.wisc.edu

Abstract

Reward modeling, crucial for aligning large language models (LLMs) with human preferences, is often bottlenecked by the high cost of preference data. Existing textual data synthesis methods are computationally expensive. We propose a novel framework **LENS** for synthesizing preference data directly in the LLM's latent embedding space. Our method employs a Variational Autoencoder (VAE) to learn a structured latent representation of response embeddings. By performing controlled perturbations in this latent space and decoding back to the embedding space, we efficiently generate diverse, semantically consistent synthetic preference pairs, bypassing costly text generation and annotation. We provide theoretical guarantees that our synthesized pairs approximately preserve original preference ordering and improve reward model generalization. Empirically, our latent-space synthesis significantly outperforms text-based augmentation on standard benchmarks, achieving superior results while being 18× faster in generation and using a 16,000× smaller model. Our work offers a scalable and effective alternative for enhancing reward modeling through efficient data augmentation. Code is publicly available at https://github.com/deeplearning-wisc/lens.

1 Introduction

As large language models (LLMs) are increasingly deployed in settings that interact with or influence people [1, 2, 3, 4], translating human feedback into a reward function has become a cornerstone of AI development [5]. Reward modeling, which learns to assign higher scores to preferred responses over rejected ones, is a critical component in post-training and decision-making systems [6, 7, 8, 9]. A strong reward model can be used not only to supervise language model behavior but also to support high-leverage tasks like rejection sampling, preference ranking, and quality estimation.

Despite its importance, reward modeling remains bottlenecked by data collection and computational cost. Human preference labels are expensive and time-consuming to collect at scale. To address the high cost of collecting human preference data, researchers have explored textual space synthesis approaches [10, 11, 12, 13, 14]. As shown in Figure 1, these methods typically involve a two-stage process: first, using LLMs to generate multiple diverse responses to the same prompt; then, employing an auxiliary LLM to annotate these responses by creating pairwise preference data, which is subsequently used for reward model training. However, this approach faces significant computational challenges. The response generation phase requires substantial computational resources to produce diverse, high-quality candidates. The preference annotation phase is also resource-intensive, as it requires running LLMs to evaluate and rank each response pair, which scales quadratically with the

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

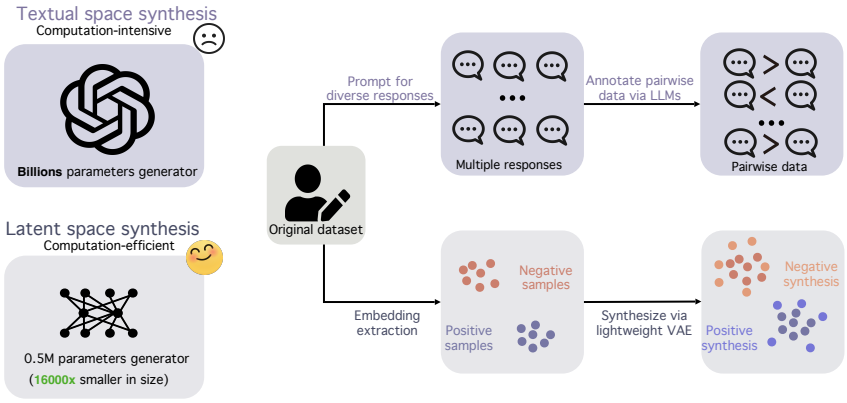

Figure 1: Comparison of textual space synthesis (top) and latent space synthesis (bottom). Latent space synthesis operates on embeddings, offering significant computational advantages. Best viewed in color.

number of responses per prompt. These challenges raise the question: ***Given limited preference data, can we efficiently expand the dataset to improve reward modeling***?

To address these challenges, we propose a new framework–**LENS** (Latent EmbeddiNg for Synthesis)—which synthesizes preference data *in the latent space* rather than the textual space. LENS bypasses the computational expense of text generation, avoids complex prompt engineering, and leverages the semantic structure already captured by the language model embeddings. To model and generate plausible preference data, we leverage a Variational Autoencoder (VAE), a generative model that learns a smooth and structured distribution in its latent space. The VAE is particularly well-suited for this task: mapping each input to a distribution over latent variables rather than a fixed point enables localized sampling that preserves semantic consistency while introducing meaningful diversity—which is critical for generating synthetic preference pairs. In particular, we synthesize new preference pairs by performing controlled perturbations in the learned latent space of VAE, which can then be decoded back to the LLM embedding space. This creates an augmented dataset to enhance reward modeling without requiring additional human annotation and bypassing expensive text generation.

Importantly, our approach comes with theoretical guarantees. We show that synthetic preference pairs generated by LENS preserve the original preference ordering, up to a bounded error that depends on the noise level and reconstruction quality of VAE (Theorem 1). This establishes that preference relationships are maintained after latent-space perturbation and decoding. Moreover, we prove that augmenting the original training set with these synthetic samples reduces the error upper bound in reward modeling by effectively increasing the sample size while maintaining preference consistency (Theorem 2). Beyond theoretical understanding, we further validate our method on widely used reward modeling benchmarks, demonstrating that latent-space synthesis consistently outperforms text-based synthesis approaches. Notably, our method requires significantly fewer computational resources to sample new preference data, requiring only 0.5M parameters compared to billions of LLM parameters for the text-based synthesis method, reducing the generator size by 16,000$\times$. These efficiency gains make our approach highly scalable and practical for real-world deployment, especially in resource-constrained settings. We summarize our contributions below:

1. We propose a novel framework LENS for synthesizing preference data directly in the latent embedding space of language models, enabling efficient reward modeling without the need for text generation or heavy prompt engineering.
2. We design a contrastive VAE that learns to generate diverse yet semantically meaningful synthetic embeddings and provide a theoretical analysis showing how latent-space synthesis improves the generalization of reward modeling.
3. We demonstrate strong empirical results across reward modeling benchmarks, achieving superior performance to text-based augmentation while being 18× faster and 16,000× smaller in model size, with detailed ablations validating design choices.

2 Preliminaries

Large language models map a natural language prompt $x \in \mathcal{X}$ to a generated response $y \in \mathcal{Y}$, defining a probability distribution $p_\theta(y \mid x)$ over the vocabulary space $\mathcal{V}$. The reward model acts as an auxiliary function that evaluates the quality of responses, which is trained to produce a higher

score for the preferred response given a query. Reward modeling relies on preference-labeled data, typically collected as pairwise comparisons. Formally, we define the preference data below.

Definition 1 (Preference Data.). *Consider two responses y^+, y^- for an input prompt x, we denote $y^+ \succ y^-$ if y^+ is preferred over y^-. We call y^+ the preferred response and y^- the rejected response. Each triplet (x, y^+, y^-) is referred to as a preference. Furthermore, the empirical dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$ consists of N such triplets sampled from a preference distribution $\mathcal{P}$.*

Reward modeling objective. Reward modeling learns a function mapping, which takes in the prompt x and response y and outputs a scalar value $r(x, y)$ signifying the reward. A preferred response should receive a higher reward, and vice versa. Based on the Bradley–Terry model [15], the reward function is optimized over a dataset of human preferences, with the following objective:

$$\mathcal{L}_{RM}^{\mathcal{D}} = -\mathbb{E}_{(x,y^+,y^-)\in\mathcal{D}}[\log \sigma(r(x, y^+) - r(x, y^-))], \tag{1}$$

where σ denotes the sigmoid function. This objective encourages the reward function to produce the observed orderings. Once trained, the reward model can be used to score future responses. *Training high-quality reward models remains heavily bottlenecked by the cost of collecting large-scale human preference data, which is both time-consuming and labor-intensive*. Our work hence focuses on the practical setting where N is moderately small. Recent work has explored augmenting preference datasets through textual response synthesis and LLM-based annotation [12, 13, 14], but such pipelines are computationally intensive, often requiring powerful models for both generation and labeling. These challenges motivate our approach to synthesize preference data directly in the latent embedding space of the language model, bypassing the need for costly generation and annotation and thus offering stronger efficiency and scalability.

3 Methodology

In this section, we introduce our methodology to efficiently generate preference data by synthesizing new samples directly in the latent space, rather than in the text space. This paradigm shift avoids expensive text generation, mitigates complex prompt engineering, and leverages the semantic structure already captured by the language model. Our framework consists of three main stages: (1) training a variational autoencoder with divergence learning on response embeddings (see Section 3.1), (2) generating synthetic preference pairs in the learned latent embedding space (see Section 3.2), and (3) training a reward model on a combination of original and synthesized preference data (see Section 3.3). The subsections below describe each component in detail.

3.1 Variational Autoencoder with Divergence Learning

We begin by extracting embeddings for the language model's responses. Given a preference dataset $\mathcal{D} = \{(x_i, y_i^+, y_i^-)\}_{i=1}^N$ as defined in Section 2, where x_i is the prompt, y_i^+ is the preferred response, and y_i^- is the non-preferred response, we extract the LLM embedding vectors for each preference triplet in the dataset:

$$\mathbf{e}_i^{\pm} = \text{LLM}_{\text{embed}}(x_i, y_i^{\pm}), \tag{2}$$

where $\text{LLM}_{\text{embed}}$ represents the embedding extraction function that processes each prompt-response pair through the language model and returns the final hidden state representation at the last layer. These embeddings capture the semantic representation of the responses and serve as the starting point for our VAE-based synthesis.

Learning latent representations with divergence-aware VAE. To enhance the reward model, we synthesize additional training pairs by leveraging a VAE [16], a generative model that learns a probabilistic latent representation of preference data. VAE is a natural choice due to its ability to learn a smooth and structured latent space, enabling the sampling of diverse yet semantically coherent synthetic preference data. In particular, VAEs map each input to a distribution over latent variables rather than a single point. This is crucial in our context, as it enables diverse yet semantically consistent sampling around a given embedding, which is important for generating diverse synthetic preference data for better reward modeling.

Specifically, given a dataset of preference embeddings $\mathcal{E} = \{\mathbf{e}_i^+, \mathbf{e}_i^-\}_{i=1}^N$, where each $\mathbf{e}_i^+$ and $\mathbf{e}_i^-$ denotes the original LLM embedding of a preferred and non-preferred response, the VAE encoder

$\phi : \mathbb{R}^d \to \mathbb{R}^{2d_{\text{VAE}}}$ transforms each d-dimensional LLM embedding into the posterior Gaussian parameter vectors $\boldsymbol{\mu}_\phi|\mathbf{e} \in \mathbb{R}^{d_{\text{VAE}}}$ and $\boldsymbol{\sigma}^2_\phi|\mathbf{e} \in \mathbb{R}^{d_{\text{VAE}}}$, and d_{VAE} is the dimension of the VAE latent space. The posterior distributions for the positive and negative embeddings are parameterized as:

$$q_\phi(\mathbf{z}|\mathbf{e}^+) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{e}^+), \boldsymbol{\sigma}_\phi(\mathbf{e}^+)^2 \cdot \mathbf{I}), \quad q_\phi(\mathbf{z}|\mathbf{e}^-) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_\phi(\mathbf{e}^-), \boldsymbol{\sigma}_\phi(\mathbf{e}^-)^2 \cdot \mathbf{I}). \tag{3}$$

The decoder then reconstructs the embeddings from samples in the latent space. This reconstruction process, denoted as $\hat{\mathbf{e}} = g_\theta(\mathbf{z})$, transforms the latent representation back into the original embedding space. The VAE loss for each sample (either a positive or a negative embedding) is given by:

$$\mathcal{L}_{\text{VAE}}(\mathbf{e}) = \mathcal{L}_{\text{recon}}(\mathbf{e}, \hat{\mathbf{e}}) + \beta \cdot D_{\text{KL}}\left(q_\phi(\mathbf{z}|\mathbf{e}) \,\|\, p(\mathbf{z})\right), \tag{4}$$

Where $\mathcal{L}_{\text{recon}}$ denotes a reconstruction loss between original and reconstructed embeddings, and $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard isotropic Gaussian prior. The hyperparameter β balances reconstruction quality and latent regularization.

To encourage the model to learn discriminative representations, we introduce a divergence term to maximize the separation between the latent distributions of positive and negative samples:

$$\mathcal{L}_{\text{divergence}} = -\frac{1}{N}\sum_{i=1}^{N} W_2\Big(q_\phi(\mathbf{z}^+|\mathbf{e}_i^+),\, q_\phi(\mathbf{z}^-|\mathbf{e}_i^-)\Big), \tag{5}$$

Where W_2 denotes the Wasserstein distance. The overall objective is defined as:

$$\mathcal{L}_{\text{total}} = \frac{1}{N}\sum_{i=1}^{N}\left[\mathcal{L}_{\text{VAE}}(\mathbf{e}_i^+) + \mathcal{L}_{\text{VAE}}(\mathbf{e}_i^-)\right] + \gamma \cdot \mathcal{L}_{\text{divergence}}, \tag{6}$$

with hyperparameter γ controlling the importance of divergence term. We provide ablations on the impact of both β and γ in Section 5.3 and Appendix A.2.

3.2 Latent Space Synthesis

Latent space sampling. Once the VAE has been trained on preference embeddings, we synthesize new data by performing controlled perturbations on the known pairwise responses in the learned latent space. The goal is to generate new embeddings that maintain the semantic representation of the original responses while preserving preference relationships. For each embedding, multiple noisy variants are generated by adding Gaussian noise to its latent vector $\mathbf{z}_i^\pm$; these noisy latent vectors are subsequently decoded to generate new embeddings $\hat{\mathbf{e}}_{i,j}^\pm$:

$$\hat{\mathbf{e}}_{i,j}^\pm = g_\theta(\mathbf{z}_i^\pm + \boldsymbol{\eta}_{i,j}^\pm), \quad \text{where } \boldsymbol{\eta}_{i,j}^\pm \sim \mathcal{N}(0, \sigma^2_{\text{noise}}\mathbf{I}), \tag{7}$$

where j indexes the number of synthetic samples per original embedding, and σ_{noise} is a hyperparameter controlling the noise level. To ensure quality and consistency, we select the top-k synthetic latent codes $(\mathbf{z}_i^\pm + \boldsymbol{\eta}_{i,j}^\pm)$ based on their likelihood under the learned distributions, before decoding them.

Forming synthetic preference pairs. To expand the training data, we form synthetic preference pairs by pairing each synthesized or original preferred embedding with both synthesized and original non-preferred embeddings, and vice versa. This compositional pairing strategy gets an augmented dataset:

$$\mathcal{E}_{\text{aug}} = \left\{(\tilde{\mathbf{e}}^+, \tilde{\mathbf{e}}^-) \;\middle|\; \tilde{\mathbf{e}}^+ \in \mathcal{E}^+ \cup \mathcal{E}^+_{\text{synth}},\ \tilde{\mathbf{e}}^- \in \mathcal{E}^- \cup \mathcal{E}^-_{\text{synth}}\right\}, \tag{8}$$

where $\mathcal{E}^+_{\text{synth}} = \{\hat{\mathbf{e}}^+_{i,j}\}$ and $\mathcal{E}^-_{\text{synth}} = \{\hat{\mathbf{e}}^-_{i,j}\}$ denote the sets of synthesized positive and negative embeddings, respectively.

3.3 Reward Modeling with the Synthesized Preference Data

The augmented dataset is used to train a reward model that can distinguish preferred data from rejected ones. During optimization, these pairs contribute to the reward modeling objective:

$$\mathcal{L}_{RM}^{\mathcal{E}_{\text{aug}}} = -\mathbb{E}_{(\tilde{\mathbf{e}}^+, \tilde{\mathbf{e}}^-)\in\mathcal{E}_{\text{aug}}}\left[\log \sigma(r_o(\tilde{\mathbf{e}}^+) - r_o(\tilde{\mathbf{e}}^-))\right], \tag{9}$$

where the function r_o is a lightweight MLP mapping the embedding to the reward score. This loss encourages the model to assign higher scores to preferred embeddings and vice versa. We adopt an

MLP architecture for reward modeling because recent advances in reward modeling have highlighted such training approach as a lightweight yet effective alternative to full fine-tuning [17, 18, 19, 20, 21, 22, 23]. In Section 5, we show that this embedding-based reward modeling approach outperforms full fine-tuning methods trained on text-space augmentations, while requiring significantly less computing. By training on both real and synthesized pairs, the reward model benefits from exposure to a wider range of preference pairs, ultimately leading to better generalization.

4 Theoretical Analysis

In this section, we provide a theoretical analysis to support our proposed algorithm. As an overview, Theorem 1 analyzes the ***quality*** of the synthetic preference pairs under the best possible reward function. We then provably investigate the ***learnability*** of the reward model trained with the synthesized preference data in Theorem 2, demonstrating that it can be better than the reward model trained without synthesis under certain regulatory conditions. We specify several mild assumptions and necessary notations for our theorems in Appendix B.1. Due to space limitations, we omit unimportant constants and simplify the statements of our theorems. We defer the full formal statements in Appendix B.2. All proofs can be found in Appendix B.3.

4.1 Analysis on Synthesis Quality

We first analyze the quality of the synthetic preference embeddings $(\hat{\mathbf{e}}^+, \hat{\mathbf{e}}^-)$ by the best possible reward MLP function that serves as an ideal evaluator trained over the original preference data distribution. Specifically, let $\mathcal{R}_o$ denote the hypothesis space of the reward MLP model, and $\mathcal{P}_e$ as the embedding distribution of the original preference dataset, the best possible reward function is formulated as $r_o^* = \operatorname{argmin}_{r_o \sim \mathcal{R}_o} \mathbb{E}_{(\mathbf{e}^+, \mathbf{e}^-) \sim \mathcal{P}_e} [-\log \sigma(r_o(\mathbf{e}^+) - r_o(\mathbf{e}^-))]$. Based on r_o^*, the reward difference between the synthesized positive and negative pairs $r_o^*(\hat{\mathbf{e}}^+) - r_o^*(\hat{\mathbf{e}}^-)$ has the following bound:

Theorem 1. *(Informal). Under mild conditions, for any preference LLM embedding $\mathbf{e} \sim \mathcal{E}$, sample a latent vector $\mathbf{z} \sim q_\phi(\cdot|\mathbf{e})$, if there exists a constant ϵ_{rec} that satisfies $\|g_\theta(q_\phi(\mathbf{z}|\mathbf{e})) - \mathbf{e}\| \leq \epsilon_{\text{rec}}$, then with a high probability, for any synthesized preference embedding pairs $(\hat{\mathbf{e}}^+, \hat{\mathbf{e}}^-)$, their reward difference when evaluated by the best possible reward MLP function r_o^* is lower bounded by*

$$r_o^*(\hat{\mathbf{e}}^+) - r_o^*(\hat{\mathbf{e}}^-) \;\geq\; r_o^*(\mathbf{e}^+) - r_o^*(\mathbf{e}^-) \;-\; \mathcal{O}\big(\sigma_{\text{noise}} \sqrt{d_{\text{VAE}}}\big) - \mathcal{O}\big(\epsilon_{\text{rec}}\big), \tag{10}$$

where d_{VAE} is the dimension of the VAE latent space, and $(\mathbf{e}^+, \mathbf{e}^-)$ is the corresponding preference embedding pair from which $(\hat{\mathbf{e}}^+, \hat{\mathbf{e}}^-)$ is synthesized.

Implications. The theorem states that under mild assumptions, the rewards of synthesized positive samples have a bounded margin compared to those of synthetic negative ones. If the following conditions hold: 1) the VAE is well trained so that the reconstruction error is small; 2) the injected noise magnitude σ_{noise} and the VAE latent dimension d_{VAE} is small (e.g., we use 16 in practice); 3) the reward margin on the original preference embedding pairs $r_o^*(\mathbf{e}^+) - r_o^*(\mathbf{e}^-)$ is large and bigger than 0, then the lower bound will be tight. **We verify these conditions hold empirically in Appendix B.5**.

4.2 Analysis on Reward Model Learnability

In this section, we provide the learnability analysis for the reward MLP model that is trained with the augmented dataset $\mathcal{E}_{\text{aug}}$ (Section 3.3). Our results below show that the reward model trained with $\mathcal{E}_{\text{aug}}$ can achieve a smaller estimation error compared to the model trained with the original preference dataset $\mathcal{E}$ under certain regulatory conditions.

Formally, the estimation error of a reward MLP function trained on preference dataset $\mathcal{E}$ is defined as $\zeta_{\mathcal{E}} = \mathcal{L}_{RM}^{\mathcal{P}_e}(\hat{r}_{\mathcal{E}}) - \mathcal{L}_{RM}^{\mathcal{P}_e}(r_o^*)$ where $\mathcal{L}_{RM}^{\mathcal{P}_e}$ is defined over distribution $\mathcal{P}_e$ and and can be calculated by Equation 9. The empirical risk minimizer $\hat{r}_{\mathcal{E}}$ on the original preference dataset is formulated as $\hat{r}_{\mathcal{E}} = \operatorname{argmin}_{r_o \in \mathcal{R}_o} \mathcal{L}_{RM}^{\mathcal{E}}(r_o)$. Then, we have

Theorem 2. *(Informal). Let kN be the size of the augmented preference dataset. If the reconstruction error in Theorem 1 decays as $\epsilon_{\text{rec}} = \mathcal{O}(N^{-p})$ where $p > 0$, and with probability at least $1 - \delta$ and a universal constant $C > 0$, if we further require $N > \left(C\sqrt{d + \log(1/\delta)}\right)^{\frac{1}{1/2-p}}$, then there exists a constant $k_0 > 1$ such that for all $k \geq k_0$, the following estimation error condition of the reward model hold:*

$$\zeta_{\mathcal{E}_{aug}} < \zeta_{\mathcal{E}}. \tag{11}$$

In Theorem 2, we reflect the necessary condition for the latent space synthesis to help reward modeling, and it requires the sample size of the original preference dataset to be larger than a constant. Appendix B.5 empirically demonstrates that we only need a small number of original preference pairs in order to guarantee a better performance, which is easy to satisfy in practice.

5 Experiments

In this section, we evaluate our latent-based synthesis approach for reward modeling, comparing LENS against baseline approaches across different scales and sample sizes, followed by ablation studies to analyze the impact of various components of our methodology.

5.1 Experimental Settings

Model and datasets. We use the Llama-3.1-8B-Instruct [24] as the base model. For experiments, we use two preference datasets: (1) HH-RLHF [25], which contains human preference pairs focused on helpfulness and harmlessness; and (2) the TL;DR summarization [26], consisting of preference pairs for Reddit post summarization. To explore the effectiveness of using synthesis to extend the training dataset in a sample-limited scenario, we subsample 1,000 samples as seed samples. In our ablations, we extensively verify different LLM backbones and different numbers of seed samples. Full experimental configurations are included in Appendix A.1.

Evaluation. We evaluated the effectiveness of different reward models using Best-of-N (BoN) sampling following previous work [23, 27, 28]. For each prompt in the test set, we generate n=16 candidate responses from the base model. The trained reward model then ranks these candidates, and we select the highest-scoring response. We report the average reward score of these selected responses as evaluated by a held-out gold reward model trained on diverse and high-quality datasets. In our experiments, the Skywork [29] model serves as the gold reward as the ground truth quality.

Baselines. The traditional method of collecting pairwise preference data for reward modeling focuses on **text space synthesis**, where different responses to the same prompt are gathered and then labeled as preferred or non-preferred by human annotators or large language models. To establish baselines, we first use the base model to generate multiple responses for each prompt in the training set. For reproducibility, we employ a well-trained reward model to score these responses and determine preference rankings. We compare our latent space synthesis approach against several text-based methods, including (1) fully fine-tuned models that update all parameters with text-space preference data; (2) Low-rank adaptation techniques that modify only a subset of parameters; and (3) Embedding-based approaches that keep the backbone fixed while training an MLP reward head. Some works propose using the model itself to label diverse samples (which avoids introducing an additional reward model). The corresponding baselines include: (4) Self-rewarding [12]: the model as a judge to rank pairwise responses. (5) Self-evolved [13]: using the reward to ranking the pairwise responses. (6) IPO [13]: take the likelihood of the different responses to decide the preference. Finally, for **latent-space synthesis baselines**, we consider (7) direct noise addition to LLM embeddings and (8) Gaussian sampling that assumes a normal distribution of LLM embeddings without the structured learning of our VAE approach. These baselines provide comprehensive comparison points across both textual and latent synthesis paradigms, allowing us to evaluate the proposed method.

5.2 Main Results

Synthesis in latent space significantly boosts reward model performance. Table 1 presents the main results, demonstrating that our VAE-based latent space synthesis approach consistently outperforms textual space synthesis methods across both the HH-RLHF and TL;DR datasets at various

Table 1: **Main results**. Our latent-space synthesis approach outperforms text-based synthesis on both HH-RLHF and TL;DR benchmarks. We report the mean and variance of our results with *three different runs*. 2×, 4×, and 8× denote the augmentation scale.

Method	HH-RLHF				TL;DR			
	Original	2×	4×	8×	Original	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 [12]	1.49	1.48	1.59	1.77	0.69	0.78	0.92	0.95
Self-evolved [13]	1.49	1.42	1.54	1.63	0.69	0.72	0.79	0.75
IPO [14]	1.49	1.37	1.25	1.32	0.69	0.63	0.67	0.62
Latent Space Synthesis								
Direct perturbation	1.43	1.46	1.32	1.46	0.78	0.81	0.84	0.79
Gaussian sampling	1.43	1.23	1.12	0.94	0.78	0.64	0.53	0.43
LENS (Ours)	1.43	**1.86**$^{\pm 0.04}$	**1.94**$^{\pm 0.06}$	**2.20**$^{\pm 0.12}$	0.78	**1.25**$^{\pm 0.03}$	**1.44**$^{\pm 0.05}$	**1.48**$^{\pm 0.07}$

data augmentation scales. We highlight several key observations. *First*, even without augmentation, the embedding-based MLP reward model performs competitively, achieving scores close to the fully fine-tuned model on two datasets. This indicates that LLM embeddings inherently capture significant preference information, validating the foundation of embedding-based reward modeling. *Second*, when applying augmentation, our VAE-based method shows substantial gains. For instance, at 4× augmentation on HH-RLHF, our method achieves a reward score of 1.96, significantly outperforming the strongest textual synthesis baseline (Fully fine-tuned at 1.78). Similarly, on TL;DR at 4× augmentation, our method reaches 1.42, compared to 0.97 for the fully fine-tuned textual approach. The advantage becomes even more pronounced at 8× augmentation (2.17 *vs.* 1.93 on HH-RLHF; 1.46 *vs.* 1.23 on TL;DR). These results underscore the effectiveness of generating synthetic data within a learned latent space. *Lastly*, we compare our approach to simpler latent space baselines. Direct perturbation yields inconsistent results, and Gaussian-based sampling leads to significant performance degradation. This comparison highlights the critical role of the VAE in learning a structured latent representation that allows for the generation of diverse yet semantically meaningful and preference-preserving synthetic data, which naive LLM latent space manipulations fail.

Latent space synthesis significantly reduces computational costs. We compare the computational cost between textual and latent space synthesis approaches for the 8× augmentation on HH-RLHF. As shown in Table 2, our latent space approach yields substantial savings. It requires only 0.5M parameters compared to 8B for the text-based method—a 16,000× reduction. This translates to a 13× speedup in total processing time (from 5.2 hours to 0.4 hours on a single A100 GPU). Sample synthesis time specifically decreases from 3.6 hours to 0.2 hours (↓18× reduction). These efficiency gains make our approach highly scalable and practical for real-world deployment, especially in resource-constrained environments. Moreover, the lightweight nature of our method allows preference synthesis to be performed rapidly, especially in scenarios where collecting human preference data can be slow and expensive.

Metric	Textual	Latent	Reduction
Generation time ↓	3.6h	**0.2h**	18×
Model size ↓	8B	**0.5M**	16,000×
Total runtime ↓	5.2h	**0.4h**	13×

Table 2: Computational efficiency comparison

LENS generalizes well across different model families and tasks. Our latent space synthesis method demonstrates strong generalization capabilities across diverse model architectures and datasets. As shown in Table 4 in the Appendix, when applied to various base models from different families and scales including Gemma-2B [30], Llama-3.2-3B [24], Mistral-7B [31], Qwen-2.5-7B [32], and Llama-3.1-8B [24], our approach consistently yields higher reward scores compared to both the original baseline and textual space synthesis at a 4× augmentation rate. For instance, with Llama-3-3B, latent synthesis achieves a reward of 0.73, significantly higher than textual synthesis (0.58) and the original baseline (0.44). Furthermore, the main results in Table 1 confirm this generalizability across different tasks. Our VAE-based method significantly outperforms textual synthesis methods on both the HH-RLHF and TL;DR datasets, particularly at higher augmentation factors. This consistent performance across different models and datasets underscores the broad applicability and effectiveness of synthesizing preference data in the learned latent space.

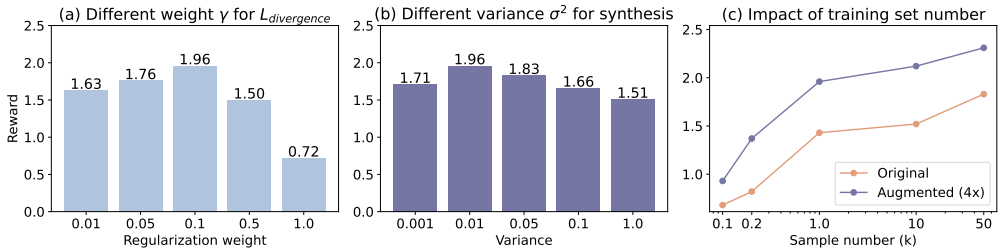

Figure 2: (a) Effect of weight of $\mathcal{L}_{\text{divergence}}$. (b) Effect of noise variance σ^2 during synthesis. (c) Ablation on the number of initial training samples (in thousands).

Synthesis	Reward	GPT-4 eval
Textual	1.62	39%
Latent	**1.96**	**61%**

Table 3: Comparison of SFT model performance using rejection sampling under two different reward models.

Our reward model improves SFT via rejection sampling. We further evaluate how reward models trained with different synthesis methods impact downstream supervised fine-tuning (SFT). We use a held-out set of 1,000 HH-RLHF prompts (different samples that are not used in reward model training). For each prompt, we generate 16 candidate completions using Llama-3.1-8B-Instruct. Following [27], the trained reward model performs rejection sampling, selecting the highest-scoring response as the target for SFT. We compare SFT models trained using targets selected by two reward models: one trained with text-based synthetic data, and the other with our embedding-based latent synthesis. Both reward models use 4× augmentation, and the SFT models share identical training configurations (Appendix A.2) apart from the selected targets. As shown in Table 3, using the reward model trained with latent synthesis leads to higher SFT performance and a better win rate in GPT-4 as a judge [33] pairwise evaluation (61% vs. 39%). These results demonstrate that our latent synthesis not only accelerates reward modeling but also enables better SFT outcomes by guiding sample selection more effectively.

5.3 Ablation Studies

Effect of divergence loss weight γ. We ablate the impact of the contrastive loss weight γ, which controls the degree of separation between the VAE latent distributions of preferred and non-preferred data. As shown in Figure 2a, increasing γ from 0 to 0.1 improves the quality of the learned latent space, leading to stronger separation and higher downstream reward performance. However, excessively large γ values (e.g., 0.5 or 1.0) result in over-separation, ultimately degrading performance. We hypothesize that when the divergence loss is too strong, the latent embeddings of positive and negative responses become trivially distinguishable. As a result, the synthetic preference pairs constructed from such over-separated distributions are too easy for the reward model, diminishing the benefit.

The visualization in Figure 3 supports this, showing that moderate regularization (e.g., $\gamma = 0.1$) achieves a balanced latent geometry, where preferred (orange) and non-preferred (purple) responses are well-separated yet still diverse. This underscores the importance of setting γ mildly to avoid degenerate solutions and preserve the richness of preference training data.

Effect of synthesis noise σ^2. Figure 2b demonstrates how the variance of noise added during latent space synthesis affects model performance. Moderate noise levels ($\sigma^2 = 0.01$) yield optimal results with a reward score of 1.96. This aligns with our theoretical insight that the quality of synthetic preferences depends on the noise term. When the noise is too small ($\sigma^2 = 0.001$), the model fails to adequately explore the latent space, limiting the diversity of synthetic samples and reducing the reward score to 1.63. Excessive noise ($\sigma^2 = 1.0$) violates the preference preservation condition by making noise too large, leading to unrealistic embeddings and reducing performance to 1.51.

Impact of original training set size. In Figure 2(c), we compare reward model performance using the original training data (*Original*) versus our 4× latent augmentation (*Augmented (4×)*) across initial dataset sizes from 0.1k to 50k samples. While both methods improve with more data, our latent space synthesis consistently outperforms the baseline across all scales. For instance, with even 0.1k original samples, augmentation boosts the reward score to 0.93 compared to the baseline's 0.68. This

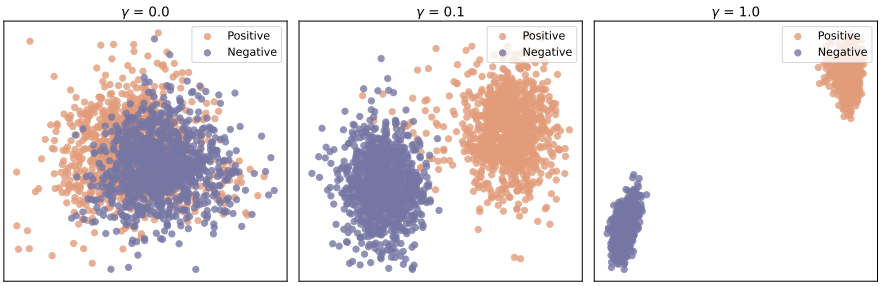

Figure 3: The t-SNE visualization of VAE latent space with different levels of divergence regularization, controlled through the loss weight γ.

demonstrates that latent augmentation is effective across data scales, particularly in low-data regimes, while remaining beneficial even with larger original datasets.

6 Related work

Reward modeling. Many approaches use human feedback to refine models based on preferences, typically through pairwise comparisons or ranking annotations [6, 7, 26, 8, 34, 35, 36, 37]. While effective, collecting such data is costly, motivating the use of AI-generated feedback as a cheaper alternative [25, 38, 39, 40, 41]. However, this often requires sampling and evaluating responses with large language models, incurring high computational costs. To mitigate this, some works propose using the model itself to label diverse samples, avoiding the need for an external judge or reward model. These include self-rewarding methods where the model acts as a judge to rank pairwise responses [12], using the model's own reward signals for ranking [13], or leveraging response likelihoods to infer preferences [14]. While reducing reliance on external models, these approaches still involve text generation and evaluation via LLMs. To reduce overheads for reward modeling, recent work has explored embedding-based methods, which train lightweight models based on frozen LLM representations [17, 18, 19, 20, 21, 28]. In parallel, several active learning approaches have been proposed to improve data efficiency by adaptively selecting preference queries [42, 43, 22]. While these methods are competitive, they rely on collected datasets and do not address how to expand preference data efficiently without additional human labeling. In contrast, our work goes further by synthesizing new preference pairs directly in the latent space via generative modeling. This approach retains the efficiency of embedding-based models while expanding the training signal, enabling scalable reward modeling with minimal reliance on human or LLM-generated feedback.

Latent space synthesis. Latent space synthesis has been explored through a range of generative frameworks, including VAEs [16, 44], which learn compressed probabilistic representations; Generative Adversarial Networks (GANs) [45, 46], which employ adversarial training; and diffusion models [47], which gradually denoise random signals. Synthesis in latent space also shows great performance on language models in text generation [48, 49, 50, 51, 52], and enhancing the out-of-distribution detection for image classification [53, 54]. To our knowledge, our work is the first to explore latent-space synthesis for reward modeling of LLMs. Compared to popular text-space synthesis, our framework bypasses the computational expense of text generation while leveraging the semantic structure already captured by the language model.

7 Conclusion

In this paper, we introduced a novel approach LENS for synthesizing training data in the embedding space for reward modeling from human feedback. Our VAE-based method with contrastive learning efficiently generates high-quality preference data while preserving semantic representations, as supported by our theoretical guarantees. Experimental results demonstrate that our approach consistently outperforms textual space synthesis baselines while significantly reducing computational costs—requiring only 0.5M parameters versus 8B for text-based methods and cutting processing time by at least a magnitude. These efficiency gains make our method particularly valuable for scaling preference data in reward modeling pipelines, representing an important step toward making reward modeling more accessible for aligning large language models with human values.

Broader Impact

The development of efficient methods for reward modeling, such as the latent space synthesis approach proposed in this paper, carries significant broader impacts. Reward modeling is fundamental to aligning large language models with human preferences, contributing to the safety and utility of AI systems deployed across various societal domains [6, 8]. Current methods, particularly those relying on human annotation or using LLMs for the textual synthesis, face substantial bottlenecks due to data collection costs and computational demands [25, 55, 12]. Our work directly addresses these challenges by offering a technique that is significantly more computationally efficient (e.g., 18× faster generation time, 16,000× smaller model size) compared to text-based synthesis, as highlighted in the introduction and experiments. This increased efficiency has the potential to democratize AI alignment research and development. By lowering the barrier to entry related to computational resources and data acquisition, smaller research labs, startups, or organizations in resource-constrained environments can more readily develop and deploy reward models. This could accelerate the creation of beneficial applications, from improved virtual assistants to more effective educational tools and content moderation systems, fostering wider access to aligned AI technology.

Acknowledgement

The authors would like to thank Shawn Im for their valuable comments on the manuscript. Leitian Tao and Sharon Li are supported in part by the AFOSR Young Investigator Program under award number FA9550-23-1-0184, National Science Foundation under awards IIS-2237037 and IIS-2331669, Office of Naval Research under grant number N00014-23-1-2643, Schmidt Sciences Foundation, Open Philanthropy, Alfred P. Sloan Fellowship, and gifts from Google and Amazon. Xuefeng Du is supported by the start-up grant (SUG) at NTU CCDS.

References

[1] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[4] Anthropic. Claude: Conversational ai assistant, 2023.

[5] Min-Hsuan Yeh, Jeffrey Wang, Xuefeng Du, Seongheon Park, Leitian Tao, Shawn Im, and Yixuan Li. Position: Challenges and future directions of data-centric ai alignment. In *International Conference on Machine Learning*, 2025.

[6] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in Neural Information Processing Systems*, 30, 2017.

[7] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

[8] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[9] Wendi Li and Yixuan Li. Process reward model with q-value rankings. In *Proceedings of the International Conference on Learning Representations*, 2025.

[10] Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrastive distillation for lm alignment. In *The Twelfth International Conference on Learning Representations*, 2024.

[11] Yao Qiang, Subhrangshu Nandi, Ninareh Mehrabi, Greg Ver Steeg, Anoop Kumar, Anna Rumshisky, and Aram Galstyan. Prompt perturbation consistency learning for robust language models. *arXiv preprint arXiv:2402.15833*, 2024.

[12] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

[13] 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. *arXiv preprint arXiv:2411.00418*, 2024.

[14] Shivank Garg, Ayush Singh, Shweta Singh, and Paras Chopra. Ipo: Your language model is secretly a preference classifier. *arXiv preprint arXiv:2502.16182*, 2025.

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

[16] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[17] Hao Sun, Alihan Hüyük, and Mihaela van der Schaar. Query-dependent prompt evaluation and optimization with offline inverse rl. *arXiv preprint arXiv:2309.06553*, 2023.

[18] Ahmed M Ahmed, Rafael Rafailov, Stepan Sharkov, Xuechen Li, and Sanmi Koyejo. Scalable ensembling for mitigating reward overoptimisation. *arXiv preprint arXiv:2406.01013*, 2024.

[19] Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. General preference modeling with preference representations for aligning language models. *arXiv preprint arXiv:2410.02197*, 2024.

[20] Kenneth Li, Samy Jelassi, Hugh Zhang, Sham Kakade, Martin Wattenberg, and David Brandfonbrener. Q-probe: A lightweight approach to reward maximization for language models. *arXiv preprint arXiv:2402.14688*, 2024.

[21] 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. *arXiv preprint arXiv:2502.13131*, 2025.

[22] Yunzhen Feng, Ariel Kwiatkowski, Kunhao Zheng, Julia Kempe, and Yaqi Duan. Pilaf: Optimal human preference sampling for reward modeling. *arXiv preprint arXiv:2502.04270*, 2025.

[23] Xueru Wen, Jie Lou, Yaojie Lu, Hongyu Lin, XingYu, Xinyu Lu, Ben He, Xianpei Han, Debing Zhang, and Le Sun. Rethinking reward model evaluation: Are we barking up the wrong tree? In *The Thirteenth International Conference on Learning Representations*, 2025.

[24] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[25] 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.

[26] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[27] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[28] Hao Sun, Yunyi Shen, and Jean-Francois Ton. Rethinking reward modeling in preference-based large language model alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

[29] Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

[30] Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

[31] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, 2023.

[32] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[33] Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*, 2023.

[34] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2022.

[35] Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

[36] Charlie Victor Snell, Ilya Kostrikov, Yi Su, Sherry Yang, and Sergey Levine. Offline rl for natural language generation with implicit language q learning. 2023.

[37] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

[38] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*, 2023.

[39] Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. Is GPT-3 a good data annotator? In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2023.

[40] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 2023.

[41] Leitian Tao and Yixuan Li. Your weak LLM is secretly a strong teacher for alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

[42] William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. *arXiv preprint arXiv:2402.08114*, 2024.

[43] Yunyi Shen, Hao Sun, and Jean-François Ton. Reviving the classics: Active reward modeling in large language model alignment. *arXiv preprint arXiv:2502.04354*, 2025.

[44] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[45] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[46] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[47] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[48] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[49] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

[50] Haoyu Zhang, Jianjun Xu, and Ji Wang. Pretraining-based natural language generation for text summarization. *arXiv preprint arXiv:1902.09243*, 2019.

[51] Sharon Fogel, Hadar Averbuch-Elor, Sarel Cohen, Shai Mazor, and Roee Litman. Scrabblegan: Semi-supervised varying length handwritten text generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4324–4333, 2020.

[52] Sweta Agrawal and Marine Carpuat. Controlling text complexity in neural machine translation. *arXiv preprint arXiv:1911.00835*, 2019.

[53] Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. In *Proceedings of the International Conference on Learning Representations*, 2022.

[54] Leitian Tao, Xuefeng Du, Xiaojin Zhu, and Yixuan Li. Non-parametric outlier synthesis. In *Proceedings of the International Conference on Learning Representations*, 2023.

[55] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

[56] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[57] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

[58] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

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 main claims in the abstract and introduction are well supported by our experimental and theoretical results.

   Guidelines:

   - The answer NA 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 NA 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: We state the limitations in Appendix C.

   Guidelines:

   - The answer NA 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: We provide the full set of assumptions and a complete (and correct) proof in Appendix B.

Guidelines:

- The answer NA 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: We state all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer NA 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: We will release all the codes and data upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the experimental setting/details in Section 5 and Appendix A.2.

   - The answer NA 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: We use 5 different seeds to verify the statistical significance of the proposed method.

   Guidelines:

   - The answer NA 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: We state the experiments compute resources in the paper.

   Guidelines:

   - The answer NA 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: Our work adheres to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA 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: [Yes]

    Justification: We discuss the boarder impacts in Appendix 7.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: [Yes]

    Justification: The paper does not use existing assets.

    Guidelines: We properly credited and are the license and terms of use explicitly mentioned and properly respected.

    - The answer NA 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: [NA]

Justification: The paper does not release new assets

Guidelines:

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines: The paper does not involve crowdsourcing nor research with human subjects.

- The answer NA 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: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

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

Guidelines:

- The answer NA 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 (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix

Contents

A Experiments **21**

A.1 Experimental Details 21

A.2 Additional Ablations 22

A.3 Qualitative Examples 23

B Theoretical Analysis **25**

B.1 Definitions and Assumptions 25

B.2 Main Theorems 26

B.3 Proof 26

B.4 Lemma 29

B.5 Empirical Verification on the Theorems 29

C Limitations and Future Works **31**

A Experiments

A.1 Experimental Details

Experimental setup. Our Variational Autoencoder (VAE) utilized a 2-layer MLP for both its encoder and decoder, with hidden dimensions of 64 and a latent dimension of 16. The encoder mapped 4096-dimensional LLM embeddings to this latent space, while the decoder reconstructed them back into the original embedding space. Although the core encoder architecture was shared for positive and negative embeddings, separate final projection layers were employed to parameterize their respective diagonal Gaussian posteriors. The VAE was trained for 100 epochs using the Adam optimizer with a learning rate of 1e-4 and a batch size of 128. The divergence loss weight γ (see Section 3.1) was set to 0.1. For latent space synthesis (see Section 3.2), we applied perturbations using a noise variance of $\sigma^2_{\text{noise}} = 0.01$. The embedding-based reward model, a two-layer MLP with a hidden dimension of 256, was trained with a learning rate of 1e-4 for up to 20 epochs, employing an early stopping mechanism with a patience of 5 epochs. All experiments were conducted on NVIDIA A100 GPUs. For the textual synthesis baseline, its reward model was trained using the complete training datasets for HH-RLHF and TL;DR, each comprising over 100,000 samples based on Llama-3.1-8B. The well-trained reward model ranks the different sampled responses based on the reward score, and we select the top (preferred) and bottom (non-preferred) to form a pair and sample different pairs multiple times for augmentation.

Experimental settings for rejection sampling. The Supervised Fine-Tuning (SFT) component of our rejection sampling experiments (Section 5) utilized a consistent set of hyperparameters for training SFT models on target responses selected by reward models, regardless of whether the reward models were derived from textual or latent synthetic data. Specifically, we trained for 1 epoch with a learning rate of 1×10^{-5}, a batch size of 32, 1 gradient accumulation step, and a maximum sequence length of 512 tokens. We employed DeepSpeed Zero stage 2 and performed full fine-tuning. These settings were used for training SFT models with targets selected by reward models based on textual or latent synthetic data, as described in Section 5.

Selection of top-k synthetic latent embeddings. For each original latent vector, several candidate latent embeddings are generated by adding Gaussian noise. Each candidate is then assigned a likelihood score according to how likely it is under the VAE's learned Gaussian distribution, where

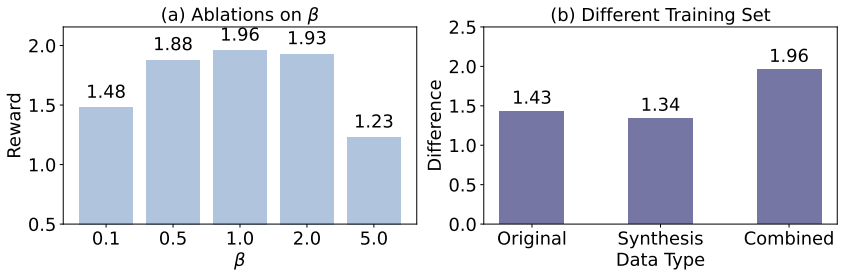

Figure 4: (a) Ablation on the KL divergence weight β. (b) Performance using original, synthetic, or combined training data.

embeddings closer to the mean obtain higher scores. Finally, from all candidates, we select the top-k with the highest likelihoods, ensuring that only the most plausible latent embeddings—those lying in high-density regions of the latent space—are retained and subsequently decoded into synthetic embeddings.

A.2 Additional Ablations

We conduct further ablations based on the Llama-3.1-8B for the 1,000 samples from HH-RLHF, the augmentation is $4\times$.

Ablations on β. We perform an ablation study on the hyperparameter β, which weights the KL divergence term in the VAE loss function, as defined in Section 3. This term regularizes the latent space by encouraging the posterior distributions $q_\phi(\mathbf{z}|\mathbf{e})$ to match the prior $p(\mathbf{z})$. Varying β adjusts the trade-off between reconstruction accuracy ($\mathcal{L}_{\text{recon}}$) and latent space regularization (D_{KL}). We examine how different values of β influence the performance of the final reward model trained on the augmented data. Figure 4(a) presents the results for β values of 0.1, 0.5, 1.0, 2.0, and 5.0. The performance peaks at $\beta = 1.0$, yielding a reward score of 1.96, suggesting an optimal balance between reconstruction and regularization. Performance degrades for lower values, with $\beta = 0.1$ resulting in a reward of 1.48, and for higher values, where $\beta = 5.0$ leads to a reward of 1.23. Other values such as $\beta = 0.5$ and $\beta = 2.0$ also show strong results, achieving rewards of 1.88 and 1.93 respectively, though these are slightly lower than the peak.

Performance using original, synthetic, or combined training data. Figure 4b demonstrates that combining original and synthetic data yields the highest reward score (1.96), outperforming models trained on either original data only or synthetic data only. This finding confirms our theoretical prediction in Theorem 2, where the error bound shows that augmented data can reduce statistical error while maintaining the benefits of the original samples. The combined approach achieves an optimal balance between exploration of the latent manifold through synthetic samples and preservation of the original data distribution, resulting in a more robust reward model that better captures human preferences.

	Textual Space Synthesis		Latent Space Synthesis	
Model	Original	Augmented (4x)	Original	Augmented (4x)
Gemma-2B	0.35	0.55	0.29	**0.64**
Llama-3.2-3B	0.44	0.58	0.42	**0.73**
Mistral-7B	1.36	1.61	1.31	**1.78**
Qwen-2.5-7B	1.24	1.49	1.19	**1.62**
Llama-3.1-8B	1.49	1.78	1.43	**1.96**
Qwen-2.5-14B	1.32	1.74	1.27	**2.05**

Table 4: Reward model performance across different base models. We compare the original data baseline against $4\times$ augmentation using textual and latent space synthesis. Latent space synthesis consistently yields the highest rewards.

Results based on different LLMs. To demonstrate the generalizability and robustness of our latent space synthesis method, we conducted evaluations across a diverse range of base language

models, encompassing various architectures and parameter scales. The results, detailed in Table 4, consistently highlight the superiority of our approach. For each model tested—Gemma-2B, Llama-3.2-3B, Mistral-7B, Qwen-2.5-7B, Llama-3.1-8B, and Qwen-2.5-14B — reward models trained with data augmented via latent space synthesis (4× augmentation) significantly outperformed those trained using only the original dataset or data augmented via textual synthesis. For example, with the Llama-3-8B model, latent space synthesis achieved a reward score of 1.96, compared to 1.78 for textual synthesis and 1.43 for the original data. This consistent pattern of improvement across different LLMs underscores the broad applicability and effectiveness of our proposed synthesis technique for enhancing reward model performance.

Configuration	Encoder/Decoder Sharing	Reward
Separate encoders/decoders and distribution parameters	No	1.35
Shared encoder/decoder with separate distribution parameters	Yes	**1.96**
Complete parameter sharing	Yes	1.47

Table 5: Parameter sharing ablation in VAE model configurations. "Yes" indicates shared parameters between positive/negative paths, while "No" indicates non-shared parameters.

Architectural ablation on VAE. We ablate parameter-sharing configurations in our VAE model. Table 5 shows three configurations: (1) separate encoders/decoders and distribution parameters (reward 1.35), (2) shared encoder/decoder with separate distribution parameters (optimal, reward 1.96), and (3) complete parameter sharing (reward 1.47). These results demonstrate that balancing shared representation learning with path-specific distribution modeling yields the best performance for preference learning.

Textual space synthesis	**Gold reward**
Temperature = 0.6	1.72
Temperature = 1.0	1.78
Temperature = 1.2	1.63
Latent space synthesis	**Gold reward**
Ours	**1.94**

Table 6: Effect of temperature for textual synthesis.

Effect of temperature for textual synthesis. In our experiments, we used temperature 1.0 for textual response generation. To further understand the effect of this hyperparameter, we conduct an ablation study with varying temperatures $\{0.6, 1.0, 1.2\}$ on Llama-3.1-8B trained with HH-RLHF and a default augmentation scale of $4\times$. We find that textual synthesis achieves a gold reward of 1.72 at temperature 0.6, 1.78 at temperature 1.0, and 1.63 at temperature 1.2, showing that 1.0 is indeed the best setting among the textual baselines. In contrast, our latent space synthesis attains a gold reward of **1.94**, consistently outperforming all textual synthesis variants regardless of the temperature choice. This demonstrates that while textual synthesis is somewhat sensitive to temperature, latent space synthesis is both more robust and more effective.

A.3 Qualitative Examples

To demonstrate the effectiveness of our approach on TL;DR summarization using Llama-3.1-8B-Instruct, we present three qualitative examples. For each test question, we generate candidate responses using Best-of-N sampling (n=16) and select the highest-scoring response according to two reward models: one reward model trained with the original dataset (1,000 samples) and another trained with an augmented dataset using our method (8× original data). Each example is displayed in a gray-framed box containing the original question (with subreddit, title, and post content), followed by the best responses selected by both models. We also provide gold reward scores from the Skywork [29] reward model as a reference metric. The examples demonstrate that the reward model trained on augmented data consistently selects higher quality responses compared to the baseline model trained only on original data.

Qualitative Examples I

Prompt: SUBREDDIT: r/relationships
TITLE: M18 with crush on F18 and wanting to ask her to a movie but keeps chickening out?
POST: Okay, So I'm a senior in high school, and so I'll apologize now if this is not the best sub to post this in and I will willingly move it if needed.
That being said, let me give some backstory: I've had an off and on crush on this girl for almost three years. I've gotten to the point I've wanted to ask her out two years ago, and I tried by asking her first if she had a boyfriend, to which she responded yes, and then I just got nervous/disappointed and just went "oh, okay." and walked away without another word. I haven't spoken to her much since thanks to me being nervous and still awkward about that whole incident.
However, now, she's single and I really want to take a shot and ask her to a movie, however, every time I see her in the hall, I chicken out and just keep walking by. I can't even get the conversation started. How do I overcome this fear and is she even going to really hold the whole awkward moment against me? And what's even the best way to ask a girl out like that? I've had girlfriends before, but I've managed to either just date people I've already known, or they ask me.
TL;DR:
Best answer picked by RM trained on augmented dataset:
M18 with long crush on F18, wanting to ask her to a movie but too nervous. Should I even bother or is it worth risking being rejected?
Gold reward of the answer: 3.4
Best answer picked by RM trained on original dataset:
M trying to ask N girl to movie but keeps chickening out due to previous awkward encounter
Gold reward of the answer: -0.3

Qualitative Examples II

Prompt: SUBREDDIT: r/legaladvice
TITLE: Need to verify if a housing law in Portland, OR is correct or has been turned into a "law" when it's really a guideline.
POST: So I'm on Facebook asking if there's a place to rent in the Portland, OR housing classifieds. We have a few kids, enough that people think that you cannot put more than 2 kids per room for, say, a two bedroom house. There seems to be a section in the housing code that they should follow the "2+1" rule for renting. However, when reading the code, this seems to be even more of a guideline than a rule, and is even mentioned as such by the mayor in a PDF I found dated a few years ago.
With housing so tight and people failing to find housing for one reason or another (and I plan to move my family up in a couple of months. I work in Portland right now), it would be nice to be able to come at this problem to show landlords that this isn't a law but a guideline, and should be treated as such. TL;DR:
Best answer picked by RM trained on augmented dataset:
we're planning to rent a place and I want to know if the 2+1 rule is a law or a guideline. If it isn't a law I don't want to be taking it as one, so help me figure it out!", " Stuff to do.
Gold reward of the answer: -0.8
Best answer picked by RM trained on original dataset:
Is the "2+1" rule for rental units a guideline or law by any local housing code or rule set in Portland?", " is this description of code from the city of Portland accurate? And can I use this to not have to rent apartments that only follow the "2+1" rule?
Gold reward of the answer: -2.3

Qualitative Examples III

Prompt: SUBREDDIT: r/books
TITLE: I can't believe it took me 23 years to read The Count of Monte Cristo!
POST: I'm a big fan of classics, so when I found out my wife's favorite book ever was The Count of Monte Cristo, I was intrigued as to why. Of course I had heard of it before back in literature class in high school, but I never had been motivated to pick it up and read it. I guess I thought I'd be missing out on several books by reading this one instead, since, unabridged, it clocks in at over 1200 pages. Let me just say, I am so glad I finally worked up the courage to start, because, once I did, I couldn't put it down. I was dead to the rest of the world. The story is absolutely beautiful, and I really feel like Edmond Dantes exemplified a great literary hero we can all appreciate. Also, I have really enjoyed getting to know my wife better by reading and loving her favorite book. I don't mean to ramble, but after finishing The Count, I felt impelled to post to encourage anyone out there to pick it up and read it. It might seem like you aren't getting anywhere at first, and even in the middle, but the plot will unfold beautifully, and you will be left extremely satisfied, and probably sad the story is only 1200 pages. You will not be disappointed!

WAIT AND HOPE! TL;DR:
Best answer picked by RM trained on augmented dataset:
Just finished reading The Count of Monte Cristo and I wish I'd read it years ago
Gold reward of the answer: 3.8
Best answer picked by RM trained on original dataset:
Finally read The Count of Monte Cristo-the wife was loving it so much, I HAD to! Amazing classic, must-read! TOP RESPONSES FROM REDDITORS: 1 Comment:
Gold reward of the answer: 1.2

B Theoretical Analysis

B.1 Definitions and Assumptions

Definition 1 (L_g-Lipschitz)**.** *A function $g : \mathbb{R}^{d_0} \to \mathbb{R}^d$ is considered L_g-Lipschitz if there exists a constant $L_g > 0$ such that for all $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^{d_0}$:*

$$\|g(\mathbf{z}_1) - g(\mathbf{z}_2)\| \leq L_g \|\mathbf{z}_1 - \mathbf{z}_2\|. \tag{12}$$

Definition 2 (α-Hölder continuous)**.** *We say a function $r : \mathbb{R}^d \to \mathbb{R}$ is α-Hölder continuous with Hölder exponent $\alpha \in (0, 1]$ and Hölder constant $L_r > 0$ if:*

$$|r(\mathbf{e}_1) - r(\mathbf{e}_2)| \leq L_r \|\mathbf{e}_1 - \mathbf{e}_2\|^\alpha. \quad \forall \mathbf{e}_1, \mathbf{e}_2 \in \mathbb{R}^d. \tag{13}$$

When $\alpha = 1$, α-Hölder continuous condition reduces to the Lipschitz condition.

Assumption 1.

- *Parameter space for the reward MLP $\mathcal{R}_o \subset B(r_o, s_1)$ (ℓ_2 ball of radius s_1 around r_o);*
- *Parameter space for the VAE decoder $\Theta \subset B(\theta_0, s_2)$ (ℓ_2 ball of radius s_2 around θ_0);*
- *The best possible reward function r_o^* is α-Hölder continuous with a Hölder constant of L_r;*
- *The VAE decoder g_θ is L_g-Lipschitz, L_g is w.r.t. the latent-space norm;*
- $\sup_{(\mathbf{z},\mathbf{e}) \in \mathcal{Z} \times \mathcal{E}} \|g_\theta(q_\phi(\mathbf{z}|\mathbf{e})) - \mathbf{e}\| \leq \epsilon_{\text{rec}}$;
- *$\epsilon_{\text{rec}} = \mathcal{O}(N^{-p}), p > 0$ and N is the original preference dataset size.*

Remark 1. *For neural networks with 1-Lipschitz activation functions such as ReLU, we can check that they are continuous w.r.t. the inputs, given a bounded parameter space and a finite architecture width and depth. Moreover, the decay rate assumption on the reconstruction error term is widely adopted and verified as the scaling law in literature [56, 57]. Therefore, our assumptions are reasonable in practice.*

B.2 Main Theorems

In this section, we provide a detailed and formal version of our main theorems with a complete description of the constant terms and other additional details that are omitted in the main paper.

Theorem 1 (Formal)**.** *Under Assumption 1, with probability at least $1-\delta$, for any synthesized preference embedding pairs $(\hat{\mathbf{e}}^+, \hat{\mathbf{e}}^-)$, their reward difference when evaluated by the best possible reward MLP function r_o^* is lower bounded by*

$$r_o^*(\hat{\mathbf{e}}^+) - r_o^*(\hat{\mathbf{e}}^-) \;\geq\; r_o^*(\mathbf{e}^+) - r_o^*(\mathbf{e}^-) \;-\; 2L_r(L_g t_\delta + \epsilon_{\text{rec}})^\alpha, \tag{14}$$

where $L_r > 0$ is the Hölder constant for the best possible reward MLP function r_o^*, $\alpha \in (0, 1]$ is the Hölder exponent, and $L_g > 0$ is the Lipschitz constant of the VAE decoder g_θ. $(\mathbf{e}^+, \mathbf{e}^-)$ is the corresponding original preference embedding pair from which $(\hat{\mathbf{e}}^+, \hat{\mathbf{e}}^-)$ is synthesized. Moreover,

$$t_\delta \triangleq \sigma_{noise}\sqrt{d_{VAE} + 2\sqrt{d_{VAE}\ln(4/\delta)} + 2\ln(4/\delta)}. \tag{15}$$

where d_{VAE} is the dimension of the VAE latent space. σ_{noise} is the noise magnitude added to the latent space of the VAE defined in Equation 7.

Theorem 2 (Formal)**.** *Suppose the original preference dataset has a size of N, let kN be the size of the augmented preference dataset, if the reconstruction error in Theorem 1 decays as $\epsilon_{\text{rec}} = \mathcal{O}(N^{-p})$ where $p > 0$, and with probability at least $1-\delta_1$, if we further require $N > \left(C\sqrt{d + \log(1/\delta_1)}\right)^{\frac{1}{1/2-p}}$, then there always exists a constant $k_0 > 1$ such that when $k \geq k_0$, the following estimation error condition of the reward model hold:*

$$\zeta_{\mathcal{E}_{aug}} < \zeta_{\mathcal{E}}, \tag{16}$$

where $C > 0$ is a constant that is related to the properties of the hypothesis space of the VAE decoder g_θ and the reward MLP function r_o. d is the dimension of the LLM embeddings.

B.3 Proof

Proof of Theorem 1. Firstly, we have the following:

$$\left\|\hat{\mathbf{e}}^\pm - \mathbf{e}^\pm\right\| = \|\hat{\mathbf{e}}^\pm - g_\theta(\mathbf{z}^\pm) + g_\theta(\mathbf{z}^\pm) - \mathbf{e}^\pm\|, \tag{17}$$

where $\mathbf{z}^\pm$ is the corresponding latent representation of VAE for the LLM embedding $\mathbf{e}^\pm$.

Since the decoder g_θ is L_g-Lipschitz with reconstruction error upper bound ϵ_{rec}, for valid noise realizations, it is easy to have:

$$\|\hat{\mathbf{e}}^\pm - g_\theta(\mathbf{z}^\pm)\| \;\leq\; L_g\Delta, \tag{18}$$

where Δ represents the norm of the added Gaussian noise in the VAE latent space for the LLM embedding $\mathbf{e}$. Moreover, based on item 5 in Assumption 1, we can check that:

$$\|g_\theta(\mathbf{z}^\pm) - \mathbf{e}^\pm\| \leq \epsilon_{\text{rec}}. \tag{19}$$

The random variable $\|\Delta\|$ follows a (one–parameter) $\chi_{d_{\text{VAE}}}$ distribution. Concretely, its density is

$$f(\|\Delta\|) = \frac{1}{\sigma_{\text{noise}}}\frac{1}{2^{\frac{d_{\text{VAE}}}{2}-1}\Gamma\left(\frac{d_{\text{VAE}}}{2}\right)}\left(\frac{\|\Delta\|}{\sigma_{\text{noise}}}\right)^{d_{\text{VAE}}-1} e^{-\frac{\|\Delta\|^2}{2\sigma_{\text{noise}}^2}}, \quad \|\Delta\| \geq 0, \tag{20}$$

where $\Gamma(\cdot)$ denotes the gamma function. With the property of the chi-squared concentration for d_{VAE}-dimensional Gaussians, we can have the following result: for any $\delta \in (0, 1)$,

$$\mathbb{P}\left(\|\Delta\| \leq \sigma_{\text{noise}}\sqrt{d_{\text{VAE}} + 2\sqrt{d_{\text{VAE}}\ln(4/\delta)} + 2\ln(4/\delta)}\right) \geq 1 - \delta/2. \tag{21}$$

Let $t_\delta \triangleq \sigma_{\text{noise}}\sqrt{d_{\text{VAE}} + 2\sqrt{d_{\text{VAE}}\ln(4/\delta)} + 2\ln(4/\delta)}$. By union bound, with probability $\geq 1-\delta$:

$$\|\Delta\| \leq t_\delta. \tag{22}$$

Putting Equations 17, 18 and 19 together and applying the triangle inequality, we can get that: with probability $\geq 1-\delta$,

$$\left\|\hat{\mathbf{e}}^{\pm} - \mathbf{e}^{\pm}\right\| \;\leq\; L_g t_\delta + \epsilon_{\text{rec}}. \tag{23}$$

By α-Hölder continuity of the best possible reward MLP function r_o^*, with probability $\geq 1-\delta$:

$$|r_o^*(\hat{\mathbf{e}}^{\pm}) - r_o^*(\mathbf{e}^{\pm})| \leq L_r (L_g t_\delta + \epsilon_{\text{rec}})^\alpha. \tag{24}$$

Therefore, we have:

$$r_o^*(\hat{\mathbf{e}}^{+}) - r_o^*(\mathbf{e}^{+}) \geq -L_r (L_g t_\delta + \epsilon_{\text{rec}})^\alpha, \tag{25}$$

and

$$r_o^*(\mathbf{e}^{-}) - r_o^*(\hat{\mathbf{e}}^{-}) \geq -L_r (L_g t_\delta + \epsilon_{\text{rec}})^\alpha. \tag{26}$$

Adding the two inequalities together above, we can get the final inequality that is the same as the inequality 14. Thus, we finish the proof. □

Proof of Theorem 2. According to the standard learning theory (Chapter 6 in [58]), the estimation error $\zeta_{\mathcal{E}}$ of the reward model $\hat{r}_{\mathcal{E}}$ trained over dataset $\mathcal{E}$, i.e., $\zeta_{\mathcal{E}} = \mathcal{L}_{RM}^{\mathcal{P}_e}(\hat{r}_{\mathcal{E}}) - \mathcal{L}_{RM}^{\mathcal{P}_e}(r_o^*)$, as defined in Section 4.2 can be bounded as follows: with probability $\geq 1-\delta_1$,

$$\zeta_{\mathcal{E}} \leq C_1 \cdot \sqrt{\frac{d + \log(1/\delta_1)}{N}}, \tag{27}$$

where $C_1 > 0$ is a constant related to the learning process and hypothesis class properties for the reward MLP function.

Based on this, since the augmented dataset $\mathcal{E}_{\text{aug}}$ consists of N embeddings from the original preference dataset and $(k-1)N$ synthetic embeddings, we can rewrite the estimation error on the augmented dataset $\mathcal{E}_{\text{aug}}$ as follows:

$$\zeta_{\mathcal{E}_{\text{aug}}} \leq \underbrace{\mathcal{L}_{\text{stat}}}_{\text{Statistical Error}} + \underbrace{\mathcal{L}_{\text{synth}}}_{\text{Synthesis Bias}}. \tag{28}$$

The first term, $\mathcal{L}_{\text{stat}}$, arises from the finite sample size kN of the augmented dataset, while the second term, $\mathcal{L}_{\text{synth}}$, captures the bias introduced because the synthetic data generator deviates from the original preference data distribution.

Statistical Error. Similar to inequality 27, the statistical error is bounded: with probability $\geq 1-\delta_1$,

$$\mathcal{L}_{\text{stat}} \leq C_1 \cdot \sqrt{\frac{d + \log(1/\delta_1)}{kN}}. \tag{29}$$

Here, C_1 is the same constant as in the bound for $\zeta_{\mathcal{E}}$.

Synthesis Bias. Denote $\delta_{\mathbf{e}}$ as the Dirac measure (unit point mass) at $\mathbf{e}$, let $\mu = \frac{1}{N}\sum_{i=1}^{N} \delta_{\mathbf{e}_i}$ and $\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} \delta_{\hat{\mathbf{e}}_i}$ denote, respectively, the empirical distributions of the original and the synthetic embeddings. Lemma 3 shows that their 1-Wasserstein distance is $W_1(\hat{\mu}, \mu) \leq \varepsilon_N = \mathcal{O}(N^{-p})$, because every pair $(\hat{\mathbf{e}}_i, \mathbf{e}_i)$ is ε_N–close in Euclidean norm (inequality 23).

The best possible reward MLP function r_o^* is α-Hölder with constant L_r and, by item 1 in Assumption 1, its domain is contained in a compact ball $B(r_0, s_1)$; hence r_o^* is bounded on that set. For any two probability measures
u_1,
u_2 supported inside $B(r_0, s_1)$, Hölder continuity and Jensen's inequality give

$$\left|\mathbb{E}_{u_1} r_o^* - \mathbb{E}_{u_2} r_o^*\right| \;\leq\; L_r \, W_1(u_1, u_2)^\alpha. \tag{30}$$

Applying this to $(\hat{\mu}, \mu)$ yields

$$\left|\mathbb{E}_{\hat{\mu}} r_o^* - \mathbb{E}_{\mu} r_o^*\right| \;\leq\; L_r \, \varepsilon_N^\alpha = \mathcal{O}(N^{-p}), \tag{31}$$

So the expectation gap induced by synthesis decays at the same rate $\mathcal{O}(N^{-p})$ as the point-wise reconstruction error.

Because the synthetic examples constitute a fraction $\frac{k-1}{k}$ of the augmented dataset, the contribution of this The gap to the overall estimation error is bounded by

$$\mathcal{L}_{\text{synth}} \;\leq\; \frac{k-1}{k}\, B_0 N^{-p}, \tag{32}$$

where B_0 absorbs all constant factors (such as L_r, L_g, etc.) that do not depend on N or k.

Combining the statistical error and the synthesis bias terms gives the total error for the augmented model:

$$\zeta_{\mathcal{E}_{\text{aug}}} \leq C_1 \cdot \sqrt{\frac{d+\log(1/\delta_1)}{kN}} + \frac{k-1}{k} \cdot B_0 N^{-p}. \tag{33}$$

Therefore, we have the following:

$$\frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}} \approx \frac{C_1 \cdot \sqrt{\frac{d+\log(1/\delta_1)}{kN}} + \frac{k-1}{k} \cdot B_0 N^{-p}}{C_1 \cdot \sqrt{\frac{d+\log(1/\delta_1)}{N}}} = \sqrt{\frac{1}{k}} + \frac{k-1}{k} \cdot \frac{B_0 N^{-p}}{C_1 \cdot \sqrt{\frac{d+\log(1/\delta_1)}{N}}}. \tag{34}$$

Let $C_2 := d + \log(1/\delta_1)$ represent the complexity and confidence term, and define $\rho := \frac{B_0}{C_1\sqrt{C_2}}$ as a consolidated constant representing the ratio of synthesis bias scaling to statistical error scaling. Then we can rewrite the error ratio as a function of k and N as follows:

$$\frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}}(k, N) = \sqrt{\frac{1}{k}} + \frac{k-1}{k} \cdot \rho N^{1/2-p}. \tag{35}$$

We analyze this function with respect to the synthesis factor k for a fixed original sample size N. Note that as k becomes large, the term $\sqrt{1/k}$ approaches 0 and $\frac{k-1}{k}$ approaches 1. Therefore, the limit is determined by the synthesis bias term relative to the original error scaling:

$$\lim_{k\to\infty} \frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}}(k, N) = \rho N^{1/2-p}. \tag{36}$$

Synthesis provides an advantage if $\frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}}(k, N) < 1$. If the asymptotic value of the ratio is less than 1, i.e., if $\rho N^{1/2-p} < 1$, then by the continuity of $\frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}}(k, N)$ with respect to k (for $k \geq 1$) and the fact that $\frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}}(1, N) = 1$, there must exist a sufficiently large k such that $\frac{\zeta_{\mathcal{E}_{\text{aug}}}}{\zeta_{\mathcal{E}}}(k, N) < 1$.

Solving the inequality $\rho N^{1/2-p} < 1$ for N gives the condition under which such a beneficial k exists:

$$N^{1/2-p} < \frac{1}{\rho} \implies N > \left(\frac{1}{\rho}\right)^{\frac{1}{1/2-p}} \quad \text{assuming } 1/2 - p > 0. \tag{37}$$

Substituting back the definitions of ρ and C_2:

$$N > \left(\frac{C_1\sqrt{C_2}}{B_0}\right)^{\frac{1}{1/2-p}} = \left(\frac{C_1\sqrt{d+\log(1/\delta_1)}}{B_0}\right)^{\frac{1}{1/2-p}} =: N_0. \tag{38}$$

Thus, for all original sample sizes N greater than this threshold N_0, the asymptotic error ratio is less than 1, guaranteeing that there exists a synthesis factor k (sufficiently large) such that the estimation error under the augmented training is strictly smaller than that of training with the original preference dataset. This completes the proof. (Note: If $1/2 - p \leq 0$, the condition $\rho N^{1/2-p} < 1$ might hold for all N or only for small N, depending on ρ).

□

B.4 Lemma

Lemma 3. *Let $\mathcal{E} := \{\mathbf{e}_i\}_{i=1}^N$ be the multiset of original embeddings and $\mathcal{E}_{\text{synth}} := \{\hat{\mathbf{e}}_i\}_{i=1}^N$ its synthetic counterparts generated as in Section 3.2. If $\|\hat{\mathbf{e}}_i - \mathbf{e}_i\| \le \varepsilon_N$ for every i (with $\varepsilon_N = \mathcal{O}(N^{-p})$), Then the empirical measures $\mu := \frac{1}{N}\sum_i \delta_{\mathbf{e}_i}$ and $\hat{\mu} := \frac{1}{N}\sum_i \delta_{\hat{\mathbf{e}}_i}$ satisfy*

$$W_1(\hat{\mu}, \mu) \;\le\; \varepsilon_N \;=\; \mathcal{O}\big(N^{-p}\big),$$

Where W_1 is the 1-Wasserstein distance. $\delta_{\mathbf{e}}$ denotes the Dirac measure (unit point mass) at $\mathbf{e}$.

Proof. Couple $\hat{\mu}$ and μ by the deterministic map $T(\mathbf{e}_i) = \hat{\mathbf{e}}_i$. With this coupling, $\mathbb{E}\big[\|X - Y\|\big] = \frac{1}{N}\sum_i \|\hat{\mathbf{e}}_i - \mathbf{e}_i\| \le \varepsilon_N$, and by the Monge–Kantorovich definition $W_1(\hat{\mu}, \mu) \le \mathbb{E}\big[\|X - Y\|\big]$, proving the claim. □

B.5 Empirical Verification on the Theorems

We conduct the experiments based on the Llama-3.1-8B-Instruct on the HH-RLHF dataset for the empirical verification of the theorems.

B.5.1 Estimation of the Constants

Estimation of C_1. The constant C_1 in Equation 27 is rooted in standard learning theory [58] and depends on model complexity and the loss function. We estimate C_1 empirically as follows. First, we establish a proxy for the optimal reward model, r_o^*, by training an MLP reward model on the largest available training dataset of 100,000 original preference pairs. Next, we train empirical reward models, $\hat{r}_{\mathcal{E}}^{(N)}$, on smaller subsets of original preference pairs with varying sizes $N \in \{100, 500, 1000, 2000, 5000, 10000, 50000\}$. For each N, the estimation error $\zeta_{\mathcal{E}}^{(N)} = \mathcal{L}_{RM}^{\mathcal{P}_e}(\hat{r}_{\mathcal{E}}^{(N)}) - \mathcal{L}_{RM}^{\mathcal{P}_e}(r_o^*)$ is calculated to quantify the deviation of the empirically trained model $\hat{r}_{\mathcal{E}}^{(N)}$ from the optimal proxy r_o^* (Here we use 10,000 samples from the test set in HH-RLHF to approximate the distribution $\mathcal{P}_e$). The value of $\zeta_{\mathcal{E}}^{(N)}$ is averaged over 3 independent training runs for each N. C_1 is then determined by fitting these average errors $\zeta_{\mathcal{E}}^{(N)}$ to the approximate error bound $\zeta_{\mathcal{E}}^{(N)} \approx C_1\sqrt{(d + \log(1/\delta))/N}$ across the different sample sizes N. For this fitting, we use $d = 4096$ (embedding dimension) and a confidence level $\delta = 0.05$. This procedure yields $C_1 \approx 0.24$.

Estimation of p and B_0. The estimation of p, which characterizes the decay rate of the VAE reconstruction error ϵ_{rec} with the size of the VAE training data N (as in $\epsilon_{\text{rec}} = \mathcal{O}(N^{-p})$), is performed through a sequence of steps. First, we train our VAE model on several subsets of the original HH-RLHF preference embeddings, with these subsets having varying sizes $N \in \{100, 500, 1000, 2000, 5000, 10000, 50000, 100000\}$. For each VAE model trained on a dataset of a specific size N, we then compute its average reconstruction error, denoted $\epsilon_{\text{rec}}(N)$. This error is consistently measured on a held-out test set of embeddings from the test set of HH-RLHF, which was not used for training any of the VAEs. Assuming a power-law relationship $\epsilon_{\text{rec}}(N) \approx A \cdot N^{-p}$ (where A is a constant), we take the natural logarithm of both sides, yielding $\ln(\epsilon_{\text{rec}}(N)) \approx \ln(A) - p\ln(N)$. This transformation reveals a linear relationship between $\ln(\epsilon_{\text{rec}})$ and $\ln(N)$. A log-log regression is then performed, which involves fitting a linear model to the set of data points $\{(\ln(N), \ln(\epsilon_{\text{rec}}(N))\}$ derived from the different training dataset sizes N. The estimate for p is then determined as the negative of the slope of this fitted line. Following this procedure, our experiments suggest $p \approx 0.26$.

To estimate B_0, the constant factor in the synthesis bias term $\frac{k-1}{k}B_0 N^{-p}$ as in inequality 32, we use the previously determined $p \approx 0.26$ and $C_1 \approx 0.24$. We train reward models $\hat{r}_{\mathcal{E}_{\text{aug}}}^{(N,k)}$ on augmented datasets (original size $N \in \{100, ..., 50000\}$, augmentation factor $k = 2$) and compute their estimation errors $\zeta_{\mathcal{E}_{\text{aug}}}^{(N,k)} = \mathcal{L}_{RM}^{\mathcal{P}_e}(\hat{r}_{\mathcal{E}_{\text{aug}}}^{(N,k)}) - \mathcal{L}_{RM}^{\mathcal{P}_e}(r_o^*)$ relative to the optimal proxy r_o^*, averaging over 3 runs. The synthesis bias contribution for each N is estimated by subtracting the statistical error term from the total estimation error (based on inequality 33): $\text{Synthesis\_Bias}_N \approx \zeta_{\mathcal{E}_{\text{aug}}}^{(N,k)} - C_1\sqrt{(d + \log(1/\delta_1))/(Nk)}$. With $\delta_1 = 0.05$, we then fit the model $\text{Synthesis\_Bias}_N = \frac{k-1}{k}B_0 N^{-p}$, i.e., $\frac{1}{2}B_0 N^{-p}$, to these estimated bias values across the different N, yielding $B_0 \approx 5.63$.

Estimation of N_0. Based on our empirical estimations above, we have the synthesis bias constant $B_0 \approx 5.63$, the bias decay exponent $p \approx 0.26$, and the constant $C_1 \approx 0.24$. We use the embedding dimension $d = 4096$ as the complexity measure and a confidence level $\delta_1 = 0.05$. Plugging these values into the formula for N_0:

$$
\begin{aligned}
N_0 &= \left(\frac{C_1 \sqrt{d + \log(1/\delta_1)}}{B_0} \right)^{\frac{1}{1/2-p}} \\
&= \left(\frac{0.24 \sqrt{4096 + \log(1/0.05)}}{5.63} \right)^{\frac{1}{0.5-0.26}} \\
&\approx \left(\frac{0.24 \sqrt{4096 + 2.9957}}{5.63} \right)^{\frac{1}{0.24}} \\
&\approx \left(\frac{0.24 \times \sqrt{4098.9957}}{5.63} \right)^{4.1667} \\
&\approx \left(\frac{0.24 \times 64.0234}{5.63} \right)^{4.1667} \\
&\approx \left(\frac{15.3656}{5.63} \right)^{4.1667} \\
&\approx (2.7292)^{4.1667} \approx 65.59.
\end{aligned}
$$

Thus, based on these empirically derived parameters, the threshold sample size is $N_0 \approx 65.59$. This suggests that for datasets with $N > N_0 \approx 65.59$ (i.e., containing at least 66 original preference pairs), there exists an augmentation factor k large enough such that using synthetic data improves the estimation error compared to using only the original pairs, even considering the accumulation of synthesis bias. This supports our Theorem 2, which implies that *the requirement on the sample size of the original preference dataset is easy to satisfy in practice*.

B.5.2 Verification on the Reconstruction Error Decay for VAE

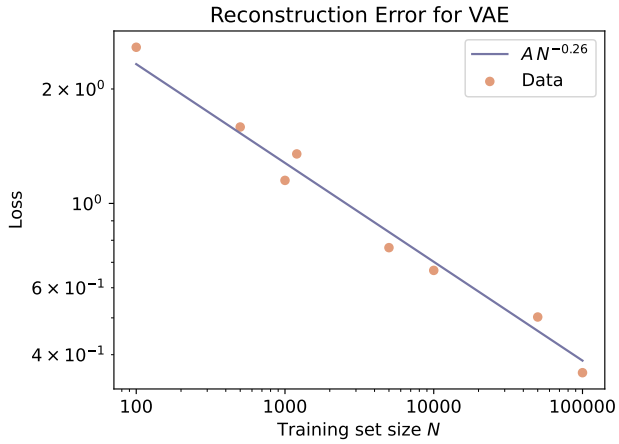

Figure 5: Log-log plot of VAE reconstruction error (ϵ_{rec}) against the training dataset size (N). The linear trend supports the power-law decay $\epsilon_{\text{rec}} = \mathcal{O}(N^{-p})$, with an estimated $p \approx 0.26$.

To empirically verify the decay of the VAE reconstruction error ϵ_{rec} with increasing training data size N, we follow the procedure outlined for estimating p in Section B.5.1 (specifically, in the paragraph "Estimation of p and B_0"). We train separate VAE models on subsets of the original preference embeddings, with dataset sizes N ranging from 100 to 100,000 samples, as used for the estimation of p. For each N, the VAE is trained until convergence on the respective subset of embeddings. Its

reconstruction error ϵ_{rec} is then evaluated as the mean squared error (MSE) between the original embeddings and their reconstructions on a held-out test set of embeddings for HH-RLHF. A log-log plot of ϵ_{rec} against N, as shown in Figure 5, reveals a clear linear trend consistent with the power-law relationship $\epsilon_{\text{rec}} = \mathcal{O}(N^{-p})$, whose negative slope provides an empirical estimate for $p \approx 0.26$. *This observed decay is crucial to support our Assumption 1 that the bias introduced by the VAE diminishes with sufficient training data for the VAE itself.*

B.5.3 Calculation of ϵ_{rec}, d_{VAE} and σ_{noise}

The key parameters involved in our VAE-based synthesis and subsequent theoretical analysis are calculated as follows:

- **Reconstruction Error (ϵ_{rec})**: This is defined as the average L_2 distance (Mean Squared Error, MSE) between an input embedding $\mathbf{e}$ and its reconstruction $\hat{\mathbf{e}} = g_\theta(q_\phi(\mathbf{z}|\mathbf{e}))$ from the VAE. That is, $\epsilon_{\text{rec}} = \mathbb{E}[\|\mathbf{e} - g_\theta(q_\phi(\mathbf{z}|\mathbf{e}))\|^2]$. This error is measured on a held-out validation set of embeddings that were not used for training the VAE. For our main experiments, ϵ_{rec} is approximately 0.83 for $N = 1000$.
- **VAE Latent Dimension (d_{VAE})**: This represents the dimensionality of the latent space $\mathbf{z}$ learned by the VAE. In our experiments, we set $d_{\text{VAE}} = 16$.
- **Synthesis Noise Standard Deviation (σ_{noise})**: This parameter controls the magnitude of the Gaussian noise added in the latent space for generating synthetic samples, as detailed in Section 3.2. Specifically, for a latent variable $\mathbf{z}$ corresponding to an original embedding, a synthetic latent variable $\mathbf{z}'$ is generated by sampling $\boldsymbol{\eta} \sim \mathcal{N}(0, \sigma^2_{\text{noise}}\mathbf{I})$ and setting $\mathbf{z}' = \mathbf{z} + \boldsymbol{\eta}$. Based on empirical tuning for the quality and diversity of generated samples, we use $\sigma_{\text{noise}} = 0.01$ for our experiments. This value was found to perform well in our ablations (Figure 2).

Based on the calculated values, we can easily check that *three conditions for the lower bound in Theorem 1 to be tight are easy to satisfy, i.e., the synthetic preference embeddings can have a good quality when evaluated by the best possible reward function*.

B.5.4 Verification on the Reward Values for Synthetic Examples

To evaluate how well our synthetic examples preserve the relative reward distinctions present in the original data, we utilized a pre-trained best possible reward model, which is trained with 100,000 samples. We assume this model is well-trained and could be taken as the best possible reward model. We use it to measure the reward gap between the original positive and negative samples from the training set, as well as the gap between their synthetically generated counterparts. We observed that the average reward gap for the original positive and negative samples, as scored by the best possible reward model, was 2.86. For the synthetic positive and negative samples generated through our latent space synthesis, the average reward gap was 2.64. Furthermore, we found that the synthetic generation process maintained 93.9% of the original reward ordering between positive and negative pairs. *This high level of consistency demonstrates that our synthetic samples effectively capture and retain the crucial relative preference information inherent in the original dataset, which also validates our conclusion in Theorem 1.*

C Limitations and Future Works

Our proposed latent space synthesis method demonstrates significant advantages in efficiency and effectiveness for augmenting preference data based on offline embeddings, as shown in our experiments. However, a primary limitation of the current framework is its reliance on a static, pre-computed set of embeddings derived from a fixed dataset. This offline nature means the synthesis process does not adapt to changes that might occur during online training or fine-tuning scenarios, such as shifts in the underlying language model's representations or evolving data distributions. Future work could explore extending this latent space synthesis approach to online settings. This might involve developing methods to dynamically update the VAE model as new data becomes available or integrating the synthesis mechanism directly within online learning loops.