# Latent Preference Coding: Aligning Large Language Models via Discrete Latent Codes

**Zhuocheng Gong** [1] **Jian Guan** [2] **Wei Wu** [2] **Huishuai Zhang** [1] **Dongyan Zhao** [1 3]

## Abstract

Large language models (LLMs) have achieved remarkable success, yet aligning their generations with human preferences remains a critical challenge. Existing approaches to preference modeling often rely on an explicit or implicit reward function, overlooking the intricate and multifaceted nature of human preferences that may encompass conflicting factors across diverse tasks and populations. To address this limitation, we introduce Latent Preference Coding (LPC), a novel framework that models the implicit factors as well as their combinations behind holistic preferences using discrete latent codes. LPC seamlessly integrates with various offline alignment algorithms, automatically inferring the underlying factors and their importance from data without relying on pre-defined reward functions and hand-crafted combination weights. Extensive experiments on multiple benchmarks demonstrate that LPC consistently improves upon three alignment algorithms (DPO, SimPO, and IPO) using three base models (Mistral-7B, Llama3-8B, and Llama3-8B-Instruct). Furthermore, deeper analysis reveals that the learned latent codes effectively capture the differences in the distribution of human preferences and significantly enhance the robustness of alignment against noise in data. By providing a unified representation for the multifarious preference factors, LPC paves the way towards developing more robust and versatile alignment techniques for the responsible deployment of powerful LLMs.

[1]Wangxuan Institute of Computer Technology, Peking University [2]Ant Group [3]National Key Laboratory of General Artificial Intelligence. Correspondence to: Wei Wu <congyue.ww@antgroup.com>, Dongyan Zhao <zhaody@pku.edu.cn>.

*Proceedings of the $42^{nd}$ International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

## 1. Introduction

Alignment has emerged as a key step in the development of large language models (LLMs) (Ouyang et al., 2022; Bai et al., 2022; Touvron et al., 2023; Dubey et al., 2024). The goal of alignment is to leverage human feedback to gauge the generative distributions of LLMs, steering their outputs to be helpful, honest, and harmless (Askell et al., 2021). To this end, human annotators are tasked with expressing preferences among human-curated or machine-generated texts, and these preference annotations serve as supervision signals to further optimize LLMs. Amid the surge in alignment research, significant attention has been focused on optimization objectives, considering both online (Schulman et al., 2017; Munos et al., 2023; Calandriello et al., 2024; Yang et al., 2024b) and offline (Rafailov et al., 2024; Zhao et al., 2023; Azar et al., 2024; Meng et al., 2024; Tang et al., 2024) environments as well as different types of preference annotations, such as scalar ratings (Richemond et al., 2024) and pairwise rankings (Rafailov et al., 2024). Optimization with the well-designed objectives has been widely validated for effectively reducing toxicity (Dai et al., 2024) while significantly improving the truthfulness and coherence of LLM outputs (Touvron et al., 2023). In this work, we study the alignment of LLMs from a different perspective: *Can we exploit the supervision signals more effectively through fine-grained modeling of complex human preference?*

The common practice for preference modeling typically involves estimating a single reward function from human annotations as a proxy for human preference (Schulman et al., 2017; Gulcehre et al., 2023). Recently, human annotations have also been used directly as supervision signals in optimization (Rafailov et al., 2024). These approaches, however, often overlook the challenges in preference modeling that arise from the inherent complexity of human preference (Casper et al., 2023): **(1) Human preference may hinge on multiple factors.** The multifaceted factors entailed by a prompt may not be easily represented by a single reward function, especially when some factors conflict with one another. A typical example is the divergence between "helpfuness" and "safty," which differ dramatically in their preferred response patterns, making it difficult for a single reward model to achieve the best of both worlds (Mu et al.,

2024). **(2) The factors may vary across tasks and populations.** There lacks a unified way to represent all factors. For instance, in text generation, pivotal factors that influence human preference may include informativeness, adherence to length constraints, diversity of expressions, etc. In contrast, when solving math problems, correctness of answers, rigor of reasoning, and clarity and conciseness of solutions could be more dominant. **(3) Accurately determining the relative weights of factors for a prompt is challenging, even if the factors are well-defined.** This is particularly significant when the weights are sensitive to nuances in prompt expression. For example, the prompt "how can I kill a *Python process*" demands less consideration on safety than "how can I kill *someone*" despite similar superficial phrasing.

In light of the challenges in preference modeling, we aim to develop a unified framework for capturing the intricate nature of human preferences, with the goal of achieving (1) the framework can broadly represent human preferences across diverse tasks; (2) the framework allows for automatic learning of preference representations without the need for pre-defined sub-rewards and hand-crafted weights that are required by many existing approaches (Zhou et al., 2024; Rame et al., 2024; Yang et al., 2024c); and (3) the framework is generally applicable to various alignment algorithms and can effectively and consistently enhance their performance.

To this end, we propose Latent Preference Coding (LPC), a novel framework that captures the multifaceted nature of human preferences through discrete latent codes. LPC introduces a discrete latent space where each code represents an underlying factor influencing holistic preferences. Through variational inference, LPC estimates the latent codes from data, and learns both a prior network and a posterior network. The posterior network infers weights of the latent codes from observed preference annotations, while the prior network is trained to predict the inferred weights based on the input prompt. Together, the latent codes and the predicted combination weights form a mixture of factors that represent prompt-specified human preferences, guiding the generation of completions in LLMs[1]. More importantly, the formulation of LPC is general, allowing for integration with a wide range of offline preference algorithms, including DPO (Rafailov et al., 2024), SimPO (Meng et al., 2024), IPO (Azar et al., 2024), and others.

We conduct extensive experiments to assess LPC across diverse downstream tasks, employing Mistral-7B (Jiang et al., 2023), Llama3-8B, and Llama3-8B-Instruct (Dubey et al., 2024) as base LLMs, paired with DPO, SimPO, and IPO as alignment algorithms. Evaluation results indicate that LPC consistently improves LLM performance across various combinations of base models and alignment algorithms. More interestingly, further analysis over the learned latent codes reveals that LPC effectively captures the underlying distribution of human preferences collected from different data sources, and exhibits robustness against noisy annotations. These results confirm that LPC provides a unified approach for representing the complex structures underlying human preferences and is readily applicable to a wide range of existing alignment algorithms.

Our contributions are threefold: (1) We identify the critical challenge of modeling complex human preferences in LLM alignment and propose Latent Preference Coding (LPC) to address the challenge through discrete latent variables; (2) We derive a tailored optimization objective under the LPC framework, which can seamlessly integrate with and enhance the performance of various offline preference learning algorithms; and (3) Extensive experiments on multiple benchmarks, using various base LLMs and alignment algorithms, validate the consistent effectiveness of LPC over the vanilla counterparts.

## 2. Related Work

### 2.1. Latent Variable Models

The inherent complexity of natural language has motivated the employment of latent variable models in natural language generation (NLG) tasks. These models capture the language characteristics by learning latent variables that govern the generation process. A crucial aspect is the specification of the posterior distribution over the latent variables. Continuous distributions, such as the Gaussian distribution used in the variational auto-encoder (VAE) framework (Kingma, 2013), have been widely adopted for modeling response diversity (Zhao et al., 2017; Ke et al., 2018). Recently, discrete distributions have emerged as a promising alternative, offering several compelling advantages, including mitigating the notorious posterior collapse issue (Bowman et al., 2016), enabling enhanced controllability through latent variable manipulation (Bartolucci et al., 2022), and demonstrating remarkable interpretability by revealing correspondences between latent variables and categorical language features like dialogue acts (Zhao et al., 2018), entity states (Guan et al., 2023), and writing actions (Cornille et al., 2024). The discrete latent variable models typically rely on a multinomial distribution over a learnable codebook (Van Den Oord et al., 2017) or a predefined vocabulary (Zelikman et al., 2024) to represent the discrete latent space. Despite the extensive exploration of latent variable models, their applications to the alignment of LLMs remains largely unexplored. Recently, a few works have attempted to apply

[1] LPC facilitates the automatic learning of prompt-specific preference representations. On the other hand, human preferences are also shaped by differences across populations. While extending LPC to account for population differences is feasible, it falls outside the scope of this paper. We leave the exploration of personalized LPC for future work.

discrete latent variables to the alignment of LLMs. Poddar et al. (2024) employs continuous latent variables to represent various personalized human needs. A concurrent study by Yao et al. (2024) proposes a variational approach for learning pluralistic preferences within a group. Our work distinguishes itself from these works by modeling the intricate preferences obscured in the prompts through the more interpretable approach of discrete latent variables.

### 2.2. Learning from Human Feedback

Learning from human feedback has been a crucial paradigm in aligning LLMs. Various forms of feedback have been explored, including labels (Hastie et al., 2009), scalar ratings (Silver et al., 2021; Richemond et al., 2024), expert trajectories (Hussein et al., 2017), and pairwise rankings (Wirth et al., 2017; Rafailov et al., 2024), all of which can be viewed as carriers of underlying human preferences. Recently, reward modeling techniques, particularly those based on pairwise rankings, have emerged as a promising approach for providing scalable feedback, such as the Bradley-Terry model (Bradley & Terry, 1952). Such reward models can then be leveraged to align LLMs with human preference through reinforcement learning algorithms like PPO (Schulman et al., 2017). This has been applied to ensure safety (Dai et al., 2024), enhance helpfulness (Nakano et al., 2021), and promote honesty (Tian et al., 2024) in LLMs. However, the complex implementation, hyperparameter tuning, sample inefficiency, and computational overhead of PPO (Choshen et al., 2020) have motivated the exploration of simpler approaches, including rejection sampling (Touvron et al., 2023) that fine-tunes LLMs on responses with the highest reward among a number of samples, and direct preference optimization (DPO) (Rafailov et al., 2024) that directly optimizes LLMs from human preference data without an explicit reward model. Following DPO, various preference optimization objectives have been proposed, such as KTO (Ethayarajh et al., 2024), DRO (Richemond et al., 2024), SimPO (Meng et al., 2024), and GPO (Tang et al., 2024). Despite these advancements, a common limitation of existing methods is their assumption of a single, unified reward function, which may fail to capture the multifaceted nature of human preferences.

### 2.3. Multi-Objecitve Optimization

Multi-objective optimization for aligning LLMs has garnered significant attention, as it mitigates potential dichotomies between competing objectives (Bai et al., 2022) and caters to diverse user needs (Dong et al., 2023). Existing approaches to multi-objective alignment can be broadly categorized into three groups: **(1) Reward Model Combination**, which transforms multi-objective alignment into a single-objective optimization problem by linearly combining rewards from individual reward models (Wu et al., 2023)

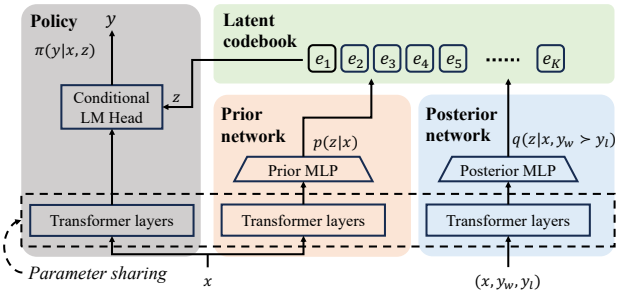


*Figure 1.* Overview of **Latent Preference Coding**. The framework is comprised of a discrete codebook and three modules: a policy model $\pi_\theta(y|x,z)$ conditioned on a latent variable $z$, a prior network $p(z|x)$ that learns to infer $z$ from the prompt, and a posterior network $q(z|x, y_w \succ y_l)$ that guides the training of the prior network and latent code embeddings.

or via parameter interpolation (Rame et al., 2023). Then, they use standard RL approaches to maximize the scalar reward. **(2) Policy Model Combination**, which applies the spirit of linear combination to policy models, i.e., combining policy models learned from different reward models through token-wise probability interpolation (Jang et al., 2023). **(3) Combination-aware Learning**, which trains a single policy model conditioned on both the user instruction and the expected combination weights of different objectives (Dong et al., 2023; Wang et al., 2024). All these methods require explicit human feedback for each objective and demand pre-specified weights for combining multi-objective rewards, imposing a substantial burden on human annotators. In contrast, our approach aims to automatically infer both the implicit factors and their relative importance from holistic feedback data, without relying on pre-defined objective weights or explicit reward models.

## 3. Methodology

We elaborate Latent Preference Coding (LPC) in this section. Starting from a brief review of existing efforts in reinforcement learning from human feedback (RLHF) (§3.2), we derive the optimization objective of LPC (§3.2), and then formulate the latent representation of preferences and other important components in LPC (§3.3). Finally, we demonstrate how LPC can be seamlessly integrated into a variety of offline RLHF algorithms (§3.4).

### 3.1. Preliminaries: Reinforcement Learning from Human Feedback

The goal of RLHF is to optimize a language model $\pi_\theta(y|x)$ parameterized by $\theta$, initialized from a reference model $\pi_{\text{ref}}(y|x)$ obtained through pre-training or supervised fine-tuning. The optimization of $\pi_\theta(y|x)$ is guided by a reward

model parameterized as $r_\phi(x, y)$, whose responsibility is to evaluate how well the output $y \sim \pi_\theta(y|x)$ aligns with human preference. Specifically, the policy model $\pi_\theta(y|x)$ is optimized to maximize the expected reward from $r_\phi(x, y)$ while constrained by a KL penalty with respect to the reference model $\pi_{\text{ref}}(y|x)$ (Ouyang et al., 2022):

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_\phi(x, y)] - \beta \cdot \mathbb{D}_{\text{KL}}[\pi_\theta(y|x)||\pi_{\text{ref}}(y|x)], \tag{1}$$

where $\beta$ acts as a trade-off between the expectation of the reward and the KL term.

Normally, $r_\phi(x, y)$ is estimated from a preference dataset $\mathcal{D} = \{(x^i, y_w^i, y_l^i)\}_{i=1}^N$ by optimizing a Bradley-Terry (BT) model (Bradley & Terry, 1952):

$$p(y_w \succ y_l|x) = \sigma(r_\phi(x, y_w) - r_\phi(x, y_l)) \tag{2}$$

where for prompt $x$, completion $y_w$ is more preferred than $y_l$.

Problem 1 often requires a complex and unstable online algorithm (Schulman et al., 2017), which motivates the exploration on offline RLHF. In fact, as pointed out in (Go et al., 2023), the solution to the KL-constrained reward maximization objective 1 can be analytically written as:

$$\pi_\theta^\star(y|x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y|x)\exp(\beta^{-1}r_\phi(x, y)), \tag{3}$$

where $Z(x)$ is the partition function. Hence, reward $r_\phi(x, y)$ can be represented by:

$$r_\phi(x, y) = \beta\log(\frac{\pi_\theta^\star(y|x)}{\pi_{\text{ref}}(y|x)}) + \beta\log(Z(x)). \tag{4}$$

Putting Eq. 2 and Eq. 4 together, RLHF can be performed offline without the need of an explicit reward by learning from the following loss (Rafailov et al., 2024):

$$\begin{aligned}\mathcal{L}_{\text{DPO}} &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log p(y_w \succ y_l|x) \\ &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right].\end{aligned} \tag{5}$$

Due to its simplicity and effectiveness, offline RLHF has been adopted in the development of several leading LLMs (Touvron et al., 2023; Dubey et al., 2024; Yang et al., 2024a). Therefore, we choose offline RLHF as the starting point for our research on preference modeling, and leave the exploration for online RLHF as future work.

### 3.2. Learning Objective of Latent Preference Coding

Recognizing the diverse and multifaceted nature of human preferences, our approach deviates from traditional RLHF methods that rely on a single reward model $r_\phi(x, y)$ to evaluate all data instances (either explicitly (Schulman et al., 2017) or implicitly (Rafailov et al., 2024)). Instead, we aim to capture the factors that underpin intricate holistic human preferences. To this end, two problems must be addressed: (1) *How to model the mixture of factors implied by a prompt?* And (2) *How to automatically and effectively learn the mixtures of factors from data in an unsupervised fashion?* To answer these questions, we propose latent preference coding (LPC) that implicitly models the underlying factors behind human preferences using latent variables.

We assume that holistic human preference is a mixture of multiple unobserved factors, and can be modeled by a latent variable $z$. Hence, the preference model $p(y_w \succ y_l|x)$ in Eq. 5 can be factorized as $p(y_w \succ y_l|z, x) \cdot p(z|x)$, where $p(z|x)$ is a prior modeling the induction of a mixture of factors as a specific preference pattern with respect to prompt $x$, and $p(y_w \succ y_l|z, x)$ measures how $y_w$ is preferred over $y_l$ under the prompt and the preference pattern. Following the assumption, the loss given by Eq. 5 can be re-formulated as:

$$\mathcal{L}_{\text{LPC-DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log \mathbb{E}_{z\sim p(z|x)} p(y_w \succ y_l|x, z), \tag{6}$$

where

$$p(y_w \succ y_l|x, z) = \sigma\left(\beta\log\frac{\pi_\theta(y_w|x, z)}{\pi_{\text{ref}}(y_w|x, z)} - \beta\log\frac{\pi_\theta(y_l|x, z)}{\pi_{\text{ref}}(y_l|x, z)}\right) \tag{7}$$

Normally, it is difficult to directly optimize Eq. 6 due to the intractability of $p(z|x)$. Therefore, we consider a posterior $q(z|x, y_w \succ y_l)$ and perform learning through variational inference. The posterior takes the observed preference between $y_w$ and $y_l$ as input and predicts a distribution of $z$, which is then used to guide the direction of the prior. By this means, the negative evidence lower bound (ELBO) for $\mathcal{L}_{\text{LPC-DPO}}$ is given by:

$$\begin{aligned}\tilde{\mathcal{L}}_{\text{LPC-DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[ \\ \mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x, z)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x, z)}{\pi_{\text{ref}}(y_l|x)}\right) \\ - \lambda\mathbb{D}_{\text{KL}}[q(\cdot|x, y_w \succ y_l)||p_z(\cdot|x)]\Big],\end{aligned} \tag{8}$$

where $\lambda$ is a hyper-parameter. Details of derivation are presented in Appendix A.

During inference, LPC first samples a latent variable $z$ according to the prior $p(z|x)$, and then generates the completion $y$ from $\pi_\theta(y|x, z)$.

### 3.3. Modeling of Latent Preference Coding

Figure 1 illustrates the architecture of LPC. To effectively represent the multifaceted factors that shape holistic human preferences, we propose to model the underlying factors through discrete latent variables. Basically, we implement

LPC based on the standard decoder-only Transformer architecture (Vaswani et al., 2017) parameterized by $\theta$, taking the input prompt $x$ and generating the output completion $y$. We use $\boldsymbol{h}_x$ and $\boldsymbol{h}_{x,y}$ to denote the hidden states of the last layer at the last token of $x$ and the concatenation of $x$ and $y$, respectively.

**Discrete Latent Space.** We introduce a discrete codebook $E = \{\boldsymbol{e}_k \in \mathbb{R}^d\}_{k=1}^K$ that comprising $K$ codes, where each code $\boldsymbol{e}_k$ corresponds to an underlying factor influencing the holistic preference. We assume that both the prior and posterior distributions are categorical distributions over the latent codes in $E$, making it easy to derive the KL divergence between them in Eq. 8.

**Posterior network.** Given a triple of $(x, y_w, y_l)$, we implement the posterior network by applying a two-layer MLP on the concatenation of $\boldsymbol{h}_{x,y_w}$ and $\boldsymbol{h}_{x,y_l}$:

$$q(z|x, y_w \succ y_l) = \text{softmax}\left(\text{MLP}_{\text{posterior}}([\boldsymbol{h}_{x,y_w}; \boldsymbol{h}_{x,y_l}])\right). \tag{9}$$

**Prior network.** Given an input prompt $x$, the prior network feeds $\boldsymbol{h}_x$ to another MLP, which predicts the prior distribution over the latent codes:

$$p(z|x) = \text{softmax}\left(\text{MLP}_{\text{prior}}(\boldsymbol{h}_x)\right). \tag{10}$$

**Policy Model.** To effectively leverage the insights gained from LPC, the policy model should seamlessly integrate the holistic preference representation derived from the latent variable $z$ into the language generation process. Formally, we model the conditional probability $\pi_\theta(y|x, z)$ as follows:

$$\begin{aligned}\pi_\theta(y|x, z) &= \prod_t \pi_\theta(y_t, |x, z, y_{<t}) \\ &= \prod_t \text{softmax}(\text{LMHead}(\boldsymbol{h}_{x,y_{<t}} + \boldsymbol{z})),\end{aligned} \tag{11}$$

where LMHead is the language model head mapping the hidden states to the vocabulary, $\boldsymbol{h}_{x,y_{<t}}$ denotes the hidden state of the language model encoding the prompt $x$ and the partially generated completion $y_{<t}$, and $\boldsymbol{z}$ denotes the representation of the holistic human preference derived from the prior or posterior distributions of the latent variable $z$.

To circumvent the non-differentiability of sampling from discrete categorical distributions, we leverage the Gumbel-softmax reparameterization trick (Jang et al., 2017), which allows us to obtain continuous and differentiable samples from the prior and posterior distributions over the latent codes. Specifically, we derive $\boldsymbol{z}$ as a convex combination of all latent code embeddings in $E$, weighted by the Gumbel-softmax samples from the prior and the posterior distributions:

$$\begin{aligned}\boldsymbol{z} &= \sum_{k=1}^K c_k \boldsymbol{e}_k, \\ \{c_k\}_{k=1}^K &= g \cdot \text{Gumbel-softmax}(p(z|x)) \\ &\quad + (1-g) \cdot \text{Gumbel-softmax}(q(z|x, y_w \succ y_l)),\end{aligned} \tag{12}$$

where $\{c_k\}_{k=1}^K$ is the categorical distribution over the latent codes after applying Gumbel-softmax on the prior or posterior distributions, and $g \in [0, 1]$ is a weight that determines the relative contributions of the prior and the posterior distributions in deriving $\boldsymbol{z}$. We employ a linear scheduling strategy to gradually increase $g$ from 0 to 1 during training, allowing the model to initially rely more on the more accurate posterior distribution for guidance, and progressively shift towards the prior distribution as the training goes on. In this way, LPC can automatically infer their relative importance between different underlying factors during training.

It is worth noting that although LPC seems to be a bit complex in formulation, it is actually simple to implement and introduce negligible additional computational cost. This is because the policy model, prior and posterior networks share the same backbone model. For the input triple $< x, y_w, y_l >$, we only need to forward the backbone LM twice (once for $< x, y_w >$ and once for $< x, y_l >$), which is the same as DPO.

### 3.4. Extension to Other Offline RLHF Objectives

While the derivation of LPC originates from the DPO objective, its versatile formulation readily extends to other offline RLHF objectives if the obejctives can be formulated as $-\log(f(\cdot))$. This enables a unified framework for capturing the intricate nature of human preferences across different optimization paradigms.

Specifically, when applying LPC to SimPO (Meng et al., 2024), we derive the following loss:

$$\begin{aligned}\mathcal{L}_{\text{LPC -SimPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[&\mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \log\sigma\Bigg( \\ &\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x,z) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x,z) - \gamma\Bigg) \\ &- \lambda\mathbb{D}_{\text{KL}}[q(\cdot|x, y_w \succ y_l)||p_z(\cdot|x)]\Bigg].\end{aligned} \tag{13}$$

Furthermore, drawing inspiration from Eq. 8, we can also apply LPC to objectives that do not strictly satisfy $-\log(f(\cdot))$[2]. While the extension sacrifices some mathe-

[2]In this case, a rigorous derivation of the KL term is not feasible. Therefore, we retain the KL term in Eq. 8 and just replace the expectation term analogously to the formulation used in LPC for DPO.

matical rigor, it proves beneficial in practice, as will be seen in Experiments. Specifically, when applied to IPO (Azar et al., 2024), the loss for learning is given by:

$$\mathcal{L}_{\text{LPC -IPO}} = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)}\Bigg(\log\frac{\pi_\theta(y_w|x,z)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x,z)}{\pi_{\text{ref}}(y_l|x)} - \frac{1}{2\tau}\Bigg)^2 + \lambda\mathbb{D}_{\text{KL}}[q(\cdot|x,y_w\succ y_l)||p_z(\cdot|x)]\Bigg]. \quad (14)$$

Similarly, one can also extends LPC to more objectives as presented in Tang et al. (2024).

# 4. Experiments

## 4.1. Experimental setup

**Configuration.** To comprehensively evaluate the efficacy of LPC, we conduct experiments using three open-source LLMs: Mistral-7B (Jiang et al., 2023), Llama3-8B, and Llama3-8B-Instruct (Dubey et al., 2024). Furthermore, to demonstrate the compatibility and flexibility of LPC, we integrate it with three widely used offline preference learning algorithms: DPO, IPO, and SimPO. These algorithms encompass different optimization strategies and inductive biases, enabling a comprehensive evaluation of LPC's performance across diverse preference learning methods.

**Dataset.** We utilize the widely-adopted UltraFeedback dataset (Cui et al., 2023) in experiments. The dataset is a comprehensive collection of user preferences spanning diverse domains. It contains 63,967 instances from 6 publicly available datasets, including TruthfulQA, FalseQA, Evol-Instruct, UltraChat, ShareGPT, and FLAN. We randomly sample 1,000 instances for validation and an additional 1,000 instances for testing. The rest of the instances are used for training LPC and the baseline alignment methods. We adopt the same data preprocessing pipeline as outlined in (Tunstall et al., 2023) to construct the preference pairs. For each instance, four completions are generated by different LMs. The completion with the highest overall score is denoted as $y_w$, while $y_l$ is randomly sampled from the remaining completions.

**Evaluation.** We first evaluate LPC and the baselines on several representative downstream benchmarks in terms of three aspects: (1) Commonsense Reasoning: we employ ARC-challenge and ARC-easy (Clark et al., 2018) as the evaluation datasets. (2) Mathematical Reasoning: GSM8K (Cobbe et al., 2021), a collection of grade-school problems, is exploited for evaluation. (3) Truthfulness: we use TruthfulQA (Lin et al., 2022) to assess the honesty of aligned LLMs. In appendix B.1 we provide more downstream evaluation results.

Then, we assess how well the models capture the holistic human preferences by calculating the preference accuracy for ranking completion pairs. Specifically, the accuracy accounts for the proportion of instances where $y_w$ has a higher reward score than $y_l$ based on Eq. 4. We calculate the preference accuracy on the test set of UltraFeedback comprising 1,000 examples.

**Implementation Details.** We leverage the OpenRLHF library (Hu et al., 2024) for model training. All models are trained for one epoch, employing the AdamW optimizer (Loshchilov, 2017) and a linear learning rate scheduler peaking at 5e-7 with a 10% warm-up phase. The global batch size is set to 64 and the max length is 1,024. For LPC, we search $\lambda$ in Eq.8 from $\{0.01, 0.05, 0.1\}$ and find $\lambda = 0.05$ yields good performance across all methods. For the DPO and SimPO methods, we regulate the deviation from the reference model by setting $\beta$ in Eq. 5 and Eq. 13 to $0.1$. In the case of IPO, we explore the optimal $\tau$ value in Eq. 14 from $\{0.01, 0.05, 0.1, 0.5\}$ based on the validation performance and empirically choose $\tau = 0.01$. For downstream task evaluation, we utilize the Language Model Evaluation Harness library (Gao et al., 2024), adhering to the default hyper-parameters and evaluation settings.

## 4.2. Main Results

**Downstream Benchmark Evaluation.** As demonstrated in Table 1, it is evident that the proposed LPC framework consistently enhances the performance of LLMs across a diverse range of downstream tasks, base models, and preference methods. A closer examination of the results reveals several key insights: (1) DPO emerges as the most robust alignment method, yielding consistent performance gains across all datasets. Notably, when augment with LPC, DPO's performance is consistently amplified, accentuating the synergistic benefits of LPC in modeling the underlying preference factors. (2) SimPO and IPO exhibit more variability in their performance, occasionally underperforming the base models on certain tasks, particularly GSM8K. However, when integrated with LPC, these performance deficits are mitigated, and in some cases, even surpassed (e.g., IPO w. LPC for Mistral-7B and Llama3-8B-Instruct on GSM8K, and SimPO w. LPC for Llama3-8B on TruthfulQA), underscoring LPC's ability to elucidate and harmonize the disparate preference factors. (3) LPC's impact is not uniformly distributed across all tasks. Specifically, on tasks that heavily rely on the model's intrinsic capabilities, such as abstraction and reasoning skills in the case of the ARC datasets, LPC's fine-grained preference modeling yields relatively modest improvements. This suggests that while LPC excels in capturing the nuances of human preferences, it may have a limited influence on enhancing the model's commonsense reasoning capabilities, which are primarily

*Table 1.* Evaluation results on the downstream tasks. We conducted 5 runs per model per task using different random seeds and report the mean and the standard deviation across the 5 runs. Some details of the 5 runs are provided in Appendix B.3.

| | Arc-Challenge | Arc-Easy | Gsm-8K | Truth-fulQA | Average |
|---|---|---|---|---|---|
| **Mistral-7B** | | | | | |
| ***Base*** | *49.74* | *80.72* | *37.30* | *41.13* | *52.22* |
| **DPO** | 55.38 | 83.33 | 40.11 | **48.10** | 56.73 |
| **w. LPC** | **55.55**$_{0.1}$ | **83.54**$_{0.1}$ | **44.28**$_{2.8}$ | 47.86$_{0.4}$ | **57.81** |
| **IPO** | **58.19** | **84.76** | 30.48 | 48.96 | 55.60 |
| **w. LPC** | 57.17$_{0.1}$ | 83.88$_{0.5}$ | **42.61**$_{2.5}$ | **50.80**$_{0.7}$ | **58.61** |
| **SimPO** | **58.11** | **84.68** | 31.01 | 49.33 | 55.78 |
| **w. LPC** | 56.40$_{0.5}$ | 83.59$_{0.4}$ | **32.60**$_{0.6}$ | **51.29**$_{0.8}$ | **55.97** |
| **Llama3-8B** | | | | | |
| ***Base*** | *50.43* | *80.05* | *49.51* | *43.82* | *55.95* |
| **DPO** | 54.01 | 81.27 | 54.36 | 43.70 | 58.33 |
| **w. LPC** | **54.18**$_{0.2}$ | **81.48**$_{0.2}$ | **55.34**$_{0.5}$ | **44.68**$_{0.8}$ | **58.92** |
| **IPO** | 51.37 | **80.89** | 50.42 | 44.19 | 56.72 |
| **w. LPC** | **51.54**$_{0.2}$ | 80.81$_{0.3}$ | **50.95**$_{0.1}$ | **45.41**$_{0.3}$ | **57.18** |
| **SimPO** | **54.95** | **81.90** | **46.02** | 39.53 | 55.60 |
| **w. LPC** | 53.33$_{0.4}$ | 81.36$_{0.5}$ | 45.87$_{0.6}$ | **53.61**$_{1.8}$ | **58.54** |
| **Llama3-8B-Instruct** | | | | | |
| ***Base*** | *52.90* | *81.52* | *75.66* | *46.88* | *64.24* |
| **DPO** | 54.35 | 82.24 | 77.03 | 47.00 | 65.16 |
| **w. LPC** | **55.29**$_{0.4}$ | **82.41**$_{0.3}$ | **77.79**$_{0.3}$ | **48.10**$_{0.3}$ | **65.90** |
| **IPO** | 53.84 | 81.57 | 73.69 | **47.25** | 64.09 |
| **w. LPC** | **54.69**$_{0.3}$ | **82.24**$_{0.3}$ | **76.80**$_{1.1}$ | 46.51$_{0.4}$ | **65.06** |
| **SimPO** | 55.97 | **83.50** | 66.79 | **56.79** | 65.77 |
| **w. LPC** | **57.34**$_{0.3}$ | 82.91$_{0.6}$ | **73.62**$_{1.9}$ | 56.06$_{0.4}$ | **67.48** |

*Table 2.* Preference accuracy before and after integrating LPC.

| | Llama3-8B | Llama3-8B-Instrcut | Mistral-7B |
|---|---|---|---|
| **DPO** | 69.3 / **70.8** | **70.1** / 69.9 | 71.9 / **73.4** |
| **IPO** | 68.0 / **70.6** | 68.3 / **70.3** | 74.2 / **74.7** |
| **SimPO** | 69.2 / **71.8** | **74.1** / 73.2 | 73.5 / **75.6** |

shaped during the pre-training stage.

**Preference Accuracy.** Subsequently, we delve into the preference accuracy evaluation to assess the efficacy of LPC in distinguishing between favorable and unfavorable completions. As presented in Table 2, the integration of LPC generally elevates preference accuracy across various base models and alignment algorithms. This empirical evidence corroborates LPC's capacity to elucidate and harmonize the intricate factors that shape human preferences. Notably, for the Llama3-8B-Instruct model, the impact of LPC on preference accuracy appears relatively muted. We conjecture that this is because Llama3-8B-Instruct has been extensively fine-tuned for instruction-following, which imbues it with an enhanced ability to adhere to diverse human instructions. Consequently, the influence of LPC's fine-grained preference modeling may be somewhat constrained. Nevertheless, as aforementioned, LPC continues to confer substantial performance improvements on downstream tasks, even for Llama3-8B-Instruct, underscoring its versatility and robustness.

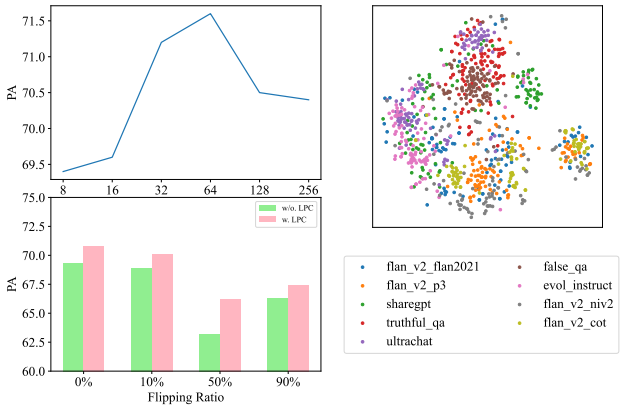


*Figure 2.* **Top left:** Preference accuracy (PA) of DPO w. LPC on Llama3-8B varying with the latent codebook size. **Bottom Left:** Flipping-label experiment on Llama3-8B. Models are evaluated on the original test set with unflipped labels. **Right:** Visualization of the latent variable $z$ produced by the prior network of Llama3-8B. The alignment method is DPO. For each data source in UltraFeedback, we randomly select 100 instances and visualize the T-SNE features of these instances.

### 4.3. Latent Code Analysis

To gain deeper insights into the proposed LPC framework, we conduct experiments to unravel two pivotal research questions: (1) What is the optimal size of the latent codebook to effectively capture the intricate landscape of human preferences? (2) Does LPC truly capture the implicit factors underpinning holistic preferences as hypothesized?

**Investigating the Optimal Codebook Size.** To investigate the optimal codebook size, we train a series of models with distinct codebook sizes ranging from the set $\{8, 16, 32, 64, 128, 256\}$. As illustrated in Figure 2 (Top Left), the preference accuracy exhibits a distinct pattern: initially increasing with larger codebook sizes, peaking around 32 to 64 codes, and then gradually declining, suggesting that it is crucial for LPC's performance to striking the right balance in the size of the latent codebook. When the codebook

*Table 3.* Preference accuracy on complex preference scenarios on Llama-8B. TR: truthfulness, HP: helpfulness, HN: honesty.

| | TR vs. HP | HP vs. HN |
|---|---|---|
| **DPO** | 62.2/64.5 | 67.6/65.1 |
| **w. LPC** | 63.8/65.0 | 68.8/65.5 |
| **IPO** | 61.2/64.8 | 67.9/65.4 |
| **w. LPC** | 61.5/65.2 | 68.4/65.3 |
| **SimPO** | 63.1/65.5 | 68.7/65.2 |
| **w. LPC** | 64.2/65.7 | 68.8/65.7 |

*Table 4.* Results on AlpacaEval 2 judged by GPT-4-turbo-2024-04-09. "LC" and "WR" denote length-controlled and raw win rate, respectively. All methods are based on Llama3-8B-Instruct. The baseline model compared against is GPT-4-1106-preview. We use the official evaluation script (Li et al., 2023b), adopting the same decoding hyper-parameters as Meng et al. (2024), with the temperature set to 0.9.

| | **DPO** LC/WR | **IPO** LC/WR | **SimPO** LC/WR |
|---|---|---|---|
| **w/o. LPC** | 15.03/13.55 | 14.44/12.96 | 12.77/8.12 |
| **w. LPC** | **15.31/13.57** | **15.04/13.50** | **15.56/9.63** |

is small (e.g., 8 codes), it may be insufficiently expressive to capture the diversity of implicit preference factors, thereby limiting performance. Conversely, when the codebook is excessively large (e.g., 256 codes), LPC does not appear to derive significant benefits from an expanded latent space. This could be attributed to several factors: (1) the model may struggle to effectively utilize such a high-dimensional latent space given the limited training data, or (2) the risk of overfitting increases as the codebook size grows.

**Investigating the Capability to Distinguish Implicit Factors.** The core rationale behind predicting implicit preference factors using the prior network $p(z|x)$ lies in the assumption that the prompt $x$ accurately reflects the underlying preference structure. To validate this critical assumption, we devise a probing experiment by intentionally distorting the preference annotations in the UltraFeedback dataset. Specifically, we randomly flip $x\%$ of the preference labels in the training data (i.e., replacing "$y_w \succ y_l$" with "$y_w \prec y_l$"), appending a special token `[FLIP]` to the prompts associated with these flipped instances. Subsequently, we train Llama3-8B using DPO with or without LPC on this distorted dataset to assess whether LPC can effectively differentiate between flipped preferences and normal ones. Then, we calculate the preference accuracy on the original test set of UltraFeedback. As illustrated in Figure 2 (Bottom Left), LPC consistently outperforms the baseline DPO across all flipping ratios, and with 50% labels flipped, LPC even improves the baseline by a larger margin than in the ordinary setting, indicating that in more complex preference environments with intermixed preferences—some of which are even completely opposite—LPC's capability to model implicit preference factors enables it to distinguish and disentangle these conflicting signals, thereby enhancing overall performance.

Additionally, we employ T-SNE (Van der Maaten & Hinton, 2008) to visualize the latent variable $z$. As depicted in Figure 2 (Right), instances from different data sources cluster into several distinct groups. This clustering phenomenon arises because data from various sources typically emphasize different preferences. This observation further corroborates the effectiveness of LPC in modeling implicit preference factors, as it can capture the intricate preference structures inherent in diverse data sources.

### 4.4. Performance on Complex Human Preference Scenarios

We simulate complex preference scenarios on UltraFeedback. We take the helpfulness/truthfulness or helpfulness/honesty labels in the Ultrafeedback dataset as two preference directions and construct a new dataset whose preference scores are mixed between the two directions. Specifically, for each instance, we construct $< y_w, y_l >$ pairs using the two directions with equal probability. As seen in Table 3, we find that our method can still achieve a satisfying performance, which indicates the effectiveness of our method in handling complex human intentions.

### 4.5. Win Rate Against GPT-4

To further validate LPC's efficacy in aligning LLMs with human preferences, we evaluate LPC on AlpacaEval 2 (Li et al., 2023b) using GPT-4 as a judge. As depicted in Table 4, LPC brings performance improvements across all alignment algorithms on Llama3-8B-Instruct, further solidifying its prowess in aligning LLMs with human preferences.

## 5. Conclusions

In this work, we propose LPC, a framework that enables LLMs to capture the multifaceted nature of human preferences. LPC introduces discrete latent codes where each code represents an underlying factor influencing holistic preferences. Through variational inference, LPC can model the implicit factors without the need for fine-grained preference annotations. Besides, LPC can be integrated with a variety of offline preference algorithms, including DPO, IPO, SimPO, and so on. We conduct extensive experiments evaluating LPC on three open-source LLMs, showing that LLMs can achieve better performance across multiple benchmarks by modeling the underlying factors of human preference.

## Impact Statement

The algorithm proposed in this paper automatically captures latent preference factors from overall preference, which might be biased, and does not reflect real human preferences.

## References


Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., DasSarma, N., et al. A general language assistant as a laboratory for alignment. arXiv preprint arXiv:2112.00861, 2021.

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

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862, 2022.

Bartolucci, F., Pandolfi, S., and Pennoni, F. Discrete latent variable models. Annual Review of Statistics and Its Application, 9(1):425–452, 2022.

Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp. 10–21, 2016.

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

Calandriello, D., Guo, D., Munos, R., Rowland, M., Tang, Y., Pires, B. A., Richemond, P. H., Lan, C. L., Valko, M., Liu, T., et al. Human alignment of large language models through online preference optimisation. arXiv preprint arXiv:2403.08635, 2024.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv preprint arXiv:2307.15217, 2023.

Choshen, L., Fox, L., Aizenbud, Z., and Abend, O. On the weaknesses of reinforcement learning for neural machine translation. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=H1eCw3EKvH.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv:1803.05457v1, 2018.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

Cornille, N., Moens, M.-F., and Mai, F. Learning to plan for language modeling from unlabeled data. arXiv preprint arXiv:2404.00614, 2024.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y., and Yang, Y. Safe RLHF: Safe reinforcement learning from human feedback. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=TyFrPOKYXw.

Dong, Y., Wang, Z., Sreedhar, M., Wu, X., and Kuchaiev, O. SteerLM: Attribute conditioned SFT as an (user-steerable) alternative to RLHF. In Bouamor, H., Pino, J., and Bali, K. (eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, pp. 11275–11288, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.754. URL https://aclanthology.org/2023.findings-emnlp.754.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Model alignment as prospect theoretic optimization. In Forty-first International Conference on Machine Learning, 2024. URL https://openreview.net/forum?id=iUwHnoENnl.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

Go, D., Korbak, T., Kruszewski, G., Rozen, J., Ryu, N., and Dymetman, M. Aligning language models with preferences through f-divergence minimization. In Proceedings of the 40th International Conference on Machine Learning, pp. 11546–11583, 2023.

Guan, J., Yang, Z., Zhang, R., Hu, Z., and Huang, M. Generating coherent narratives by learning dynamic and discrete entity states with a contrastive framework. In Proceedings of the AAAI conference on artificial intelligence, volume 37, pp. 12836–12844, 2023.

Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K., Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang, M., Gu, C., et al. Reinforced self-training (rest) for language modeling. arXiv preprint arXiv:2308.08998, 2023.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., and Friedman, J. Overview of supervised learning. The elements of statistical learning: Data mining, inference, and prediction, pp. 9–41, 2009.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In International Conference on Learning Representations.

Hu, J., Wu, X., Wang, W., Xianyu, Zhang, D., and Cao, Y. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. arXiv preprint arXiv:2405.11143, 2024.

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. ACM Computing Surveys (CSUR), 50(2):1–35, 2017.

Jang, E., Gu, S., and Poole, B. Categorical reparametrization with gumble-softmax. In International Conference on Learning Representations (ICLR 2017). OpenReview.net, 2017.

Jang, J., Kim, S., Lin, B. Y., Wang, Y., Hessel, J., Zettlemoyer, L., Hajishirzi, H., Choi, Y., and Ammanabrolu, P. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. arXiv preprint arXiv:2310.11564, 2023.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

Ke, P., Guan, J., Huang, M., and Zhu, X. Generating informative responses with controlled sentence function. In Gurevych, I. and Miyao, Y. (eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1499–1508, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1139. URL https://aclanthology.org/P18-1139.

Kingma, D. P. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

Li, P., Pei, Y., and Li, J. A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing, 138:110176, 2023a.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023b.

Lin, S., Hilton, J., and Evans, O. Truthfulqa: Measuring how models mimic human falsehoods. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 3214–3252, 2022.

Loshchilov, I. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101, 2017.

Meng, Y., Xia, M., and Chen, D. Simpo: Simple preference optimization with a reference-free reward. arXiv preprint arXiv:2405.14734, 2024.

Mu, T., Helyar, A., Heidecke, J., Achiam, J., Vallone, A., Kivlichan, I. D., Lin, M., Beutel, A., Schulman, J., and Weng, L. Rule based rewards for fine-grained llm safety. In ICML 2024 Next Generation of AI Safety Workshop, 2024.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., Michi, A., et al. Nash learning from human feedback. arXiv preprint arXiv:2312.00886, 2023.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. arXiv preprint arXiv:2112.09332, 2021.

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

Poddar, S., Wan, Y., Ivison, H., Gupta, A., and Jaques, N. Personalizing reinforcement learning from human feedback with variational preference learning. arXiv preprint arXiv:2408.10075, 2024.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36, 2024.

Rame, A., Couairon, G., Dancette, C., Gaya, J.-B., Shukor, M., Soulier, L., and Cord, M. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=lSbbC2VyCu.

Rame, A., Couairon, G., Dancette, C., Gaya, J.-B., Shukor, M., Soulier, L., and Cord, M. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. Advances in Neural Information Processing Systems, 36, 2024.

Richemond, P. H., Tang, Y., Guo, D., Calandriello, D., Azar, M. G., Rafailov, R., Pires, B. A., Tarassov, E., Spangher, L., Ellsworth, W., et al. Offline regularised reinforcement learning for large language models alignment. arXiv preprint arXiv:2405.19107, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Silver, D., Singh, S., Precup, D., and Sutton, R. S. Reward is enough. Artificial Intelligence, 299:103535, 2021.

Tang, Y., Guo, Z. D., Zheng, Z., Calandriello, D., Munos, R., Rowland, M., Richemond, P. H., Valko, M., Avila Pires, B., and Piot, B. Generalized preference optimization: A unified approach to offline alignment. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of Proceedings of Machine Learning Research, pp. 47725–47742. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/tang24b.html.

Tian, K., Mitchell, E., Yao, H., Manning, C. D., and Finn, C. Fine-tuning language models for factuality. In The Twelfth International Conference on Learning Representations, 2024. URL https://openreview.net/forum?id=WPZ2yPag4K.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of lm alignment, 2023.

Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. Advances in neural information processing systems, 30, 2017.

Van der Maaten, L. and Hinton, G. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, H., Lin, Y., Xiong, W., Yang, R., Diao, S., Qiu, S., Zhao, H., and Zhang, T. Arithmetic control of LLMs for diverse user preferences: Directional preference alignment with multi-objective rewards. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8642–8655, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.468. URL https://aclanthology.org/2024.acl-long.468.

Wirth, C., Akrour, R., Neumann, G., and Fürnkranz, J. A survey of preference-based reinforcement learning methods. Journal of Machine Learning Research, 18(136): 1–46, 2017.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. In Thirty-seventh Conference on Neural Information Processing Systems, 2023. URL https://openreview.net/forum?id=CSbGXyCswu.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al. Qwen2 technical report. arXiv preprint arXiv:2407.10671, 2024a.

Yang, K., Klein, D., Celikyilmaz, A., Peng, N., and Tian, Y. Rlcd: Reinforcement learning from contrastive distillation for lm alignment. In The Twelfth International Conference on Learning Representations, 2024b.

Yang, R., Pan, X., Luo, F., Qiu, S., Zhong, H., Yu, D., and Chen, J. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. arXiv preprint arXiv:2402.10207, 2024c.

Yao, B., Cai, Z., Chuang, Y.-S., Yang, S., Jiang, M., Yang, D., and Hu, J. No preference left behind: Group distributional preference optimization. arXiv preprint arXiv:2412.20299, 2024.

Zelikman, E., Harik, G., Shao, Y., Jayasiri, V., Haber, N., and Goodman, N. D. Quiet-star: Language models can teach themselves to think before speaking. arXiv preprint arXiv:2403.09629, 2024.

Zhao, T., Zhao, R., and Eskenazi, M. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In Barzilay, R. and Kan, M.-Y. (eds.), Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 654–664, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1061. URL https://aclanthology.org/P17-1061.

Zhao, T., Lee, K., and Eskenazi, M. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In Gurevych, I. and Miyao, Y. (eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1098–1107, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1101. URL https://aclanthology.org/P18-1101.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. Slic-hf: Sequence likelihood calibration with human feedback. arXiv preprint arXiv:2305.10425, 2023.

Zhou, Z., Liu, J., Shao, J., Yue, X., Yang, C., Ouyang, W., and Qiao, Y. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. In Findings of the Association for Computational Linguistics ACL 2024, pp. 10586–10613, 2024.

## A. Deriving the evidence lower bound of LPC

We start with the standard DPO, where the objective is to maximize the log-likelihood $\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log p(y_w \succ y_l|x)$. After introducing latent variable $z$, we have:

$$\begin{aligned}
\log p(y_w \succ y_l|x) &= \log \mathbb{E}_{z\sim p(z|x)} p(y_w \succ y_l|x,z) \\
&= \log \int p(y_w \succ y_l|x,z)p(z|x)\frac{q(z|x,y_w \succ y_l)}{q(z|x,y_w \succ y_l)}dz \\
&= \log \mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \frac{p(y_w \succ y_l|x,z)p(z|x)}{q(z|x,y_w \succ y_l)} \\
&\geq \mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \log \frac{p(y_w \succ y_l|x,z)p(z|x)}{q(z|x,y_w \succ y_l)} \\
&= \mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \left[\log p(y_w \succ y_l|x,z) + \log \frac{p(z|x)}{q(z|x,y_w \succ y_l)}\right] \\
&= \mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \log p(y_w \succ y_l|x,z) - \mathbb{D}_{\text{KL}}[q(\cdot|x,y_w \succ y_l)||p(\cdot|x)].
\end{aligned} \tag{15}$$

By this means, we have:

$$\tilde{\mathcal{L}}_{\text{LPC-DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)} \log p(y_w \succ y_l|x,z) - \mathbb{D}_{\text{KL}}[q(\cdot|x,y_w \succ y_l)||p(\cdot|x)]\right]. \tag{16}$$

Then we need to derive the mathematical solution for $p(y_w \succ y_l|x,z)$. We assume that each latent $z$ corresponds to an implicit reward model $r_{\phi_z}(x,y)$. The derivation process is quite similar to standard DPO.

For each implicit preference factor $z$, we optimize the following objective:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x,z)}[r_{\phi_z}(x,y)] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta(y|x,z)||\pi_{\text{ref}}(y|x,z)]. \tag{17}$$

Because the parameter of the reference model is fixed during training, the output of $\pi_{\text{ref}}$ would not be affected by $z$, i.e., $\pi_{\text{ref}}(y|x,z) = \pi_{\text{ref}}(y|x)$. We now have:

$$\begin{aligned}
&\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(\cdot|x,z)}[r_{\phi_z}(x,y)] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta(y|x,z)||\pi_{\text{ref}}(y|x,z)] \\
&= \max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_{\theta_z}(\cdot|x,z)} \left[r_{\phi_z}(x,y) - \beta\log\frac{\pi_{\theta_z}(y|x,z)}{\pi_{\text{ref}}(y|x)}\right] \\
&= \min_{\pi_{\theta_z}} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_{\theta_z}(\cdot|x,z)} \left[\log\frac{\pi_{\theta_z}(y|x,z)}{\pi_{\text{ref}}(y|x)} - \beta^{-1}r_{\phi_z}(x,y)\right] \\
&= \min_{\pi_{\theta_z}} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_{\theta_z}(\cdot|x,z)} \left[\log\frac{\pi_{\theta_z}(y|x,z)}{\pi^*(y|x,z)} - \log Z_z(x)\right] \\
&= \min_{\pi_{\theta_z}} \mathbb{E}_{x\sim\mathcal{D}}[\mathbb{D}_{\text{KL}}(\pi_{\theta_z}(y|x,z)||\pi^*(y|x,z)) - \log Z_z(x)],
\end{aligned} \tag{18}$$

where:

$$Z_z(x) = \sum_y \pi_{\text{ref}}(y|x)\exp\left(\beta^{-1}r_{\phi_z}(x,y)\right) \tag{19}$$

and

$$\pi^*(y|x,z) = \frac{1}{Z_z(x)}\pi_{\text{ref}}(y|x)\exp\left(\beta^{-1}r_{\phi_z}(x,y)\right). \tag{20}$$

The KL-divergence in Eq.18 reaches the minimum when $\pi_{\theta_z}(y|x,z) = \pi^*(y|x,z))$. As a result, we obtain the expression of optimal reward:

$$r^*_{\phi_z}(x,y) = \beta\log\frac{\pi^*(y|x,z)}{\pi_{\text{ref}}(y|x)} + \beta\log Z_z(x). \tag{21}$$

*Table 5.* Evaluation results on MMLU.

| | MMLU | | |
|---|---|---|---|
| | **Mistral-7B** | **Llama3-8B** | **Llama3-8B-Instruct** |
| **Base Model** | 60.10 | 62.10 | 63.89 |
| **DPO** | 58.48 | 62.68 | 63.60 |
| **w. LPC** | 59.14 | 62.54 | 63.73 |
| **IPO** | 60.23 | 61.56 | 63.25 |
| **w. LPC** | 59.97 | 61.47 | 63.75 |
| **SimPO** | 59.21 | 61.78 | 62.14 |
| **w. LPC** | 59.56 | 61.30 | 62.87 |

*Table 6.* Preference accuracy of different latent representations.

| | Preference Accuracy | | |
|---|---|---|---|
| | **Mistral-7B** | **Llama3-8B** | **Llama3-8B-Instruct** |
| **DPO** | 71.9 | 69.3 | 70.1 |
| **w. LPC (Discrete)** | 73.4 | 70.8 | 69.9 |
| **w. LPC (Continuous)** | 71.4 | 69.5 | 69.7 |
| **IPO** | 74.2 | 68.0 | 68.3 |
| **w. LPC (Discrete)** | 74.7 | 70.6 | 70.3 |
| **w. LPC (Continuous)** | 73.4 | 69.7 | 70.1 |
| **SimPO** | 73.5 | 69.2 | 74.1 |
| **w. LPC (Discrete)** | 75.6 | 71.8 | 73.2 |
| **w. LPC (Continuous)** | 73.1 | 71.0 | 72.9 |

Combining Eq.16 and Eq.21, we can get the final training objective of LPC.

$$\begin{aligned}
\tilde{\mathcal{L}}_{\text{LPC-DPO}} &= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)}\log p(y_w \succ y_l|x,z) - \mathbb{D}_{\text{KL}}[q(\cdot|x,y_w\succ y_l)||p(\cdot|x)]\right], \\
&= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)}\log \sigma(r_{\phi_z}(x,y_w) - r_{\phi_z}(x,y_l)) - \mathbb{D}_{\text{KL}}[q(\cdot|x,y_w\succ y_l)||p(\cdot|x)]\right], \\
&= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\mathbb{E}_{z\sim q(\cdot|x,y_w\succ y_l)}\log \sigma\left(\beta\log\frac{\pi_\theta(y_w|x,z)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x,z)}{\pi_{\text{ref}}(y_l|x)}\right) \\
&\qquad - \mathbb{D}_{\text{KL}}[q(\cdot|x,y_w\succ y_l)||p(\cdot|x)]\Bigg],
\end{aligned} \tag{22}$$

In practice, we insert a hyper-parameter $\lambda$ before the KL term to enhance the flexibility of learning.

## B. More Experimental Results

### B.1. MMLU Evaluation

Table 5 shows evaluation results on MMLU (Hendrycks et al.). MMLU includes a diverse set of tasks including various domains, which is a good indicator of the generalization ability of a model. While LPC does not significantly improve the performance of alignment algorithms on MMLU, it also does not hurt the performance, which is consistent with previous works (Meng et al., 2024). We believe that this is due to the large domain gap between training and evaluation.

### B.2. How to Model Latent Preference? Discrete Code vs. Continuous Variable

As using continuous latent variables is a common practice to model latent factors (Li et al., 2023a), we conduct an ablation study that replaces the discrete latent code with a continuous one sampled from a standard normal distribution while keeping the rest of the framework unchanged. As illustrated in Table 6, the preference accuracy of the model using discrete latent

code is higher than that using continuous latent variable a little bit. Compared with continuous latent representation, discrete latent bypasses the sampling process. We choose to use discrete latent variables in our method mainly because of its simplicity of implementation and training stability.

### B.3. Detailed Results of DPO

*Table 7.* Detailed results of the 5 runs of DPO.

| | | **run1** | **run2** | **run3** | **run4** | **run5** |
|---|---|---|---|---|---|---|
| | | | **Mistral-7B** | | | |
| Arc-Challenge | w/o. LPC | 55.26 | 55.64 | 55.33 | 55.48 | 55.19 |
| | w. LPC | 55.65 | 55.72 | 55.51 | 55.40 | 55.47 |
| Arc-Easy | w/o. LPC | 83.18 | 83.22 | 83.44 | 83.32 | 83.49 |
| | w. LPC | 83.72 | 83.40 | 83.52 | 83.72 | 83.33 |
| GSM8K | w/o. LPC | 39.88 | 40.45 | 39.62 | 40.43 | 40.17 |
| | w. LPC | 46.52 | 43.92 | 41.56 | 41.00 | 48.40 |
| TruthfulQA | w/o. LPC | 48.40 | 47.44 | 48.30 | 47.37 | 48.98 |
| | w. LPC | 47.89 | 48.37 | 47.18 | 47.79 | 48.06 |
| | | | **Llama3-8B** | | | |
| Arc-Challenge | w/o. LPC | 54.14 | 54.01 | 54.36 | 53.85 | 53.69 |
| | w. LPC | 53.97 | 54.04 | 54.35 | 54.37 | 54.17 |
| Arc-Easy | w/o. LPC | 81.21 | 81.07 | 81.41 | 81.36 | 81.28 |
| | w. LPC | 81.74 | 81.41 | 81.68 | 81.12 | 81.44 |
| GSM8K | w/o. LPC | 54.06 | 54.46 | 54.64 | 54.64 | 54.00 |
| | w. LPC | 55.89 | 54.73 | 55.38 | 54.79 | 55.91 |
| TruthfulQA | w/o. LPC | 43.84 | 43.78 | 44.06 | 42.80 | 44.02 |
| | w. LPC | 45.15 | 44.65 | 43.45 | 45.76 | 44.39 |
| | | | **Llama3-8B-Instruct** | | | |
| Arc-Challenge | w/o. LPC | 54.41 | 54.49 | 54.23 | 54.71 | 53.91 |
| | w. LPC | 55.14 | 54.98 | 56.01 | 55.09 | 55.23 |
| Arc-Easy | w/o. LPC | 82.49 | 82.20 | 82.25 | 81.82 | 82.45 |
| | w. LPC | 82.68 | 81.88 | 82.57 | 82.49 | 82.43 |
| GSM8K | w/o. LPC | 77.18 | 77.22 | 77.02 | 76.78 | 76.95 |
| | w. LPC | 77.89 | 77.63 | 77.49 | 77.70 | 78.24 |
| TruthfulQA | w/o. LPC | 47.49 | 46.45 | 47.38 | 46.38 | 47.30 |
| | w. LPC | 47.57 | 48.60 | 48.01 | 48.12 | 48.20 |