
Mix- and MoE-DPO: A Variational Inference Approach to Direct Preference Optimization

Jason Bohne*

Department of Applied Mathematics and Statistics
Stony Brook University
Stony Brook, NY 11794
jason.bohne@stonybrook.edu

Paweł Polak*

Department of Applied Mathematics and Statistics
Institute for Advanced Computational Science
Center of Excellence in Wireless and Information Technology (CEWIT)
AI Innovation Institute
Stony Brook University
Stony Brook, NY 11794
pawel.polak@stonybrook.edu

David Rosenberg

Bloomberg
Toronto, ON M5J 2S1
drosenberg44@bloomberg.net

Brian Bloniarz

Bloomberg
San Francisco, CA 94105
bbloniarz@bloomberg.net

Gary Kazantsev

Bloomberg
New York, NY 10022
gkazantsev@bloomberg.net

Abstract

Direct Preference Optimization (DPO) has recently emerged as a simple and effective alternative to reinforcement learning from human feedback (RLHF) for aligning large language models (LLMs) with user preferences. However, existing DPO formulations rely on a single monolithic model, which limits their expressivity in multi-task settings and their adaptability to heterogeneous or diverse preference distributions. In this work, we propose Mix- and MoE-DPO, a framework that extends DPO with both soft mixture models and mixture-of-experts (MoE) architectures, using a stochastic variational inference approach. Our method introduces a latent-variable model over expert assignments and optimizes a variational evidence lower bound (ELBO), enabling stable and efficient learning of specialized expert policies from preference data. Mix- and MoE-DPO provides three key advantages over standard DPO: (i) generalization via universal function approximation through mixtures; (ii) reward and policy specialization through expert components tailored to distinct preference modes; and (iii) contextual alignment through input-dependent soft gating that enables user-specific mixture policies. Our framework supports both shared base architectures with expert-specific policy heads and fully independent

*Equal contribution.

expert models, allowing flexible trade-offs between parameter efficiency and specialization. We validate our approach on a variety of model sizes and multi-preference datasets, demonstrating that Mix- and MoE-DPO offers a powerful and scalable method for preference-based LLM alignment.

1 Introduction

Aligning large language models (LLMs) with human preferences is a central objective in building safe and reliable AI systems. Direct Preference Optimization (DPO) [1] has emerged as a widely adopted and computationally efficient alternative to Reinforcement Learning from Human Feedback (RLHF) [2–4] for the alignment of LLMs. Using a direct optimization framework over preference pairs, DPO avoids the need for explicit reward modeling and optimization complexities required in RLHF methods, while still achieving competitive performance with traditional methods.

Despite its advantages, standard DPO methods are still inherently limited by their reliance on a single, monolithic policy. In scenarios with multi-expert or heterogeneous preferences arising from varied user groups, task domains, or annotation styles, this uniformity restricts expressivity and may induce suboptimal alignment. Although extensions of DPO [5–12] offer improvements, such modifications to date have been limited to a single-policy framework.

To address the limitations of single-policy preference optimization, we propose a modular framework for aligning large language models (LLMs) with heterogeneous human preferences. While Direct Preference Optimization (DPO) has demonstrated strong empirical performance, its monolithic structure limits its capacity to represent diverse user behaviors, task-specific feedback, or multi-objective alignment. This limitation is particularly significant in light of the growing ecosystem of domain-specialized LLMs, which offer reusable components for structured preference modeling.

We introduce a mixture-based generalization of DPO by modeling the policy as a latent mixture over expert components. Each expert specializes in a distinct preference mode, and the overall model is trained via a variational inference procedure grounded in a Mixture-of-Bradley–Terry (MBT) likelihood. This framework establishes a component-wise policy–reward equivalence and accommodates two architectural regimes: (i) expert-specific heads on a shared encoder for parameter-efficient sharing, and (ii) independently parameterized expert models for maximal specialization. In both cases, expert policies can be initialized from task-specific pretrained heads or independent models for fine-tuning or deployment in a zero-shot setting, enabling efficient and scalable adaptation.

To optimize the mixture model, we propose a stochastic variational EM algorithm that alternates between inferring expert responsibilities and updating policies and rewards. We instantiate this in two variants: *Mix-DPO*, with fixed mixture weights updated via posterior averaging, and *MoE-DPO*, which uses a soft gating network to assign input-dependent or user-specific weights. Expert policies are trained via closed-form updates minimizing expert-specific KL-regularized objectives, while the gating network is optimized to match predicted weights to inferred posteriors.

Our framework is designed for compatibility with modular deployment in real-world systems. Expert components can be swapped, added, or reused with minimal retraining. In shared-encoder settings, new capabilities can be added efficiently via head specialization. Moreover, MoE-DPO supports user personalization by conditioning the gating network on user metadata, allowing for user-specific mixtures without modifying the expert policies. This flexibility makes the framework well suited for scalable alignment in heterogeneous and multi-task environments.

Contributions. This work makes the following theoretical and empirical contributions:

Multi-Expert Preference Alignment. We generalize DPO to a latent mixture of expert policies, combining a Mixture-of-Bradley–Terry model with KL-regularized reward-based objectives.

Variational Training Algorithm. We develop a scalable variational EM algorithm for modular policy optimization under latent expert assignments, preserving closed-form policy updates.

Multi-Reward Generalization with Mix-DPO. We show that Mix-DPO improves generalization across multiple reward signals by enabling posterior specialization of expert policies.

Contextual Multi-Task Alignment with MoE-DPO. We demonstrate that MoE-DPO enables

effective user- or task-specific routing via input-dependent gating, yielding improved multi-domain alignment.

Modular Deployment and Personalization. We highlight that our framework supports integration of pretrained models and can enable efficient customization without retraining via gating.

The paper is structured as follows. Section 2 provides our mixture-model formulation for the policy, reward, and preference model. Section 3 outlines our variational training framework and special cases of Mix-DPO and MoE-DPO. Section 4 presents experimental results for preference alignment of language models on a variety of tasks. Thorough related work on preference alignment methods and variational inference, along with proofs and additional experimental results, are in the Appendix.

Notation. We use the following notation throughout. Prompt–response–preference triplets are denoted as $\{(x_i, y_i^+, y_i^-)\}_{i=1}^n \sim \mathcal{D}$, where $y_i^+ \succ y_i^-$ indicates the preferred response for prompt $x_i \sim p$. The base policy is denoted $\pi(y | x)$, the reward function is $r(x, y)$, and the temperature parameter is β . The latent expert index is denoted $z \in \{1, \dots, K\}$, with conditional prior $p(z = k | x) = w_k(x)$, where $w_k(x)$ is the mixture weight assigned to expert k . Accordingly, the expert-specific policy is $\pi_k(y | x)$, the expert reward is $r_k(x, y)$, and the reference policy is $\pi_{\text{ref}(k)}(y | x)$. The variational posterior over expert assignments for a preference triplet is denoted $q_k(x, y^+, y^-)$.

2 Multi-Expert Preference Alignment

Let $\{(x_i, y_i^+, y_i^-)\}_{i=1}^n$ denote a dataset of preference triplets, where each prompt $x_i \sim p$ is associated with two responses sampled from a policy π , and y_i^+ is preferred over y_i^- . To support modular specialization and better capture heterogeneous user preferences or task distributions, we extend DPO to a mixture-of-experts formulation with input-dependent gating:

$$\pi(y | x) := \sum_{k=1}^K w_k(x) \pi_k(y | x), \quad (1)$$

where each π_k is an expert policy and $w_k(x) \geq 0$ are gating weights satisfying $\sum_{k=1}^K w_k(x) = 1$. The gating function assigns context-dependent responsibilities to expert components, enabling the model to adaptively route different inputs to specialized policies. This structure increases expressivity while preserving compositional modularity and aligns with recent empirical evidence that combining specialized models enhances generalization in multi-domain LLM settings [13, 14].

To define the reward associated with the policy in (1), we aggregate the expert-specific rewards $r_k(x, y)$ using a softmax-style combination:

$$r(x, y) := \log \left(\sum_{k=1}^K w_k(x) \exp(r_k(x, y)) \right). \quad (2)$$

This reward corresponds to a smooth maximum over the expert rewards and recovers the single-expert case when the gating weights are deterministic; i.e., (2) reduces to $r(x, y) = \sum_{k=1}^K w_k(x) r_k(x, y)$ when $w_k(x) \in \{0, 1\}$.

We assume the number of mixture components in the policy and reward spaces is equal, with shared gating function $w_k(x)$ across $k = 1, \dots, K$. This modeling choice reflects the assumption that the same latent structure governs both generation and evaluation: the specific type of expert responsible for generating a response is also responsible for assessing its quality by assigning a reward to the output. This alignment ensures consistency between policy optimization and reward modeling and is particularly realistic in settings where expert components reflect stable sources of heterogeneity. Examples of such heterogeneity include user types, task domains, or annotation protocols, as considered in our experiments.

Analogous to the original DPO formulation, our training objective maximizes the expected reward while applying KL regularization. Namely, for a given regularization strength $\beta > 0$, we optimize:

$$\max_{\{w_k, \pi_k\}} \mathbb{E}_{x \sim p, y \sim \pi(y|x)} [r(x, y)] - \beta \sum_{k=1}^K \mathbb{E}_{x \sim p} [w_k(x) D_{\text{KL}}(\pi_k(y | x) \| \pi_{\text{ref}(k)}(y | x))]. \quad (3)$$

Since the model supports expert specialization, the regularization term is applied to each expert individually, encouraging proximity to its corresponding reference policy. The gating weights $w_k(x)$ modulate the strength of this penalty based on the expert’s responsibility for each input, thereby promoting prompt- (or more general user-context-) dependent regularization. Moreover, analogous to the original DPO formulation, the combined structure of (1), (2), and (3) admits a closed-form solution for the optimal policy in terms of the reward, as shown in Theorem 4.

Before deriving specific properties, we formalize the second core component of our model: reward learning consistent with the structure of our mixture preference model. The classical Bradley–Terry (BT) model defines the probability of preferring y^+ over y^- in context x using a single latent reward function $r(x, y)$:

$$\mathbb{P}(y^+ \succ y^- \mid x) = \frac{e^{r(x, y^+)}}{e^{r(x, y^+)} + e^{r(x, y^-)}}.$$

This formulation arises from a logistic likelihood and is typically fit via maximum likelihood estimation over observed preference triplets $(x_i, y_i^+, y_i^-) \sim \mathcal{D}$:

$$\max_r \sum_i \log \left(\frac{\exp(r(x_i, y_i^+))}{\exp(r(x_i, y_i^+)) + \exp(r(x_i, y_i^-))} \right).$$

To incorporate the modular structure introduced by the policy (1), we extend the classical BT model to account for expert-specific rewards. In particular, since the policy is defined as a mixture over expert policies π_k with gating weights $w_k(x)$, it is natural to model preferences under a mixture of corresponding reward functions r_k . This leads to the Mixture-of-Bradley–Terry (MBT) model, where the latent expert index $z \in \{1, \dots, K\}$ governs which reward function is used to evaluate the pairwise preference.

Conditioned on expert $z = k$, the preference likelihood follows the standard BT model:

$$\mathbb{P}(y^+ \succ y^- \mid x, z = k) = \frac{e^{r_k(x, y^+)}}{e^{r_k(x, y^+)} + e^{r_k(x, y^-)}}. \quad (4)$$

Marginalizing over the latent expert assignment according to the gating distribution $w_k(x) = p(z = k \mid x)$, we obtain the MBT model:

$$\mathbb{P}(y^+ \succ y^- \mid x) = \sum_{k=1}^K w_k(x) \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))}.$$

We learn the reward functions $\{r_k\}$ by minimizing the negative log-likelihood of the observed preferences under the marginal model. This corresponds to maximum marginal likelihood under the latent-variable formulation of the MBT model:

$$\mathcal{L}_{\text{MBT}} = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\log \sum_{k=1}^K w_k(x) \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))} \right]. \quad (5)$$

Direct optimization of the marginal log-likelihood in (5) leads to high instabilities in the gradients, obscures the latent structure of the model, and provides limited insight into expert specialization. To address this, we derive a variational evidence lower bound (ELBO) that decomposes the marginal likelihood into an expected per-expert log-probability term and a KL divergence between the variational posterior and the expert prior. This decomposition not only facilitates scalable optimization via stochastic gradient methods but also enables interpretation of expert responsibilities in terms of soft assignments over preference pairs.

Theorem 1 (ELBO for the MBT Model). *Let (x, y^+, y^-) be a preference triplet with $y^+ \succ y^-$, and let $z \in \{1, \dots, K\}$ be a latent expert index with prior $p(z = k \mid x) = w_k(x)$. Let $\sigma_k(x, y^+, y^-)$ denote the Bradley–Terry likelihood under expert k given in (4). Then, for any variational distribution $q(z \mid x, y^+, y^-)$ satisfying $\sum_k q_k(x, y^+, y^-) = 1$, we have:*

$$\begin{aligned} \log \mathbb{P}(y^+ \succ y^- \mid x) &\geq \sum_{k=1}^K q_k(x, y^+, y^-) \log \left(\frac{w_k(x) \sigma_k(x, y^+, y^-)}{q_k(x, y^+, y^-)} \right) \\ &= \mathbb{E}_{z \sim q} [\log \sigma_z(x, y^+, y^-)] - D_{\text{KL}}(q(z \mid x, y^+, y^-) \parallel p(z \mid x)). \end{aligned}$$

The bound is tight when $q_k(x, y^+, y^-) = \frac{w_k(x) \sigma_k(x, y^+, y^-)}{\sum_{j=1}^K w_j(x) \sigma_j(x, y^+, y^-)}$.

The following corollary follows by applying Theorem 1 under the joint distribution over preference triplets sampled from the mixture policy. In this setting, the variational posterior $q(z \mid x, y^+, y^-)$ can be chosen to match the true posterior $p(z \mid x, y^+, y^-)$, since it admits a closed-form expression and is differentiable with respect to model parameters. This choice makes the KL divergence term in Theorem 1 vanish, resulting in a tight bound that coincides with the log-likelihood. Consequently,

Corollary 1.1 (MBT Variational Loss Function). *Maximizing the variational lower bound from Theorem 1 over the preference distribution yields the MBT training loss that we aim to minimize:*

$$\mathcal{L}_{\text{MBT}} = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\sum_{k=1}^K q_k(x, y^+, y^-) \log \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))} \right]. \quad (6)$$

Reward Decomposition and Expert Alignment: We now analyze the structure of the objective in (3) by decomposing the mixture reward $r(x, y)$ into expert-specific contributions. This decomposition highlights the connection between policy inference and reward modeling in the presence of latent expert structure. In particular, we define two variational distributions that describe expert responsibilities under the policy and reward components, respectively:

$$q_k^{(\pi)}(x, y) = \frac{w_k(x) \pi_k(y \mid x)}{\pi(y \mid x)}, \quad \text{and} \quad q_k^{(r)}(x, y) = \frac{w_k(x) \exp(r_k(x, y))}{\sum_j w_j(x) \exp(r_j(x, y))}. \quad (7)$$

The distribution $q^{(\pi)}$ corresponds to the posterior over experts under the mixture policy, while $q^{(r)}$ arises from the softmax aggregation of expert rewards. The following result expresses the mixture reward exactly in terms of the policy-induced posterior and its divergence from the reward-induced posterior.

Theorem 2 (Equality Decomposition of Reward Mixture). *Let $q^{(\pi)}$ and $q^{(r)}$ be defined as in (7). Then the mixture reward satisfies the following exact decomposition:*

$$\begin{aligned} r(x, y) &= \sum_{k=1}^K q_k^{(\pi)}(x, y) \left[r_k(x, y) + \log w_k(x) - \log q_k^{(\pi)}(x, y) \right] \\ &\quad + D_{\text{KL}} \left(q^{(\pi)}(\cdot \mid x, y) \parallel q^{(r)}(\cdot \mid x, y) \right). \end{aligned} \quad (8)$$

The decomposition in Theorem 2 reveals that the reward in (2) can be expressed as an expectation under the policy-induced expert posterior $q^{(\pi)}$, corrected by a KL divergence term that quantifies the mismatch between the expert responsibilities inferred by the policy and those implied by the reward scores. Unlike in standard variational inference settings—where the variational posterior is freely optimized— $q^{(\pi)}$ is fully determined by the policy parameters $\{w_k, \pi_k\}$ and cannot be adjusted to match $q^{(r)}$. Consequently, the KL term does not vanish and must be retained as part of the objective. It plays a critical role in regularizing modular learning by penalizing structural misalignment between policy inference and reward modeling, thus enabling a principled treatment of expert specialization.

Next, we utilize the decomposition within Theorem 2 to provide the decomposition for the policy training objective in (3). Namely,

Lemma 3 (MoE-DPO Objective Decomposition). *The training objective in (3) can be written as*

$$\mathcal{L}_{\text{MoE-DPO}} = \sum_{k=1}^K \mathbb{E}_{x \sim p, y \sim \pi_k(\cdot \mid x)} \left[w_k(x) \left(\tilde{r}_k(x, y) - \beta \log \frac{\pi_k(y \mid x)}{\pi_{\text{ref}(k)}(y \mid x)} \right) \right]. \quad (9)$$

where $\tilde{r}_k(x, y) = r_k(x, y) - \log \left(\frac{q_k^{(r)}(x, y)}{w_k(x)} \right)$.

The extra term in $\tilde{r}_k(x, y)$ adjusts the reward to account for discrepancies between the gating distribution $w_k(x)$ and the posterior responsibilities $q_k^{(r)}(x, y)$ induced by the reward model. This term acts as a localized KL penalty that encourages alignment between policy-induced expert selection and reward-based expert attribution, promoting consistency between modular inference

and modular supervision. This formulation admits closed-form updates for each expert policy π_k , facilitating scalable and modular optimization. Specifically, the per-expert KL-regularized objective becomes:

$$\mathcal{L}_k(\pi_k) = \mathbb{E}_{x \sim p} \left[w_k(x) \mathbb{E}_{y \sim \pi_k(y|x)} \left[\tilde{r}_k(x, y) - \beta \log \frac{\pi_k(y|x)}{\pi_{\text{ref}(k)}(y|x)} \right] \right],$$

and the optimal solution takes the form detailed in the next theorem.

Theorem 4 (Policy-Reward Equivalence under Mixture Models). *The expert policy that maximizes $\mathcal{L}_k(\pi_k)$ is given by:*

$$\pi_k^*(y|x) = \frac{1}{Z'_k(x)} \pi_{\text{ref}(k)}(y|x) \exp \left(\frac{1}{\beta} r_k(x, y) \right),$$

where the normalizing constant is defined as $Z_k(x) = \sum_y \pi_{\text{ref}(k)}(y|x) \exp \left(\frac{1}{\beta} \tilde{r}_k(x, y) \right)$.

This closed-form update mirrors the structure of the standard DPO solution while incorporating expert-specific responsibilities and reward corrections. It supports independent optimization of each expert, making the overall training procedure efficient and amenable to parallelization.

Since the latent variable z is shared across the preference-alignment and policy-optimization models, we can leverage the closed-form correspondence between the expert-specific reward $r_k(x, y)$ and the optimal policy $\pi_k(y|x)$, as established in Theorem 4. Substituting the reward expression,

$$r_k(x, y) = \beta \log \left(\frac{\pi_k(y|x) Z_k(x)}{\pi_{\text{ref}(k)}(y|x)} \right) + \log \left(\frac{q_k^{(r)}(x, y)}{w_k(x)} \right), \quad (10)$$

into the MBT variational bound in (6) yields a reward-modulated loss that is decomposable across expert heads. Specifically, we express the total MBT loss as $\mathcal{L}_{\text{MBT}} = \sum_{k=1}^K \mathcal{L}_k^{\text{MBT}}(\pi_k)$, where each expert-specific objective $\mathcal{L}_k^{\text{MBT}}$ is given by

$$\mathcal{L}_k^{\text{MBT}}(\pi_k) = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[q_k(x, y^+, y^-) \log \frac{\left(\frac{\pi_k(y^+|x)}{\pi_{\text{ref}(k)}(y^+|x)} \right)^\beta q_k^{(r)}(x, y^+)}{\sum_{\eta \in \{y^+, y^-\}} \left(\frac{\pi_k(\eta|x)}{\pi_{\text{ref}(k)}(\eta|x)} \right)^\beta q_k^{(r)}(x, \eta)} \right], \quad (11)$$

where $q_k(x, y^+, y^-)$ is defined in Theorem 1. This decomposition isolates the contribution of each expert head π_k , allowing for fully decoupled and independent optimization of π_k parameters. However, note that the original DPO model and training objective from [1] are recovered when $K = 1$.

Finally, to derive the optimization objective for the prior weights $w_k(x)$, we first note that, from Theorem 1, the only dependence of the ELBO on $w_k(x)$ appears in the KL divergence term between the variational posterior $q_k(x, y^+, y^-)$ and the prior $w_k(x)$. Consequently, we show in the Appendix how maximizing the ELBO with respect to $w_k(x)$ is equivalent to the minimization

$$\arg \min_{w_k(x)} \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\sum_{k=1}^K q_k(x, y^+, y^-) \log \frac{q_k(x, y^+, y^-)}{w_k(x)} \right]. \quad (12)$$

This objective encourages weights to match the empirical expert responsibilities inferred from preference data, ensuring that the prior over experts reflects their posterior under the MBT model. Next, we consider two variants of our algorithm depending on the structure of the mixture weights.

Mix-DPO. The optimal weights $w_k(x)$ are fixed across inputs, and their update from (12) has a closed form given by averaging the responsibilities: $w_k \leftarrow \frac{1}{n} \sum_{i=1}^n q_k(x_i, y_i^+, y_i^-)$, and then renormalizing.

MoE-DPO. The weights are input-dependent and parameterized by a gating network $w_k(x; \phi)$ via a softmax over logits. The parameters ϕ are updated by minimizing the objective in (12), which reduces to the cross-entropy between predicted weights and inferred posteriors: $\mathcal{L}_{\text{gating}}(\phi) = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} [\sum_k q_k(x, y^+, y^-) \log w_k(x; \phi)]$. The gating function can also be conditioned on user metadata $u \in \mathcal{U}$, enabling personalized mixture weights $w_k(x, u; \phi)$ without modifying the expert policies.

3 Algorithm

Building on the variational framework and expert-specific loss decomposition in Section 2, we now present the training procedure for optimizing modular preference-aligned policies. As shown in Corollary 1.1 and Theorem 4, the MoE-DPO objective admits a per-expert decomposition, where each expert policy π_k is trained by minimizing its own KL-regularized preference loss $\mathcal{L}_k^{\text{MBT}}$. To ensure consistency between the policy and reward components, expert rewards are updated after each policy step via the closed-form transformation in (10) derived from the optimality condition.

Algorithm 1 summarizes the resulting variational EM procedure. Each iteration alternates between: (i) an E-step that computes expert responsibilities $q_k(x, y^+, y^-)$ using the MBT posterior in Theorem 1; (ii) an M-step that updates each expert policy π_k by minimizing the loss $\mathcal{L}_k^{\text{MBT}}$ in (11); (iii) a reward update via the closed-form expression in (10); and (iv) an update of the mixture weights $w_k(x)$ based on inferred responsibilities and the objective function in (12). This modular structure allows flexible optimization regimes: depending on the setting, the expert policies, the gating network, or both can be updated jointly or independently within each iteration.

Algorithm 1 Variational EM Algorithm for Mixture Preference Alignment

Require: Expert policies and rewards $\{\pi_k, r_k\}_{k=1}^K$, mixture weights $w_k(x)$, reference policies $\pi_{\text{ref}(k)}$, temperature β .

- 1: **while** not converged **do**
- 2: **Sample minibatch:** $\{(x_i, y_i^+, y_i^-)\}_{i=1}^n \sim \mathcal{D}$.
- 3: **if** Trainable expert policies **then**
- 4: **E-step:** Compute posteriors $q_k(x_i, y_i^+, y_i^-)$ from rewards r_k and weights $w_k(x)$.
- 5: **M-step:** Update each π_k using q_k , $\pi_{\text{ref}(k)}$, and temperature β .
- 6: **end if**
- 7: **Reward update:** Update each $r_k(x, y)$ using Eq. (10).
- 8: **if** Trainable mixture weights **then**
- 9: **E-step:** Recompute responsibilities $q_k(x_i, y_i^+, y_i^-)$.
- 10: **M-step:** Update gating network parameters ϕ (MoE-DPO) or fixed weights (Mix-DPO) w_k .
- 11: **end if**
- 12: **end while**

This algorithm implements the full variational EM cycle over preference-labeled data. Expert policies and rewards are jointly updated in a way that maintains the policy–reward correspondence (Theorem 4), while the mixture weights adapt to match inferred responsibilities. The structure supports parallelization across experts and modular personalization via user-conditioned gating.

4 Experimental Evaluation

This section details the experimental evaluation of our proposed Mix- and MoE-DPO methods against baseline DPO. We first present two primary experiments on the alignment of GPT-2 [15] for the review generation task [1], each exploring variations of the task: (1) a multi-reward task, where the model is trained to generate positive, informative, and grammatically correct movie reviews, and (2) a contextual preference-alignment task, where the model is trained to generate either positive movie or book reviews based on the given prompt. We provide ablation studies to assess the impact of fixed mixture weights or expert policies, which we include as an optional simplification within Algorithm 1.

4.1 Mix-DPO for Multi-Reward Movie Review Generation

Models: We apply Mix-DPO for preference alignment to the three different reward functions of sentiment, informativeness, and grammar using a supervised-fine-tuned GPT-2 [15] on the IMDB [16] dataset as the base and reference policy. We evaluate two configurations: Case 1 uses a single GPT-2 with three heads (final linear layers) for parameter-efficient sharing, and Case 2 employs three copies of the fine-tuned GPT-2 model. **Datasets:** We construct a pairwise preference dataset from IMDB similar to [1], where we first sample completions for the 25,000 prompts in

the training dataset. Preferences over these responses are annotated with respect to sentiment, informativeness, and grammar, with further details on our reward-function construction included in the Appendix. **Algorithms:** Mix-DPO minimizes a weighted loss across mixture components with learnable weights initialized at $1/3$ for both cases. For Case 1 (parameter-efficient sharing), training proceeds for 3 epochs, while for Case 2 (independent models), training proceeds for 1 epoch, both with a learning rate of 10^{-5} , batch size of 64, and the AdamW optimizer. **Metrics:** We generate completions from 10,000 prompts in the test set and compute reward scores corresponding to each ground-truth reward function. Completions can be generated via individual mixture components or the entire mixture, with both sets of metrics reported in Table 3, in addition to baseline DPO.

Results: During training of Mix-DPO, we log the posterior weights $q_k(x_i, y_i^+, y_i^-)$ corresponding to each head and annotation style. In Figure 1 (left panel), we see a clear separation of q -weights in each head based on annotation style, indicating specialization of heads for the distinct reward functions of sentiment, informativeness, and grammar. Specifically, we observe that, averaged over data from the corresponding annotation style, the posterior weights diverge over training, with Head 0 decreasing to approximately 0.31, Head 1 stabilizing around 0.34, and Head 2 increasing to 0.35.

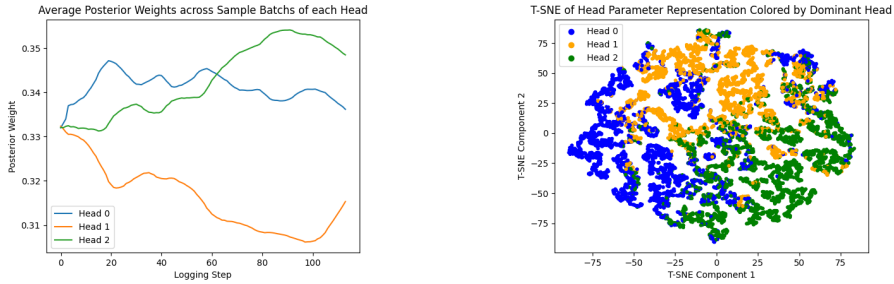


Figure 1: Left panel: Average posterior weights for Mix-DPO heads—(a) Head 0, (b) Head 1, and (c) Head 2—indicates specialization. **Right panel:** t-SNE plot of head-parameter representations indicates head separation.

On the test set, Table 3 shows that Case 1 (Mixture effectiveness in aligning LLMs) outperforms baseline DPO across sentiment (0.654 ± 0.004 vs. 0.610 ± 0.004) and grammar (0.241 ± 0.001 vs. 0.216 ± 0.001), but underperforms on informativeness (0.326 ± 0.007 vs. 0.363 ± 0.008). Ablation studies in Table 3 show that fixed weights improve over DPO but do not perform as well as Mix-DPO when averaging heads.

Table 1: Reward scores (Mean \pm SE) for Mix-DPO on the IMDb test set.

Model	Sentiment	Informativeness	Grammar
Baseline DPO	0.610 ± 0.004	0.363 ± 0.008	0.216 ± 0.001
Case 1 (Mixture)	0.654 ± 0.004	0.326 ± 0.007	0.241 ± 0.001
Case 1 (Sparse)			
Head 0	0.616 ± 0.020	0.396 ± 0.007	0.263 ± 0.001
Head 1	0.720 ± 0.003	0.394 ± 0.008	0.213 ± 0.001
Head 2	0.632 ± 0.004	0.342 ± 0.007	0.267 ± 0.001
Case 2 (Mixture)	0.664 ± 0.004	0.350 ± 0.006	0.465 ± 0.002
Fixed Weights ($w_i = 1/3$)	0.646 ± 0.004	0.318 ± 0.009	0.239 ± 0.002

A word cloud of sampled terms from the heads (Figure 2) highlights their specialization: Head 0—strong in informativeness (0.396 ± 0.007) and grammar (0.263 ± 0.001), but weaker in positive sentiment (0.616 ± 0.002)—produces terms like “Lebowski,” “Solomon,” and “average”; Head 1—excelling in positive sentiment (0.720 ± 0.003) and informativeness (0.394 ± 0.008)—generates terms like “funny,” “good,” and “Cusack,” reflecting a focus on emotional tone; and Head 2—with the highest grammar score (0.267 ± 0.001)—focuses on syntactic structure with terms like “movie,” “interesting,” and “story.”



Figure 2: Sampled words from responses generated by Mix-DPO heads in Case 1: (a) **Head 0** (Informativeness and Grammar), (b) **Head 1** (Positive Sentiment and Informativeness), and (c) **Head 2** (Grammar).

4.2 MoE-DPO for Multi-Task Review Generation

Models: We apply MoE-DPO for preference alignment to two different user groups (i.e., tasks) of book reviews and movie reviews. Using the same GPT-2 base model [15], we evaluate our algorithm in the case of parameter sharing (Case 1) and independent models (Case 2). For MoE-DPO, we utilize prompt-dependent weights via a pretrained linear classifier—i.e., $w(x) = Wx + b$ —which can be frozen or jointly learned during training. **Datasets:** We augment the IMDb dataset [16] with 25,000 pairwise preferences from Amazon Book Reviews, a subset of the Amazon review dataset [17], forming a 50,000-pair dataset with prompts labeled for movie or book reviews. Further details on the book-review dataset can be found in the Appendix. We pretrain a linear classifier for 5,000 steps on the prompts to achieve approximately 65% accuracy for the true source label. **Algorithms:** We employ the mixture-of-experts loss for both cases, and in joint learning we alternate between the two losses $\mathcal{L}_{\text{MoE-DPO}}$ and $\mathcal{L}_{\text{gating}}$. Training proceeds for 3 epochs with a learning rate of 10^{-5} , batch size of 64, and the AdamW optimizer. **Metrics:** We generate completions for 10,000 test-set prompts, equally split between 5,000 book and 5,000 movie reviews. Reward scores are computed with the ground-truth sentiment reward function.

Results: The performance of MoE-DPO across different configurations is summarized in Table 2. In Table 2, Case 1 (Frozen Gating Layer) achieves a sentiment reward on the movie task of 0.638 ± 0.005 , on par with the learnable gating layer at 0.639 ± 0.005 . For the book task, a learnable gating layer outperforms, with 0.734 ± 0.004 compared to 0.709 ± 0.004 achieved with a frozen gating layer. Both Case 1 configurations outperform baseline DPO (0.603 ± 0.005 on movie sentiment and 0.648 ± 0.005 on book sentiment), indicating that improved multi-task alignment can be achieved via our mixture model.

Table 2: Sentiment reward scores (Mean \pm SE) for MoE-DPO.

Model	Movie Sentiment	Book Sentiment
Baseline DPO	0.603 ± 0.005	0.648 ± 0.005
Case 1 (Frozen Gating Layer)	0.638 ± 0.005	0.709 ± 0.004
Case 1 (Learnable Gating Layer)	0.639 ± 0.005	0.734 ± 0.004

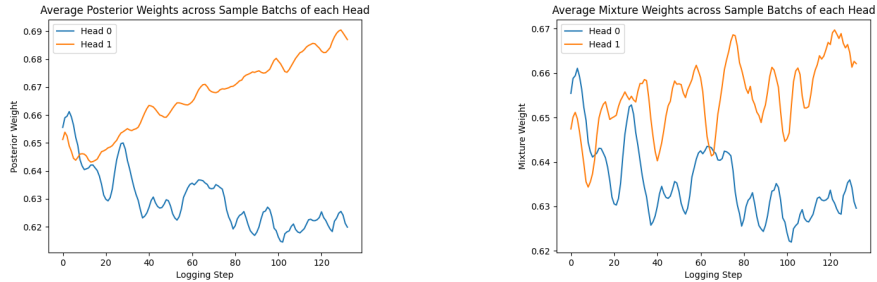


Figure 3: **Left panel:** Average posterior weights for MoE-DPO heads in Case 1—(a) **Head 0** and (b) **Head 1**—indicates specialization. **Right panel:** Average mixture weights for MoE-DPO indicates prompt separation.

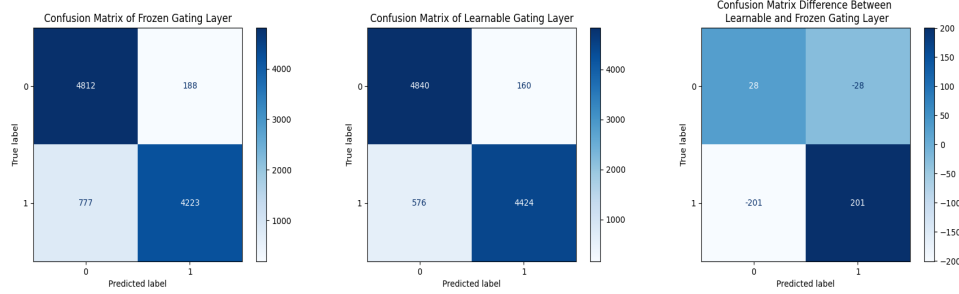


Figure 4: Confusion matrices of frozen and learnable gating layers for movie (0) vs. book (1) prompts, with the difference indicating improvements in predicted labels under joint learning during training.

5 Conclusion

Mix- and MoE-DPO provides a unified approach to modular preference alignment. It generalizes DPO with latent expert mixtures, supports efficient variational training, and enables expert reuse, contextual adaptation, and user-specific specialization—advancing the practical and theoretical foundation for aligning LLMs. Beyond the core formulation presented in this paper, the proposed variational framework supports several extensions that broaden its applicability and scalability. These include (i) user-personalized gating, (ii) multi-agent temporal modeling via hidden Markov mixtures, (iii) multimodal alignment through structured group actions, and (iv) scalable training via sparse expert activation and differentiable relaxation (Monte Carlo relaxation of the variational EM). These extensions make Mix- and MoE-DPO well suited for real-world deployment in (i) multi-user, (ii) multi-agent, (iii) multimodal, and (iv) large-scale training settings, respectively.

References

- [1] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 2023.
- [2] 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*, 2017.
- [3] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 2022.
- [4] Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. RLAIIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv preprint arXiv:2309.00267*, 2023.
- [5] Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 2024.
- [6] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Beta-DPO: Direct preference optimization with dynamic beta. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, 2024.
- [7] Tetsuro Morimura, Mitsuki Sakamoto, Yuu Jinnai, Kenshi Abe, and Kaito Ariu. Filtered direct preference optimization. *arXiv preprint arXiv:2404.13846*, 2024.
- [8] Yuzhe Gu, Wenwei Zhang, Chengqi Lyu, Dahua Lin, and Kai Chen. Mask-DPO: Generalizable fine-grained factuality alignment of LLMs. *arXiv preprint arXiv:2503.02846*, 2025.

- [9] Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. *arXiv preprint arXiv:2406.14868*, 2024.
- [10] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-DPO: Step-wise preference optimization for long-chain reasoning of LLMs. *arXiv preprint arXiv:2406.18629*, 2024.
- [11] Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.
- [12] Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in RLHF: Your SFT loss is implicitly an adversarial regularizer. *arXiv preprint arXiv:2405.16436*, 2024.
- [13] Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024.
- [14] Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, et al. The perfect blend: Redefining RLHF with mixture of judges. *arXiv preprint arXiv:2409.20370*, 2024.
- [15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. Accessed: 2024-11-15.
- [16] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea, editors, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
- [17] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.
- [18] Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J Foster. Correcting the mythos of KL-regularization: Direct alignment without overoptimization via chi-squared preference optimization. *arXiv preprint arXiv:2407.13399*, 2024.
- [19] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.
- [20] Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.
- [21] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online AI feedback. *arXiv preprint arXiv:2402.04792*, 2024.
- [22] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *Proceedings of the International Conference on Learning Representations*, 2017.
- [23] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 1951.
- [24] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the International Conference on Learning Representations*, 2017.
- [25] Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [26] Jan Ludziejewski, Maciej Pióro, Jakub Krajewski, Maciej Stefaniak, Michał Krutul, Jan Małaśnicki, Marek Cygan, Piotr Sankowski, Kamil Adamczewski, Piotr Miłoś, et al. Joint MoE scaling laws: Mixture of experts can be memory efficient. *arXiv preprint arXiv:2502.05172*, 2025.
- [27] Luca Falorsi, Patrick De Haan, Tim Davidson, Tristan Deleu, Patrick Forré, Jonathan Gordon, and Charles Blundell. Reparameterizing distributions on lie groups. In *22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

A Related Work

Recent extensions have built upon the direct optimization approach of DPO to further improve effectiveness of aligning LLMs with human preferences. Improved robustness and efficiency are achieved with methods that dynamically adjust β to handle noisy preference data, or filter low-quality preference pairs to strengthen optimization [6, 7]. The sample-efficient DPO extension of χ^2 -based preference optimization has been proposed to mitigate over-optimization in [18]. Robustness has been further improved by integrating the supervised fine-tuning loss as an implicit adversarial regularizer, or by modifying the underlying preference objective to mitigate overfitting and preference shifts, respectively, in [5, 12]. Additional extensions of DPO include [19–21].

The precision and applicability of DPO have also been advanced through targeted extensions. Fine-grained factuality alignment and error reduction in multi-turn dialogues are achieved by optimizing sentence-level preferences and adapting DPO for conversational settings, respectively [8, 9]. Alternative structures—such as reasoning steps or token-level alignment—have been integrated into direct optimization approaches in [10, 11]. These contributions collectively enhance DPO’s robustness, efficiency, precision, and applicability in addressing alignment challenges.

B Architectural Variants of Mix- and MoE-DPO

To accommodate diverse deployment scenarios, we consider two primary architectural instantiations of the Mix- and MoE-DPO framework. These variants differ in the degree of parameter sharing among expert policies, balancing expressivity, memory efficiency, and specialization. Both designs are compatible with the core variational formulation in Section 2 and the training algorithm in Section 3.

B.1 Case 1: Shared Encoder with Expert-Specific Heads

This parameter-efficient variant builds on a modular architecture in which all experts share a common encoder $f_\phi(x, y)$, parameterized by ϕ , that produces a joint representation of the input–output pair. Each expert k is defined by a head-specific transformation h_{ψ_k} producing logits for the conditional distribution:

$$\pi_k(y \mid x) = \text{softmax}(h_{\psi_k}(f_\phi(x, y))).$$

The mixture policy is given by the gating-weighted average:

$$\pi(y \mid x) = \sum_{k=1}^K w_k(x) \cdot \pi_k(y \mid x),$$

where the gating weights $\{w_k(x)\}$ form a probability simplex over experts, as in (1). KL regularization is applied using a shared reference policy $\pi_{\text{ref}(k)} = \pi_{\text{ref}}$ for all experts, consistent with (3).

This design offers favorable memory and compute scaling by replacing K independent policies with one shared encoder and K lightweight heads. Optimization can proceed jointly over $(\phi, \{\psi_k\}, \{w_k\})$, or in a partially frozen regime where ϕ is fixed and only the heads and gating parameters are trained:

$$\mathcal{L}_{\text{MoE-DPO}}(\{\psi_k\}, \{w_k\} \mid \phi).$$

This setup aligns with the multi-head specialization strategy in Section 4 and supports modular adaptation over diverse tasks without duplicating the full backbone.

B.2 Case 2: Fully Independent Experts

In this more expressive variant, each expert policy $\pi_k(y \mid x)$ is independently parameterized with its own encoder, decoder, and reward function $r_k(x, y)$. The gating function $w_k(x)$ may be fixed (as in Mix-DPO) or learned (as in MoE-DPO), and each expert may have a distinct reference policy $\pi_{\text{ref}(k)}$. This structure supports heterogeneity in both modeling and supervision, allowing for maximal task-specific adaptation.

The decomposition of the MoE-DPO loss into per-expert objectives, as shown in (11), allows for fully decoupled optimization:

$$\mathcal{L}_{\text{MoE-DPO}} = \sum_{k=1}^K \mathcal{L}_k^{\text{MBT}}(\pi_k),$$

with each π_k updated independently. This independence facilitates training under domain shifts, user-specific feedback distributions, or multi-objective alignment, as explored in Section 4. The associated reward updates and posterior responsibilities follow the closed-form expressions derived in (10) and (7), respectively.

B.3 Hybrid Architectures and Dynamic Expert Routing

The generality of Mix- and MoE-DPO supports hybrid configurations that combine shared and independent components. For instance:

- A shared encoder with K_s expert heads (Case 1) may be combined with K_i independently parameterized experts (Case 2), allowing coarse-grained specialization through routing.
- A mixture-of-mixtures architecture can be used in which gating weights distinguish between shared and independent blocks, and the overall policy is a convex combination across both groups.
- Personalized or task-conditioned routing can be implemented by conditioning $w_k(x)$ on metadata, enabling efficient multi-user adaptation (cf. Appendix E).

Such hybrid structures offer favorable trade-offs between parameter reuse and specialization, making them well suited for deployment under memory constraints, on-device adaptation, or transfer learning from expert checkpoints. The modularity of our variational framework ensures compatibility with these extensions, including user-aware gating, sparse activation, and scalable training, as discussed in Appendix E and supported by the convergence analysis in Appendix F.

C Proofs

Lemma 5. Let $a_1, \dots, a_K \in \mathbb{R}_{>0}$ be positive real numbers, and let $q \in \Delta_K$ be any probability distribution over $\{1, \dots, K\}$, where

$$\Delta_K := \left\{ q \in \mathbb{R}^K \mid q_k \geq 0, \sum_{k=1}^K q_k = 1 \right\}.$$

Then the following identity holds:

$$\log \sum_{k=1}^K a_k = \sum_{k=1}^K q_k \log \left(\frac{a_k}{q_k} \right) + D_{\text{KL}} \left(q \parallel \frac{a}{\sum_j a_j} \right),$$

where $\frac{a}{\sum_j a_j} \in \Delta_K$ is the normalized vector with components

$$p_k := \frac{a_k}{\sum_j a_j}, \quad \text{and} \quad D_{\text{KL}}(q \parallel p) = \sum_k q_k \log \frac{q_k}{p_k}.$$

Proof. Let $a_k > 0$ for each $k \in \{1, \dots, K\}$, and fix any $q \in \Delta_K$. Define the normalized version of a as

$$p_k := \frac{a_k}{\sum_j a_j} \in \Delta_K.$$

Then

$$\log \sum_k a_k = \log \left(\sum_k q_k \cdot \frac{a_k}{q_k} \right).$$

Apply Jensen’s inequality to the concave logarithm function:

$$\log \left(\sum_k q_k \cdot \frac{a_k}{q_k} \right) \geq \sum_k q_k \log \left(\frac{a_k}{q_k} \right),$$

with equality if and only if $q_k \propto a_k$, i.e., $q = p$. Now rewrite:

$$\begin{aligned} \log \sum_k a_k &= \sum_k q_k \log \left(\frac{a_k}{q_k} \right) = \sum_k q_k \left[\log \left(\frac{a_k}{p_k} \cdot \frac{p_k}{q_k} \right) \right] \\ &= \sum_k q_k \left[\log \left(\frac{a_k}{p_k} \right) + \log \left(\frac{p_k}{q_k} \right) \right] = \sum_k q_k \log \left(\frac{a_k}{p_k} \right) - \sum_k q_k \log \left(\frac{q_k}{p_k} \right) \\ &= \sum_k q_k \log \left(\frac{a_k}{p_k} \right) + D_{\text{KL}}(q \| p). \end{aligned}$$

But $\frac{a_k}{p_k} = \sum_j a_j$, so the first term becomes

$$\sum_k q_k \log \left(\sum_j a_j \right) = \log \left(\sum_j a_j \right),$$

since $\sum_k q_k = 1$. Thus,

$$\log \sum_k a_k = \sum_k q_k \log \left(\frac{a_k}{q_k} \right) + D_{\text{KL}}(q \| p),$$

which completes the proof. \square

Proof of Theorem 1. We aim to derive a variational lower bound (ELBO) on the log-likelihood of observing a preference $y^+ \succ y^-$ given a prompt x , under the Mixture-of-Bradley–Terry (MBT) model.

The marginal likelihood under the MBT model is

$$\mathbb{P}(y^+ \succ y^- | x) = \sum_{k=1}^K w_k(x) \sigma_k(x, y^+, y^-),$$

where $w_k(x) = p(z = k | x)$ is the prior mixture weight, and the expert-specific Bradley–Terry likelihood is

$$\sigma_k(x, y^+, y^-) := \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))}.$$

Let $q(z | x, y^+, y^-)$ be any variational distribution over the latent expert index $z \in \{1, \dots, K\}$, and let $q_k(x, y^+, y^-) := q(z = k | x, y^+, y^-)$. Then

$$\begin{aligned} \log \mathbb{P}(y^+ \succ y^- | x) &= \log \sum_{k=1}^K w_k(x) \sigma_k(x, y^+, y^-) \\ &= \log \sum_{k=1}^K q_k(x, y^+, y^-) \cdot \frac{w_k(x) \sigma_k(x, y^+, y^-)}{q_k(x, y^+, y^-)} \\ &\geq \sum_{k=1}^K q_k(x, y^+, y^-) \log \left(\frac{w_k(x) \sigma_k(x, y^+, y^-)}{q_k(x, y^+, y^-)} \right), \end{aligned}$$

where the inequality follows from Jensen’s inequality applied to the concave logarithm function.

This is the standard form of the evidence lower bound (ELBO), and it can be rewritten as

$$\sum_{k=1}^K q_k(x, y^+, y^-) \log \sigma_k(x, y^+, y^-) - D_{\text{KL}}(q(\cdot | x, y^+, y^-) \| p(\cdot | x)).$$

Thus, we obtain the lower bound

$$\log \mathbb{P}(y^+ \succ y^- | x) \geq \mathbb{E}_{z \sim q(\cdot | x, y^+, y^-)} [\log \sigma_z(x, y^+, y^-)] - D_{\text{KL}}(q(z | x, y^+, y^-) \| p(z | x)).$$

Tightness of the bound. The inequality becomes an equality when the variational posterior $q(z \mid x, y^+, y^-)$ matches the true posterior $p(z \mid x, y^+, y^-)$. Using Bayes' rule, the exact posterior under the MBT model is

$$\begin{aligned} p(z = k \mid x, y^+, y^-) &= \frac{p(z = k \mid x) \mathbb{P}(y^+ \succ y^- \mid x, z = k)}{\mathbb{P}(y^+ \succ y^- \mid x)} \\ &= \frac{w_k(x) \sigma_k(x, y^+, y^-)}{\sum_{j=1}^K w_j(x) \sigma_j(x, y^+, y^-)}. \end{aligned}$$

Therefore, the bound is tight when

$$q_k(x, y^+, y^-) = \frac{w_k(x) \sigma_k(x, y^+, y^-)}{\sum_{j=1}^K w_j(x) \sigma_j(x, y^+, y^-)}.$$

□

Proof of Theorem 2. Recall the definition of the mixture reward function from (2):

$$r(x, y) := \log \sum_{k=1}^K w_k(x) \exp(r_k(x, y)).$$

We apply Lemma 5 using $a_k = w_k(x) \exp(r_k(x, y))$ and setting $q_k = q_k^{(\pi)}(x, y)$. This gives

$$\begin{aligned} r(x, y) &= \sum_{k=1}^K q_k^{(\pi)}(x, y) \left[r_k(x, y) + \log w_k(x) - \log q_k^{(\pi)}(x, y) \right] \\ &\quad + \sum_{k=1}^K q_k^{(\pi)}(x, y) \log \left(\frac{q_k^{(\pi)}(x, y)}{q_k^{(r)}(x, y)} \right) \quad \left(\text{recall } q_k^{(r)}(x, y) = \frac{w_k(x) \exp(r_k(x, y))}{\sum_j w_j(x) \exp(r_j(x, y))} \right) \\ &= \sum_{k=1}^K q_k^{(\pi)}(x, y) \left[r_k(x, y) + \log w_k(x) - \log q_k^{(\pi)}(x, y) \right] + D_{\text{KL}}(q^{(\pi)} \parallel q^{(r)}), \end{aligned} \quad (13)$$

which matches the decomposition in (8). Therefore,

$$r(x, y) = \mathbb{E}_{z \sim q^{(\pi)}(\cdot \mid x, y)} \left[r_z(x, y) + \log w_z(x) - \log q_z^{(\pi)}(x, y) \right] + D_{\text{KL}}(q^{(\pi)} \parallel q^{(r)}).$$

□

Proof of Lemma 3. We begin from the MoE-DPO objective in (3):

$$\mathcal{L}_{\text{MoE-DPO}} := \mathbb{E}_{x \sim p, y \sim \pi(y \mid x)} [r(x, y)] - \beta \sum_{k=1}^K \mathbb{E}_{x \sim p} [w_k(x) D_{\text{KL}}(\pi_k(y \mid x) \parallel \pi_{\text{ref}(k)}(y \mid x))],$$

where $\pi(y \mid x) = \sum_{k=1}^K w_k(x) \pi_k(y \mid x)$ and the soft-mixture reward is defined by (2):

$$r(x, y) := \log \sum_{k=1}^K w_k(x) \exp(r_k(x, y)).$$

Apply Theorem 2 to obtain

$$r(x, y) = \sum_{k=1}^K q_k^{(\pi)}(x, y) \left[r_k(x, y) + \log w_k(x) - \log q_k^{(\pi)}(x, y) \right] + D_{\text{KL}}(q^{(\pi)} \parallel q^{(r)}), \quad (14)$$

where

$$q_k^{(\pi)}(x, y) = \frac{w_k(x) \pi_k(y \mid x)}{\pi(y \mid x)}, \quad \text{and} \quad \sum_k q_k^{(\pi)}(x, y) = 1.$$

Taking expectation over $x \sim p, y \sim \pi(y | x)$ gives

$$\begin{aligned} \mathbb{E}_{x \sim p, y \sim \pi(y|x)}[r(x, y)] &= \sum_{k=1}^K \mathbb{E}_{x \sim p, y \sim \pi(y|x)} \left[q_k^{(\pi)}(x, y) \left(r_k(x, y) + \log w_k(x) - \log q_k^{(\pi)}(x, y) \right) \right] \\ &\quad + \mathbb{E}_{x \sim p, y \sim \pi(y|x)} \left[D_{\text{KL}}(q^{(\pi)} \| q^{(r)}) \right]. \end{aligned}$$

Hence,

$$\mathbb{E}_{x \sim p, y \sim \pi(y|x)}[r(x, y)] = \sum_{k=1}^K \mathbb{E}_{x \sim p, y \sim \pi(y|x)} \left[q_k^{(\pi)}(x, y) \left(r_k(x, y) + \log w_k(x) - \log q_k^{(r)}(x, y) \right) \right].$$

This motivates defining the corrected reward

$$\tilde{r}_k(x, y) := r_k(x, y) - \log \left(\frac{q_k^{(r)}(x, y)}{w_k(x)} \right).$$

Now use the identity $q_k^{(\pi)}(x, y) \pi(y | x) = w_k(x) \pi_k(y | x)$, which implies

$$\mathbb{E}_{x \sim p, y \sim \pi(y|x)} \left[q_k^{(\pi)}(x, y) f_k(x, y) \right] = \mathbb{E}_{x \sim p, y \sim \pi_k(y|x)} [w_k(x) f_k(x, y)]$$

for any measurable function f_k . Applying this to $\tilde{r}_k(x, y)$ gives

$$\mathbb{E}_{x \sim p, y \sim \pi(y|x)} [r(x, y)] = \sum_{k=1}^K \mathbb{E}_{x, y \sim \pi_k(y|x)} [w_k(x) \tilde{r}_k(x, y)].$$

Combining with the expert-specific KL penalties in the original objective, we conclude

$$\mathcal{L}_{\text{MoE-DPO}} = \sum_{k=1}^K \mathbb{E}_{x \sim p, y \sim \pi_k(y|x)} \left[w_k(x) \left(\tilde{r}_k(x, y) - \beta \log \frac{\pi_k(y | x)}{\pi_{\text{ref}(k)}(y | x)} \right) \right],$$

which completes the proof. \square

Proof of Theorem 4. We seek the form of the expert policy $\pi_k(y | x)$ that maximizes the per-expert objective:

$$\mathcal{L}_k(\pi_k) = \mathbb{E}_{x \sim p} [\mathbb{E}_{y \sim \pi_k(y|x)} [\tilde{r}_k(x, y)] - \beta D_{\text{KL}}(\pi_k(\cdot | x) \| \pi_{\text{ref}(k)}(\cdot | x))],$$

where $\tilde{r}_k(x, y)$ is the adjusted reward, and $\pi_{\text{ref}(k)}(y | x)$ is a fixed reference policy.

This objective is additive over x , so it suffices to maximize the inner functional pointwise for each x . Fix $x \in \mathcal{X}$ and define

$$\pi(y) := \pi_k(y | x), \quad \pi_{\text{ref}}(y) := \pi_{\text{ref}(k)}(y | x), \quad \tilde{r}(y) := \tilde{r}_k(x, y).$$

We now solve the constrained optimization problem

$$\max_{\pi \in \Delta_{\mathcal{Y}}} \left\{ \sum_{y \in \mathcal{Y}} \pi(y) \tilde{r}(y) - \beta \sum_{y \in \mathcal{Y}} \pi(y) \log \frac{\pi(y)}{\pi_{\text{ref}}(y)} \right\},$$

subject to $\sum_y \pi(y) = 1$, where $\Delta_{\mathcal{Y}}$ denotes the probability simplex over the discrete output space \mathcal{Y} .

Introduce a Lagrange multiplier $\lambda \in \mathbb{R}$ for the normalization constraint and define the Lagrangian

$$\mathcal{L}(\pi, \lambda) = \sum_y \pi(y) \tilde{r}(y) - \beta \sum_y \pi(y) \log \frac{\pi(y)}{\pi_{\text{ref}}(y)} - \lambda \left(\sum_y \pi(y) - 1 \right).$$

Taking derivatives with respect to $\pi(y)$ gives

$$\frac{\partial \mathcal{L}}{\partial \pi(y)} = \tilde{r}(y) - \beta \left(\log \frac{\pi(y)}{\pi_{\text{ref}}(y)} + 1 \right) - \lambda.$$

Setting this to zero yields

$$\pi(y) = \pi_{\text{ref}}(y) \exp\left(\frac{1}{\beta} (\tilde{r}(y) - \lambda - \beta)\right).$$

Let Z be the normalizing constant ensuring $\sum_y \pi(y) = 1$. Then

$$\pi(y) = \frac{1}{Z} \pi_{\text{ref}}(y) \exp\left(\frac{1}{\beta} \tilde{r}(y)\right), \quad Z := \sum_{y'} \pi_{\text{ref}}(y') \exp\left(\frac{1}{\beta} \tilde{r}(y')\right).$$

Thus, the optimal expert policy is

$$\pi_k^*(y \mid x) = \frac{1}{Z_k(x)} \pi_{\text{ref}(k)}(y \mid x) \exp\left(\frac{1}{\beta} \tilde{r}_k(x, y)\right),$$

where $Z_k(x) = \sum_{y'} \pi_{\text{ref}(k)}(y' \mid x) \exp\left(\frac{1}{\beta} \tilde{r}_k(x, y')\right)$. The solution is unique because the objective is strictly concave in π (linear term minus KL), and the feasible set $\Delta_{\mathcal{Y}}$ is convex and compact. \square

Proof that maximizing the ELBO with respect to $w_k(x)$ is equivalent to minimizing (12). We show that maximizing the ELBO with respect to the gating weights $w_k(x)$ is equivalent to minimizing the expected KL divergence between the variational posterior and the gating prior, as given in (12).

Recall the ELBO from Theorem 1, which for a single preference triplet (x, y^+, y^-) is

$$\log \mathbb{P}(y^+ \succ y^- \mid x) \geq \sum_{k=1}^K q_k(x, y^+, y^-) \log \left(\frac{w_k(x) \sigma_k(x, y^+, y^-)}{q_k(x, y^+, y^-)} \right),$$

where $\sigma_k(x, y^+, y^-)$ is the Bradley–Terry likelihood under expert k , and $q_k(x, y^+, y^-)$ is any variational posterior over the expert index z .

Isolate the part of the ELBO that depends on $w_k(x)$. Since $\sigma_k(x, y^+, y^-)$ and $q_k(x, y^+, y^-)$ are fixed (as outputs of the E-step), the only term depending on $w_k(x)$ is

$$\sum_{k=1}^K q_k(x, y^+, y^-) \log w_k(x).$$

Thus, maximizing the ELBO over $w_k(x)$, subject to $\sum_{k=1}^K w_k(x) = 1$, is equivalent to solving

$$\max_{\{w_k(x)\}} \sum_{k=1}^K q_k(x, y^+, y^-) \log w_k(x) \quad \text{subject to} \quad \sum_k w_k(x) = 1, w_k(x) \geq 0.$$

This is equivalent to minimizing

$$D_{\text{KL}}(q(\cdot \mid x, y^+, y^-) \parallel w(\cdot \mid x)) = \sum_{k=1}^K q_k(x, y^+, y^-) \log \frac{q_k(x, y^+, y^-)}{w_k(x)},$$

which attains its minimum when $w_k(x) = q_k(x, y^+, y^-)$.

Taking the expectation over $(x, y^+, y^-) \sim \mathcal{D}$ yields the training objective

$$\arg \min_{w_k(x)} \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\sum_{k=1}^K q_k(x, y^+, y^-) \log \frac{q_k(x, y^+, y^-)}{w_k(x)} \right],$$

which is precisely (12). Hence, maximizing the ELBO with respect to $w_k(x)$ is equivalent to minimizing the expected KL divergence between the variational posterior and the gating prior. \square

D Stochastic EM Algorithm for Mix- and MoE-DPO

This appendix details the stochastic Expectation–Maximization (EM) procedure used to train the Mix- and MoE-DPO models. Rather than operating on the full dataset, each iteration of the algorithm is applied to a randomly sampled minibatch, enabling scalable training on large preference datasets. The stochastic EM loop alternates between soft expert assignment (E-step) and parameter optimization (M-step), and comprises the following four stages:

1. **E-step:** Posterior-responsibility computation over latent experts using the MBT model.
2. **M-step:** Policy update via log-normalized preference gradients.
3. **Reward update:** Reward recomputation for posterior consistency.
4. **M-step:** Mixture-prior update using minibatch-responsibility averages.

Each step is performed on a minibatch $\{(x_i, y_i^+, y_i^-)\}_{i=1}^n$ of size n , where the preference triplets are drawn from the full training dataset \mathcal{D} .

D.1 E-Step: Posterior-Responsibility Computation for Preference Triplets

We compute the posterior responsibilities $q_k(x, y^+, y^-)$ over the latent expert index $z \in \{1, \dots, K\}$ for each triplet in the minibatch. These responsibilities are derived from the expert-specific Bradley–Terry likelihoods and the current mixture weights:

Algorithm 2 Posterior-Responsibility Computation for the MBT Model

Require: Rewards $\{r_k(x, y^+), r_k(x, y^-)\}_{k=1}^K$, mixture weights $\{w_k(x)\}_{k=1}^K$.

Ensure: Posterior probabilities $\{q_k(x, y^+, y^-)\}_{k=1}^K$.

1: **for** $k = 1, \dots, K$ **do**

2: Compute preference likelihood:

$$p_k \leftarrow \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))}.$$

3: Compute unnormalized responsibility:

$$\tilde{q}_k \leftarrow w_k(x) \cdot p_k.$$

4: **end for**

5: Normalize responsibilities:

$$Z \leftarrow \sum_{j=1}^K \tilde{q}_j, \quad q_k(x, y^+, y^-) \leftarrow \frac{\tilde{q}_k}{Z}.$$

return $\{q_k(x, y^+, y^-)\}_{k=1}^K$.

D.2 M-Step: Policy Update via Log-Space Normalized Gradient

Expert policies π_k are updated by maximizing the componentwise MBT objective using stochastic gradients. The loss for expert k is derived from the log-normalized likelihood ratio:

$$\mathcal{L}_k = -\mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[q_k(x, y^+, y^-) \cdot \log \left(\frac{A^+}{A^+ + A^-} \right) \right],$$

where we define

$$\begin{aligned} A^+ &= \pi_k(y^+ | x)^\beta \cdot q_k^{(r)}(x, y^+) \cdot \pi_{\text{ref}(k)}(y^+ | x)^{-\beta}, \\ A^- &= \pi_k(y^- | x)^\beta \cdot q_k^{(r)}(x, y^-) \cdot \pi_{\text{ref}(k)}(y^- | x)^{-\beta}, \end{aligned}$$

and $\beta > 0$ is the inverse temperature.

Gradient derivation. We differentiate \mathcal{L}_k with respect to θ_k , the parameters of π_k :

$$\begin{aligned}\nabla_{\theta_k} \mathcal{L}_k &= -\mathbb{E}_{(x, y^+, y^-)} \left[q_k(x, y^+, y^-) \cdot \nabla_{\theta_k} \log \left(\frac{A^+}{A^+ + A^-} \right) \right] \\ &= -\mathbb{E}_{(x, y^+, y^-)} \left[q_k(x, y^+, y^-) \cdot \left(\frac{1}{A^+} \nabla_{\theta_k} A^+ - \frac{1}{A^+ + A^-} (\nabla_{\theta_k} A^+ + \nabla_{\theta_k} A^-) \right) \right].\end{aligned}$$

Using

$$\nabla_{\theta_k} A^+ = \beta A^+ \nabla_{\theta_k} \log \pi_k(y^+ | x), \quad \nabla_{\theta_k} A^- = \beta A^- \nabla_{\theta_k} \log \pi_k(y^- | x),$$

we substitute into the gradient:

$$\begin{aligned}\nabla_{\theta_k} \mathcal{L}_k &= -\beta \mathbb{E}_{(x, y^+, y^-)} [q_k(x, y^+, y^-) \cdot (\nabla_{\theta_k} \log \pi_k(y^+ | x) - \frac{A^+}{A^+ + A^-} \nabla_{\theta_k} \log \pi_k(y^+ | x) \\ &\quad - \frac{A^-}{A^+ + A^-} \nabla_{\theta_k} \log \pi_k(y^- | x))] \\ &= \beta \mathbb{E}_{(x, y^+, y^-)} \left[q_k(x, y^+, y^-) \cdot \left(\frac{A^+}{A^+ + A^-} \nabla_{\theta_k} \log \pi_k(y^- | x) - \frac{A^-}{A^+ + A^-} \nabla_{\theta_k} \log \pi_k(y^+ | x) \right) \right].\end{aligned}$$

Stochastic gradient estimator. On a minibatch $\{(x_i, y_i^+, y_i^-)\}_{i=1}^n$, the gradient is approximated as

$$\widehat{\nabla}_{\theta_k} \mathcal{L}_k^{\text{MBT}} = \frac{\beta}{n} \sum_{i=1}^n q_k(x_i, y_i^+, y_i^-) \left[\frac{A_i^+}{A_i^+ + A_i^-} \nabla_{\theta_k} \log \pi_k(y_i^- | x_i) - \frac{A_i^-}{A_i^+ + A_i^-} \nabla_{\theta_k} \log \pi_k(y_i^+ | x_i) \right],$$

with

$$\begin{aligned}A_i^+ &= \pi_k(y_i^+ | x_i)^\beta \cdot q_k^{(r)}(x_i, y_i^+) \cdot \pi_{\text{ref}(k)}(y_i^+ | x_i)^{-\beta}, \\ A_i^- &= \pi_k(y_i^- | x_i)^\beta \cdot q_k^{(r)}(x_i, y_i^-) \cdot \pi_{\text{ref}(k)}(y_i^- | x_i)^{-\beta}.\end{aligned}$$

This gradient is used to update θ_k via standard optimizers such as SGD or Adam.

D.3 Reward Update: Reward Recalibration for Posterior Consistency

Rewards $r_k(x, y)$ are updated to enforce agreement between the reward-induced and policy-induced posteriors, using only the current minibatch:

Algorithm 3 Reward Update for Posterior Reweighting

Require: Current policy $\pi_k(y | x)$, reference policy $\pi_{\text{ref}(k)}(y | x)$, responsibilities $q_k^{(r)}(x, y)$, mixture weights $w_k(x)$, temperature β .

Ensure: Updated reward values $r_k(x_i, y_i)$ for $y_i \in \{y_i^+, y_i^-\}$.

- 1: **for** each expert $k = 1, \dots, K$ **do**
- 2: **for** each input x_i in the minibatch **do**
- 3: Compute the partition function:

$$Z_k(x_i) = \sum_{j=1}^n \pi_{\text{ref}(k)}(y_j | x_i) \exp \left(\frac{1}{\beta} \left(r_k(x_i, y_j) + \log w_k(x_i) - \log q_k^{(r)}(x_i, y_j) \right) \right).$$

- 4: **for** each $y_i \in \{y_i^+, y_i^-\}$ **do**
- 5: Update the reward:

$$r_k(x_i, y_i) = \beta \cdot \log \left(\frac{\pi_k(y_i | x_i) Z_k(x_i)}{\pi_{\text{ref}(k)}(y_i | x_i)} \right) + \log \left(\frac{q_k^{(r)}(x_i, y_i)}{w_k(x_i)} \right).$$

- 6: **end for**
 - 7: **end for**
 - 8: **end for**
-

D.4 M-Step: Mixture-Prior Update (Batch-wise)

Mixture priors $w_k(x)$ are updated using the minibatch-specific posterior responsibilities. The update differs between Mix-DPO and MoE-DPO.

Mix-DPO (input-independent weights). The mixture weights $w_k \in \Delta^{K-1}$ are global parameters. The minibatch update is

$$w_k \leftarrow \frac{1}{n} \sum_{i=1}^n q_k(x_i, y_i^+, y_i^-),$$

where n is the minibatch size. This estimate replaces the full-batch average and reflects the current soft-assignment statistics over the minibatch.

MoE-DPO (input-dependent gating). The gating function $w_k(x) = \text{softmax}(g_k(x))$ is trained by minimizing the minibatch-averaged KL divergence:

$$\min_{w_k(x)} \frac{1}{n} \sum_{i=1}^n \sum_{k=1}^K q_k(x_i, y_i^+, y_i^-) \log \frac{q_k(x_i, y_i^+, y_i^-)}{w_k(x_i)}.$$

This corresponds to supervised learning over inputs x_i with soft labels $\{q_k(x_i, y_i^+, y_i^-)\}$, implemented via cross-entropy loss on the gating logits.

D.5 Numerical Considerations

All computations are performed on minibatches to support scalable training. We apply the log-sum-exp trick to evaluate partition functions and normalizers for numerical stability. Gradients are computed via automatic differentiation and optimized with standard stochastic optimizers.

E Mix- and MoE-DPO at Scale

The Mix- and MoE-DPO framework supports a range of scalable training regimes, offering flexibility across modeling capacity, inference fidelity, and computational constraints. Depending on the deployment setting, different components of the model can be emphasized or fixed during optimization. For instance, when expert policies $\{\pi_k\}_{k=1}^K$ are initialized from strong zero-shot models (e.g., domain-specialized LLMs), they can be held fixed while rewards are computed using the canonical DPO transformation. In such cases, only the gating network, which outputs mixture weights $\{w_k(x)\}_{k=1}^K$, is trained—including for a personalized gating function with user-specific characteristics $u \in \mathcal{U}$. This reduces the procedure to updating the gating function via the mixture M-step of a variational EM algorithm, with the E-step used to infer expert-assignment posteriors.

In contrast, the fully end-to-end optimization strategy updates the expert policies, gating network, and reward functions jointly. To support both modular and end-to-end regimes, we introduce two complementary estimation strategies. First, a *regularized variational inference* method imposes entropy- and KL-based penalties on the expert-assignment posterior, encouraging confident yet diverse expert selection while preserving a closed-form expression for the variational distribution. Second, a *Monte Carlo relaxation* leverages the Gumbel–Softmax reparameterization trick [22] to enable differentiable sampling over the latent expert index. Both approaches support efficient training with minibatches and eliminate the need for explicit (full-sample) E-step computations in classical EM. The variational strategy stabilizes the E-step by enforcing structure on the posterior, while the Monte Carlo relaxation bypasses it entirely by enabling gradient-based updates through sampled expert assignments. Practitioners can instantiate expert policies as either fixed modules or fully trainable heads, depending on the complexity of the problem, computational resources, and adaptability requirements. Our estimation strategies are compatible with large-scale minibatch training and can scale to billions of parameters without incurring dense-model costs.

The framework supports sparse activation, modular reuse, and personalized alignment, making it suitable for deployment in multi-task, multi-user, and resource-constrained settings.

E.1 Regularized Variational Responsibilities for the MBT Model

We extend the Mix- and MoE-DPO framework by introducing a richer regularization structure specifically targeting the MBT posterior distribution $q_k(x, y^+, y^-)$. The modified regularized MBT

loss is given by:

$$\begin{aligned} \mathcal{L}_{\text{MBT}}^{\text{reg}}(\{q_k\}, \{\pi_k\}, \{w_k\}) = & \sum_{k=1}^K \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[q_k(x, y^+, y^-) \ell_k(x, y^+, y^-) \right. \\ & + \lambda_{\text{ent}} \mathcal{H}(q(\cdot | x, y^+, y^-)) \\ & - \lambda_{\text{conf}} \mathcal{H}(q(\cdot | x, y^+, y^-)) \\ & - \lambda_{\text{KL-unif}} D_{\text{KL}}(q(\cdot | x, y^+, y^-) \| \text{Uniform}(\{1, \dots, K\})) \\ & - \lambda_{\text{KL-w}} D_{\text{KL}}(q(\cdot | x, y^+, y^-) \| w(\cdot | x)) \\ & \left. - \lambda_{\text{KL-w-global}} \mathbb{E}_{x \sim p(x)} [D_{\text{KL}}(w(\cdot | x) \| \text{Uniform}(\{1, \dots, K\}))] \right], \end{aligned}$$

where the per-component utility is

$$\ell_k(x, y^+, y^-) := \log \frac{\left(\frac{\pi_k(y^+ | x)}{\pi_{\text{ref}(k)}(y^+ | x)} \right)^\beta q_k^{(r)}(x, y^+)}{\sum_{\eta \in \{y^+, y^-\}} \left(\frac{\pi_k(\eta | x)}{\pi_{\text{ref}(k)}(\eta | x)} \right)^\beta q_k^{(r)}(x, \eta)},$$

and $\beta, \lambda_{\text{ent}}, \lambda_{\text{conf}}, \lambda_{\text{KL-unif}}, \lambda_{\text{KL-w}}, \lambda_{\text{KL-w-global}}$ control the strength of the respective regularizers.

Each regularizer acts specifically on $q_k(x, y^+, y^-)$ with distinct roles at various training phases:

- β : Controls deviation from reference policies $\pi_{\text{ref}(k)}$. Active throughout, typically decaying slowly.
- λ_{ent} : Promotes exploration via high entropy in responsibilities $q_k(x, y^+, y^-)$. Critical early in training.
- λ_{conf} : Drives specialization via low entropy, enforcing confident expert assignments. Crucial in later training phases.
- $\lambda_{\text{KL-unif}}$: Prevents early collapse onto few experts, regularizing toward uniformity. Emphasized early.
- $\lambda_{\text{KL-w}}$: Aligns posterior responsibilities $q_k(x, y^+, y^-)$ with prior mixture weights $w_k(x)$. Useful when priors are informative.
- $\lambda_{\text{KL-w-global}}$: Regularizes global mixture weights toward uniformity. Strong early, relaxed later.

Note that the extended MBT objective introduces additional regularization terms on the variational posterior, which modify the computation of $q_k(x, y^+, y^-)$ for each expert. Namely,

Lemma 6 (Optimal Variational Posterior under Regularized MBT Objective). *Assuming fixed expert policies $\{\pi_k\}$, mixture weights $\{w_k(x)\}$, and MBT reward structure $\{q_k^{(r)}(x, y)\}$, the optimal variational posterior $q_k(x, y^+, y^-)$ that maximizes the regularized MBT objective*

$$\begin{aligned} \mathcal{L}_{\text{MBT}}^{\text{reg}}(q) = & \sum_{k=1}^K \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[q_k(x, y^+, y^-) \log \frac{\left(\frac{\pi_k(y^+ | x)}{\pi_{\text{ref}(k)}(y^+ | x)} \right)^\beta q_k^{(r)}(x, y^+)}{\sum_{\eta \in \{y^+, y^-\}} \left(\frac{\pi_k(\eta | x)}{\pi_{\text{ref}(k)}(\eta | x)} \right)^\beta q_k^{(r)}(x, \eta)} \right. \\ & + \lambda_{\text{ent}} \mathcal{H}(q(\cdot | x, y^+, y^-)) - \lambda_{\text{conf}} \mathcal{H}(q(\cdot | x, y^+, y^-)) \\ & - \lambda_{\text{KL-unif}} D_{\text{KL}}(q(\cdot | x, y^+, y^-) \| \text{Uniform}(\{1, \dots, K\})) \\ & \left. - \lambda_{\text{KL-w}} D_{\text{KL}}(q(\cdot | x, y^+, y^-) \| w(\cdot | x)) \right] \end{aligned}$$

is given by the normalized distribution

$$q_k(x, y^+, y^-) = \frac{1}{Z(x, y^+, y^-)} (w_k(x))^{\lambda_{\text{KL-w}}/\alpha} \exp \left(\frac{1}{\alpha} \log \frac{\left(\frac{\pi_k(y^+ | x)}{\pi_{\text{ref}(k)}(y^+ | x)} \right)^\beta q_k^{(r)}(x, y^+)}{\sum_{\eta \in \{y^+, y^-\}} \left(\frac{\pi_k(\eta | x)}{\pi_{\text{ref}(k)}(\eta | x)} \right)^\beta q_k^{(r)}(x, \eta)} \right),$$

where the effective temperature is

$$\alpha := \lambda_{\text{conf}} - \lambda_{\text{ent}} + \lambda_{\text{KL-unif}} + \lambda_{\text{KL-w}},$$

and $Z(x, y^+, y^-)$ ensures normalization.

Proof of Lemma 6. We maximize the integrand of the regularized MBT objective at each fixed (x, y^+, y^-) . Define the per-component utility as

$$\ell_k(x, y^+, y^-) := \log \frac{\left(\frac{\pi_k(y^+ | x)}{\pi_{\text{ref}(k)}(y^+ | x)} \right)^\beta q_k^{(r)}(x, y^+)}{\sum_{\eta \in \{y^+, y^-\}} \left(\frac{\pi_k(\eta | x)}{\pi_{\text{ref}(k)}(\eta | x)} \right)^\beta q_k^{(r)}(x, \eta)}.$$

Grouping entropy and KL-based regularization terms, introduce

$$\alpha := \lambda_{\text{conf}} - \lambda_{\text{ent}} + \lambda_{\text{KL-unif}} + \lambda_{\text{KL-w}}.$$

The simplified optimization objective becomes

$$\mathcal{L}(q) = \sum_{k=1}^K q_k \left[\ell_k + \lambda_{\text{KL-w}} \log w_k(x) - \lambda_{\text{KL-unif}} \log K - \alpha \log q_k \right] + \text{const.}$$

Introducing a Lagrange multiplier λ for the simplex constraint $\sum_k q_k = 1$, setting derivatives to zero, and exponentiating yields

$$q_k(x, y^+, y^-) = \frac{1}{Z(x, y^+, y^-)} (w_k(x))^{\lambda_{\text{KL-w}}/\alpha} \exp \left(\frac{\ell_k(x, y^+, y^-)}{\alpha} \right).$$

Normalization ensures $\sum_k q_k(x, y^+, y^-) = 1$, completing the proof. \square

These responsibilities, utilized throughout the EM procedure, reflect a regularized optimization that balances reward alignment with entropy control and KL constraints to priors, specifically the uniform distribution and the mixture weights $w_k(x)$. While the updates of the expert policies π_k and the mixture weights $w_k(x)$ maintain their structural form—being explicit functions of these responsibilities—the responsibilities $q_k(x, y^+, y^-)$ themselves must replace their non-regularized counterparts across all EM algorithm steps. Consequently, this revised formulation preserves the modular decomposition of EM updates but mandates the use of refined, regularized responsibilities in both the M-step policy updates and the estimation of mixture weights.

For the training strategy, we recommend that (i) only a subset of the regularizers is activated during specific training phases; and (ii) typically, only one entropy-based regularizer (λ_{ent} or λ_{conf}) is employed at any given time.

We suggest dynamically adapting the regularization scheme across training phases to achieve a balance between exploration, specialization, and stability:

Phase	Objective	Active Regularizers
Early (Exploration)	Maintain diversity, avoid expert collapse	$\lambda_{\text{ent}}, \lambda_{\text{KL-unif}}, \lambda_{\text{KL-w-global}}$
Middle (Specialization)	Encourage confident expert specialization	$\lambda_{\text{conf}}, \text{moderate } \beta$
Late (Stabilization)	Solidify learned specialization, maintain stability	$\lambda_{\text{conf}}, \lambda_{\text{KL-w}}, \text{low } \beta$

During the early training phase, high-entropy and uniformity regularizers encourage broad expert participation. In the middle phase, confidence-promoting regularizers sharpen the variational responsibilities and encourage expert specialization. In the late phase, regularization terms are adjusted to maintain stability and finalize expert specialization based on learned structures.

E.2 Monte Carlo Relaxation for Mixture of Bradley–Terry Estimation

To eliminate the explicit E-step in the variational EM algorithm for the Mixture of Bradley–Terry (MBT) model, we adopt a Monte Carlo relaxation based on the Gumbel–Softmax reparameterization, which allows replacing hard assignments (e.g., selecting one expert via $\arg\max$) with soft, differentiable assignments during training. Instead of computing the closed-form posterior responsibilities $q_k(x, y^+, y^-)$, we approximate the latent expert assignment $z \in \{1, \dots, K\}$ by sampling from a continuous relaxation of the categorical distribution.

Let $\alpha_k(x, y^+, y^-)$ denote the unnormalized expert logit scores,

$$\alpha_k(x, y^+, y^-) \propto w_k(x) \cdot \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))},$$

and define $\log \alpha(x, y^+, y^-) \in \mathbb{R}^K$ as the vector of logits across components. We draw L samples from the Gumbel–Softmax distribution with temperature $\tau > 0$,

$$z^{(l)} \sim \text{GumbelSoftmax}(\log \alpha(x, y^+, y^-), \tau), \quad l = 1, \dots, L,$$

i.e.,

$$z_k = \frac{\exp((\log w_k(x) + g_k)/\tau)}{\sum_{j=1}^K \exp((\log w_j(x) + g_j)/\tau)}, \quad \text{where } g_k \sim \text{Gumbel}(0, 1).$$

Hence $z^{(l)} \in \Delta^{K-1}$ lies in the continuous simplex, and these soft assignments act as differentiable surrogates for $q_k(x, y^+, y^-)$. We replace the original MBT loss in (5) with a Monte Carlo estimate of the relaxed objective:

$$\mathcal{L}_{\text{MC}} = -\frac{1}{L} \sum_{l=1}^L \sum_{k=1}^K z_k^{(l)} \log \left(\frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))} \right).$$

Expressing the sample z as a deterministic, differentiable function of $w_k(x)$ and auxiliary random noise g_k enables low-variance gradient estimation through the sampling step.

The KL divergence between the variational posterior and the prior $\text{KL}(q(z | x, y^+, y^-) \| p(z | x))$ can also be approximated using relaxed samples:

$$\widehat{\text{KL}} = \frac{1}{L} \sum_{l=1}^L \sum_{k=1}^K z_k^{(l)} \log \left(\frac{z_k^{(l)}}{w_k(x)} \right).$$

Finally, policy parameters θ_k , rewards $r_k(x, y)$, and gating-network weights $w_k(x)$ are updated using these relaxed estimates without requiring a closed-form E-step. This enables fully **end-to-end** training using backpropagation under the Mix- and MoE-DPO frameworks while preserving a variational lower bound on the marginal likelihood.

Compatibility with General Architectures. Monte Carlo relaxation extends seamlessly to architectures where the MoE structure is embedded within transformer layers or other neural architectures, without requiring explicit policy factorization. In such settings:

- Mixture weights $w_k(x)$ are learned via differentiable gating networks.
- Expert rewards $r_k(x, y)$ can be computed via head activations or contrastive objectives.
- The relaxed latent variable z enables sparse, input-adaptive routing.

This flexibility makes Monte Carlo relaxation a method of choice in high-capacity deployments, allowing scalable and differentiable training across diverse preference tasks.

F Convergence of the Mix- and MoE-DPO Estimation

We report sufficient conditions for the convergence of the Mix- and MoE-DPO estimation procedures under two algorithmically distinct training paradigms:

1. A variational EM algorithm with closed-form expert responsibilities, alternating between inference over latent variables and maximization of a variational lower bound [23].

2. A Monte Carlo relaxation approach based on the Gumbel–Softmax trick [22, 24], which enables fully differentiable, end-to-end optimization over latent expert assignments.

While the two methods differ in their treatment of latent variables and optimization flow, they share a unified objective: maximization of a stochastic evidence lower bound (ELBO) under potentially noisy minibatch updates. To this end, we specify a set of shared assumptions on smoothness, boundedness, and variance control that guarantee convergence to stationary points of the ELBO, as detailed in Sections F.2 and F.3.

F.1 Shared Assumptions and General Conditions

Let θ denote the model parameters (e.g., expert policies $\{\pi_k\}$, gating weights $\{w_k(x)\}$), and let $\mathcal{L}(\theta)$ denote a (possibly relaxed) ELBO objective.

General Assumptions for Convergence

- **(Smoothness)** The objective $\mathcal{L}(\theta)$ is continuously differentiable in θ , and its gradient $\nabla_{\theta}\mathcal{L}(\theta)$ is Lipschitz continuous on compact subsets.
- **(Boundedness)** The ELBO objective is finite over the admissible domain:

$$-\infty < \mathcal{L}(\theta) < \infty \quad \text{for all } \theta \in \Theta.$$

- **(Coercivity)** The objective is coercive, i.e.,

$$\|\theta\| \rightarrow \infty \quad \Rightarrow \quad \mathcal{L}(\theta) \rightarrow -\infty.$$

- **(Stochastic Approximation)** If gradient updates are computed via stochastic samples, we assume:
 - Robbins–Monro step size conditions [23]: $\sum_t \eta_t = \infty, \sum_t \eta_t^2 < \infty$.
 - Unbiased stochastic gradient estimates.
 - Finite variance of the gradient noise:

$$\mathbb{E}\|\widehat{\nabla}_{\theta}\mathcal{L}(\theta) - \nabla_{\theta}\mathcal{L}(\theta)\|^2 \leq C(1 + \|\theta\|^2).$$

These assumptions are standard in stochastic variational inference and are generally satisfied in the Mix- and MoE-DPO framework under practical architectural and training choices.

Smoothness holds naturally in our model due to the use of softmax parameterizations for policies and gating networks, which are differentiable with Lipschitz-continuous gradients over compact parameter domains. When using Gumbel–Softmax relaxations for latent-variable inference, the objective remains smooth as long as the temperature τ is kept strictly positive during training. Potential violations can occur if gating functions degenerate into hard assignments prematurely ($\tau \rightarrow 0$), which may introduce sharp, non-smooth transitions.

Boundedness of the ELBO is preserved by the finite output space of softmax policies and regularization through KL terms. All terms in the ELBO remain bounded as long as the expert policies do not collapse to delta distributions disjoint from their reference policies. This risk is mitigated through KL regularization and initialization from well-calibrated base models. In practice, numerical instability is avoided by ensuring overlap in support between π_k and $\pi_{\text{ref}(k)}$, especially in early training.

Coercivity is enforced by the KL-divergence regularization, which grows superlinearly as the expert policy deviates from its reference. This ensures that unbounded parameter growth leads to penalization in the ELBO. However, coercivity may be weakened if the gating network assigns vanishing weights $w_k(x) \approx 0$ to specific experts, effectively nullifying their KL terms. This emphasizes the importance of entropy control in the gating mechanism to prevent expert collapse or neglect.

Stochastic Approximation assumptions are supported by minibatch-based training, uniform data sampling, and unbiased reparameterization gradients in the Gumbel–Softmax case. The Robbins–Monro conditions on learning-rate decay are satisfied via standard schedules. Finite gradient variance is generally ensured by using moderate temperatures ($\tau > 0.5$), sufficient samples

per minibatch, and normalized reward magnitudes. Variance may increase if reward values are poorly scaled or if the number of latent samples L is too small in the Monte Carlo setting.

In summary, these assumptions are valid under standard training setups for Mix- and MoE-DPO. Caution is warranted in highly sparse, low-temperature, or degenerate expert-routing regimes, which may require annealing strategies, sample averaging, or additional regularization to maintain convergence guarantees.

F.2 Case 1: Variational EM Algorithm

In the variational EM setting, we consider a variational posterior $q(z \mid x, y^+, y^-) \in \Delta^{K-1}$ and maximize the ELBO:

$$\mathcal{L}(q, \theta) = \mathbb{E}_q[\log p(y^+, y^-, z \mid x, \theta)] - \mathbb{E}_q[\log q(z \mid x, y^+, y^-)].$$

This objective is optimized alternately via:

- **E-step:** Maximize $\mathcal{L}(q, \theta^{(t)})$ with respect to q while holding $\theta^{(t)}$ fixed;
- **M-step:** Maximize $\mathcal{L}(q^{(t+1)}, \theta)$ with respect to θ while holding $q^{(t+1)}$ fixed.

Convergence Theorem (Variational EM). Under the smoothness, boundedness, and coercivity assumptions defined in Section F.1, the sequence of variational EM updates satisfies

$$\nabla_{\theta} \mathcal{L}(q^{(t)}, \theta^{(t)}) \rightarrow 0, \quad q^{(t)} \rightarrow q^*, \quad \theta^{(t)} \rightarrow \theta^*,$$

i.e., the sequence converges to a stationary point of the ELBO objective. This follows from standard convergence results for coordinate ascent applied to variational EM [23].

Proof. The Mix- and MoE-DPO model is trained by maximizing the marginal likelihood of preference comparisons under a latent-variable mixture model:

$$\mathbb{P}(y^+ \succ y^- \mid x) = \sum_{k=1}^K w_k(x) \cdot \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))},$$

where the latent variable $z \in \{1, \dots, K\}$ denotes the expert index, $w_k(x)$ is the input-dependent prior over experts, and $r_k(x, y)$ is the reward function for expert k .

We consider the variational EM algorithm that maximizes the evidence lower bound (ELBO) for the marginal likelihood of preference observations. Each iteration t consists of:

E-step: Update the variational posterior $q^{(t+1)}(z = k \mid x, y^+, y^-)$ using

$$q_k^{(t+1)}(x, y^+, y^-) = \frac{w_k^{(t)}(x) \cdot \frac{\exp(r_k^{(t)}(x, y^+))}{\exp(r_k^{(t)}(x, y^+)) + \exp(r_k^{(t)}(x, y^-))}}{\sum_{j=1}^K w_j^{(t)}(x) \cdot \frac{\exp(r_j^{(t)}(x, y^+))}{\exp(r_j^{(t)}(x, y^+)) + \exp(r_j^{(t)}(x, y^-))}},$$

which minimizes the KL divergence to the true posterior and hence increases the ELBO.

M-step: Perform stochastic gradient ascent on the ELBO:

$$\theta^{(t+1)} = \theta^{(t)} + \eta_t \nabla_{\theta} \mathcal{L}(q^{(t+1)}, \theta^{(t)}),$$

where θ denotes the collection of model parameters, including $\{\pi_k\}$, $\{w_k\}$, and potentially shared parameters, and η_t satisfies the Robbins–Monro conditions $\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$.

Step 1 (Monotonic improvement). The E-step yields a strict increase (or no decrease) in the ELBO due to Jensen’s inequality:

$$\mathcal{L}(q^{(t+1)}, \theta^{(t)}) \geq \mathcal{L}(q^{(t)}, \theta^{(t)}).$$

The M-step, via stochastic gradient ascent, increases $\mathbb{E}[\mathcal{L}(q^{(t+1)}, \theta^{(t+1)})]$ in expectation.

Step 2 (Boundedness). By assumption, $\mathcal{L}(q, \theta)$ is bounded above. Hence, the sequence $\{\mathcal{L}(q^{(t)}, \theta^{(t)})\}_{t=1}^{\infty}$ is monotone increasing and bounded, and therefore converges.

Step 3 (Robbins–Monro convergence). Assuming bounded variance of stochastic gradients and that $\nabla_{\theta} \mathcal{L}$ is Lipschitz, the standard Robbins–Monro theorem implies

$$\nabla_{\theta} \mathcal{L}(q^{(t)}, \theta^{(t)}) \rightarrow 0 \quad \text{almost surely.}$$

Step 4 (Convergence to a stationary point). Since the ELBO is continuously differentiable in both arguments and $\mathcal{L}(q^{(t)}, \theta^{(t)})$ converges, we obtain

$$q^{(t)} \rightarrow q^*, \quad \theta^{(t)} \rightarrow \theta^*, \quad \text{with} \quad \nabla_{\theta} \mathcal{L}(q^*, \theta^*) = 0.$$

This implies that (q^*, θ^*) is a stationary point of the ELBO objective. \square

F.3 Case 2: Monte Carlo Relaxation

In this formulation, the latent responsibility $z \in \{1, \dots, K\}$ is approximated using a soft Gumbel–Softmax sample $z^{(l)} \in \Delta^{K-1}$. The training objective becomes

$$\widehat{\mathcal{L}}(\theta) = \frac{1}{L} \sum_{l=1}^L \log p(y^+, y^-, z^{(l)} | x; \theta) - \log p(z^{(l)} | x),$$

where $z^{(l)} \sim \text{GumbelSoftmax}(\log \alpha(x), \tau)$ [22, 24].

Convergence Theorem (Monte Carlo Relaxation). Under the smoothness, boundedness, coercivity, and stochastic approximation assumptions stated in Section F.1, the iterates $\theta^{(t)}$ produced by stochastic gradient ascent satisfy

$$\nabla_{\theta} \mathbb{E}[\widehat{\mathcal{L}}(\theta^{(t)})] \rightarrow 0 \quad \text{as } t \rightarrow \infty.$$

This result follows from stochastic approximation theory [23] combined with the reparameterization-gradient method for continuous relaxations [25].

Proof. Let $\mathcal{L}(\theta)$ denote the ELBO of the MBT-based Mix- and MoE-DPO model:

$$\mathcal{L}(\theta) = \mathbb{E}_{(x, y^+, y^-) \sim \mathcal{D}} \left[\log \sum_{k=1}^K w_k(x) \cdot \frac{\exp(r_k(x, y^+))}{\exp(r_k(x, y^+)) + \exp(r_k(x, y^-))} \right] - \text{KL regularization}.$$

Due to the latent discrete structure (expert index z), we introduce a continuous relaxation via the Gumbel–Softmax (Concrete) distribution to enable reparameterization.

Let $z \sim \text{RelaxedCategorical}(\tau, w(x))$ be the reparameterized latent variable, where τ is the temperature, and let $f(z; \theta)$ be the differentiable reparameterization of the ELBO integrand. Then the Monte Carlo estimate of the relaxed ELBO is

$$\widehat{\mathcal{L}}(\theta) = \frac{1}{M} \sum_{m=1}^M f(z^{(m)}; \theta), \quad z^{(m)} \sim \text{RelaxedCategorical}(\tau, w(x)),$$

which is an unbiased estimator of the relaxed objective:

$$\mathbb{E}[\widehat{\mathcal{L}}(\theta)] = \mathcal{L}_{\tau}(\theta),$$

where $\mathcal{L}_{\tau}(\theta)$ is the temperature-smoothed ELBO that converges to the exact ELBO as $\tau \rightarrow 0$.

(i) Unbiased gradient estimate. Using the reparameterization trick, we have

$$\nabla_{\theta} \mathbb{E}[\widehat{\mathcal{L}}(\theta)] = \mathbb{E}[\nabla_{\theta} f(z; \theta)],$$

and $\nabla_{\theta} f(z; \theta)$ is an unbiased estimator of $\nabla_{\theta} \mathcal{L}_{\tau}(\theta)$.

(ii) Robbins–Monro conditions. Assume the learning rates $\{\eta_t\}$ satisfy $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$.

(iii) Variance control. Suppose that the variance of the stochastic gradient estimator $\nabla_{\theta}\widehat{\mathcal{L}}(\theta^{(t)})$ is uniformly bounded, and the relaxed objective $\mathcal{L}_{\tau}(\theta)$ is continuously differentiable and satisfies the coercivity condition $\|\theta\| \rightarrow \infty \Rightarrow \mathcal{L}_{\tau}(\theta) \rightarrow -\infty$.

Then, by the Robbins–Monro theorem [23], the stochastic gradient ascent updates

$$\theta^{(t+1)} = \theta^{(t)} + \eta_t \nabla_{\theta} \widehat{\mathcal{L}}(\theta^{(t)})$$

converge to a stationary point of the relaxed objective:

$$\nabla_{\theta} \mathbb{E}[\widehat{\mathcal{L}}(\theta^{(t)})] = \nabla_{\theta} \mathcal{L}_{\tau}(\theta^{(t)}) \rightarrow 0 \quad \text{as } t \rightarrow \infty.$$

□

G Extensions

User Personalization. Our variational framework naturally supports personalized alignment by conditioning the gating function on user metadata $u \in \mathcal{U}$, yielding personalized mixture weights $w_k(x, u; \phi)$. This extension enables MoE-DPO to tailor expert allocation to individual user profiles without retraining expert policies. In settings where pretrained or fixed experts are available (e.g., domain-specific LLMs), personalization can be implemented by fine-tuning only the gating network, effectively leveraging modular reuse for scalable deployment. Our multi-task review generation experiment (Section 4) demonstrates this capability: even when experts are fixed, prompt-conditioned gating achieves significant improvements on user-specific reward metrics.

Scaling and Sparse Activation. Our training objective admits closed-form per-expert updates (Theorem 4) and modular decomposition of the ELBO, which makes MoE-DPO compatible with sparse expert activation. At inference time, only the top- S experts (with the highest gating scores) may be activated per input, significantly reducing compute. Empirically, we observe in Section 4 that even with a single routed expert per prompt (oracle routing), alignment quality improves across tasks, suggesting the potential for scalable deployment via sparse MoE architectures. Extensions to dynamic expert growth and token-to-expert allocation can leverage scaling laws developed in [26].

Differentiable Training via Monte Carlo Relaxation. While our method employs a classical variational EM algorithm, we recommend Monte Carlo relaxation for large-scale or continuous-group settings. Discrete expert assignments can be approximated using the Gumbel–Softmax reparameterization [22, 25], enabling differentiable training without explicit E-steps. For continuous groups, such as Lie group mixtures, exponential-map reparameterization [27] supports structured sampling and backpropagation. These techniques facilitate scalable training of MoE-DPO models with integrated routing and multimodal expert structures.

H Additional Experiment Details

We include additional experimental results on Case 2, which employs independent copies of GPT-2 models fine-tuned for particular reward signals. For training, we run Mix-DPO for 3 epochs with a learning rate of 10^{-5} , batch size of 64, and the AdamW optimizer. Average test metric values with standard errors are reported in the table below for average- or model-specific inference styles. We note that average results for independent policies outperform the parameter-sharing policy specification of Case 1 and offer comparable results in the test metrics for individual models.

H.1 Mix-DPO for Multi-Reward Movie Review Generation

Table 3: Reward scores (mean \pm SE) for Mix-DPO on the IMDB test set.

Model	Sentiment	Informativeness	Grammar
Baseline DPO	0.610 ± 0.004	0.363 ± 0.008	0.216 ± 0.001
Case 1 (Avg. Heads)	0.654 ± 0.004	0.326 ± 0.007	0.241 ± 0.001
Case 1 (Head-Specific)			
Head 0	0.616 ± 0.02	0.396 ± 0.007	0.263 ± 0.001
Head 1	0.720 ± 0.003	0.394 ± 0.008	0.213 ± 0.001
Head 2	0.632 ± 0.004	0.342 ± 0.007	0.267 ± 0.001
Fixed Weights ($w_i = 1/3$)	0.646 ± 0.004	0.318 ± 0.009	0.239 ± 0.002
Case 2 (Avg. Heads)	0.664 ± 0.004	0.350 ± 0.006	0.232 ± 0.002
Case 2 (Model-Specific)			
Model 0	0.747 ± 0.004	0.324 ± 0.007	0.232 ± 0.002
Model 1	0.686 ± 0.005	0.390 ± 0.003	0.198 ± 0.001
Model 2	0.638 ± 0.004	0.371 ± 0.006	0.243 ± 0.002