COS-DPO: Conditioned One-Shot Multi-Objective Fine-Tuning Framework

Yinuo Ren[1] **Tesi Xiao**[2] **Michael Shavlovsky**[2] **Lexing Ying**[3,1] **Holakou Rahmanian**[2]

[1]Institute for Computational and Mathematical Engineering (ICME), Stanford University
[2]Amazon
[3]Department of Mathematics, Stanford University

Abstract

In LLM alignment and many other ML applications, one often faces the *Multi-Objective Fine-Tuning* (MOFT) problem, *i.e.*, fine-tuning an existing model with datasets labeled w.r.t. different objectives simultaneously. To address the challenge, we propose a *Conditioned One-Shot* fine-tuning framework (COS-DPO) that extends the Direct Preference Optimization technique, originally developed for efficient LLM alignment with preference data, to accommodate the MOFT settings. By direct conditioning on the weight across auxiliary objectives, our Weight-COS-DPO method enjoys an efficient one-shot training process for profiling the Pareto front and is capable of achieving comprehensive trade-off solutions even in the post-training stage. Based on our theoretical findings on the linear transformation properties of the loss function, we further propose the Temperature-COS-DPO method that augments the temperature parameter to the model input, enhancing the flexibility of post-training control over the trade-offs between the main and auxiliary objectives. We demonstrate the effectiveness and efficiency of the COS-DPO framework through its applications to various tasks, including the Learning-to-Rank (LTR) and LLM alignment tasks, highlighting its viability for large-scale ML deployments.

1 INTRODUCTION

Direct Preference Optimization (DPO) [Rafailov et al., 2024b] has been introduced as a memory- and computation-efficient alternative to the traditional *Reinforcement Learning with Human Feedback (RLHF)* [Christiano et al., 2017, Stiennon et al., 2020, Ouyang et al., 2022] in Large Language Model (LLM) alignment. The method fine-tunes a pre-trained LLM with additional data that indicates the preference between different proposals w.r.t. customized objectives, such as safety, verbosity, coherence, *etc.* [Wu et al., 2024b]. The idea of DPO is to reparametrize the *reward function* in RLHF and guide the fine-tuning process in a supervised learning manner with the preference data.

LLM alignment also intersects with the *Multi-Objective Optimization* (MOO) problem, which involves fine-tuning a model w.r.t. multiple objectives simultaneously [Ji et al., 2024b, Wu et al., 2024b, Zhou et al., 2023, Rame et al., 2024]. Production machine learning-based ranking models must strike a careful balance among competing objectives such as relevance, diversity, fairness, and other business-driven goals. This challenge is especially acute in Amazon's All Product Search (APS), where stakeholder priorities often vary across query slices, product categories, and domains. Conventional approaches that supervise ranking models using a single aggregated loss with fixed preference weights struggle to adapt to such diverse and evolving requirements. Updating these weights for each query slice typically involves retraining the model with extensive hyper-parameter optimization (HPO), a process that is both time- and resource-intensive. A common scenario involves starting from a pre-trained base rankertrained on shared, global objectives across locales and segmentsand needing to align it with additional, slice-specific objectives provided by partner teams. Efficiently fine-tuning the base model to incorporate these localized desirability signals without full retraining and without significantly detracting the model's performance on the main objectives remains a key challenge in scalable multi-objective ranking. This specific scenario is termed the *Multi-Objective Fine-Tuning* (MOFT) problem. As auxiliary objectives may conflict with each other, the notion of alignment is generalized to achieving *Pareto optimality* in the MOFT setting, where the goal is to profile the *Pareto front*, representing a spectrum of trade-off solutions where no single auxiliary objective can be improved without compromising another.

In this work, we address the MOFT task in a broad context through our proposed COS-DPO framework. This conditioned one-shot multi-objective fine-tuning framework is designed to (1) generalize DPO to the MOFT setting, (2) profile the Pareto front of the auxiliary objectives while maintaining the model performance on the main objectives with an efficient one-shot training process, and (3) offer flexible post-training controls over the trade-offs. Our codebase is publicly available at `https://github.com/yinuoren/cosdpo`.

1.1 CONTRIBUTIONS

The main contributions of this work are as follows:

- We propose the COS-DPO method, a conditioned one-shot multi-objective fine-tuning framework that generalizes DPO to the multi-objective setting and profiles the Pareto front through one-shot training.
- We test our Weight-COS-DPO method across diverse tasks, including Learning-to-Rank (LTR) fine-tuning and LLM alignment, demonstrating its superior performance to achieve comprehensive Pareto fronts and its efficiency against existing baselines.
- Based on our theoretical findings, we propose a novel Temperature-COS-DPO method that enhances the flexibility of post-training control over the trade-offs between the main and auxiliary objectives.

For LLM applications, we also develop a novel *Hyper Prompt Tuning* design as an engineering contribution that translates the continuous vectors into a mask applied to the prefix embedding, conveying the importance weights assigned across auxiliary objectives to the LLM without altering its architecture.

1.2 RELATED WORKS

LLM Alignment. LLM alignment has been a popular topic in the machine learning community. RLHF has been a groundbreaking technique for alignment [Christiano et al., 2017, Schulman et al., 2017, Ouyang et al., 2022, Bai et al., 2022a], which serves as a foundation for training models like GPT-4 [Achiam et al., 2023], and several advances have been made in this direction [Dong et al., 2024, Bai et al., 2022b, Lee et al., 2023]. To reduce computational complexity, DPO [Rafailov et al., 2024b] has been proposed as an alternative to RLHF, and further developed in the literature [Pal et al., 2024, Wu et al., 2024a, Gheshlaghi Azar et al., 2023, Tang et al., 2024b, Rafailov et al., 2024a, Zeng et al., 2024, Liu et al., 2024, Song et al., 2024, Zhou et al., 2023, Guo et al., 2024, Yang et al., 2024]. We refer readers to Shen et al. [2023], Wang et al. [2024c] for comprehensive reviews on LLM alignment.

Multi-Objective Optimization. MOO has been actively studied in control systems [Gambier and Badreddin, 2007] and economics [Tapia and Coello, 2007]. The main focus of the related research is the development of algorithms to profile Pareto fronts efficiently so as to understand the trade-offs between objectives. Traditional methods include the evolutionary algorithms [Zhou et al., 2011] and Bayesian optimization [Laumanns and Ocenasek, 2002]. Recently, gradient-based MOO methods have been studied in the machine learning settings [Sener and Koltun, 2018, Lin et al., 2019, Mahapatra and Rajan, 2020, Liu and Vicente, 2021, Ren et al., 2024]. Hypernetwork-based methods are also explored by a series of works [Navon et al., 2020, Lin et al., 2020, Chen and Kwok, 2022, Hoang et al., 2023].

Learning-to-Rank. LTR [Liu et al., 2009] tasks differ from traditional supervised learning in that they do not associate each sample with a simple label; instead, an optimal order of items within a group to maximize metrics, *e.g.*, Normalized Discount Cumulative Gain (NDCG) [Järvelin and Kekäläinen, 2002, Wang et al., 2013]. Typically, LTR models score documents and rank them thereby. To bridge LTR with supervised learning, various differentiable losses have been proposed as the proxy to these metrics [Burges et al., 2006, Taylor et al., 2008, Cao et al., 2007, Qin et al., 2021, Swezey et al., 2021]. In the context of Multi-Objective LTR, existing work includes label aggregation [Dai et al., 2011, Carmel et al., 2020], loss aggregation [Hu and Li, 2018, Mahapatra et al., 2023a,b, Tang et al., 2024a], and hypernetwork [Chen et al., 2023].

2 PRELIMINARIES

In this section, we briefly recapitulate the proximal and direct preference optimization frameworks for fine-tuning LLMs with preference data, and their generalization to listwise preference optimization with ranking data. We will also review the MOO problem in machine learning settings and then introduce the focus of this work, the Multi-Objective Fine-Tuning (MOFT) problem.

2.1 PROXIMAL AND DIRECT PREFERENCE OPTIMIZATION

Suppose we have a base model $p_0(\boldsymbol{y}|\boldsymbol{x})$, with $\boldsymbol{x}$ and $\boldsymbol{y}$ being the context and proposal, respectively, and $p_0(\boldsymbol{y}|\boldsymbol{x})$ the probability of generating $\boldsymbol{y}$ given $\boldsymbol{x}$. The goal of DPO is to fine-tune the model $p_0(\boldsymbol{y}|\boldsymbol{x})$ with preference data

$$\mathcal{D}_{\text{DPO}} = \{(\boldsymbol{x}^{(k)}, \boldsymbol{y}_1^{(k)} \succ \boldsymbol{y}_2^{(k)})\}_{k\in[N]}, \tag{1}$$

where $\boldsymbol{y}_1^{(k)} \succ \boldsymbol{y}_2^{(k)}$ denotes $\boldsymbol{y}_1^{(k)}$ is preferred over $\boldsymbol{y}_2^{(k)}$ in the context of $\boldsymbol{x}^{(k)}$.

Proximal Preference Optimization. In PPO [Schulman et al., 2017] or RLHF [Christiano et al., 2017], one first models the preference data by the *Bradley-Terry-Luce (BTL) model* [Bradley and Terry, 1952]:

$$\mathbb{P}(y_1 \succ y_2|\boldsymbol{x}) = \sigma\left(r(y_1|\boldsymbol{x}) - r(y_2|\boldsymbol{x})\right), \tag{2}$$

where $r(y|\boldsymbol{x})$ is the reward function and σ is the sigmoid function. PPO is carried out in two steps: (1) parametrizing $r(y|\boldsymbol{x})$ by a neural network $r_\phi(y|\boldsymbol{x})$, where parameters ϕ are trained by maximizing log-likelihood:

$$-\mathcal{L}(r_\phi; \mathcal{D}_{\mathrm{DPO}}) = \mathbb{E}\left[\log \sigma(r_\phi(y_1|\boldsymbol{x}) - r_\phi(y_2|\boldsymbol{x}))\right], \tag{3}$$

and (2) fine-tuning the base model $p_0(y|\boldsymbol{x})$ by maximizing the expected reward while maintaining the KL divergence proximity from the base model:

$$-\mathcal{L}(p_\theta; p_0, r_\phi, \beta) = \mathbb{E}\left[r_\phi(y|\boldsymbol{x})\right] - \beta D_{\mathrm{KL}}(p_\theta || p_0), \tag{4}$$

where $\beta > 0$ is called the *temperature* parameter.

Direct Preference Optimization. The observation that motivates DPO [Rafailov et al., 2024b] is that the reward function $r_\phi(\boldsymbol{x}, y)$ in (4) can be solved explicitly by letting $r_\theta(y|\boldsymbol{x}) = \beta \log \frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})}$, and therefore, the training process can be simplified to a one-shot logistic regression:

$$\begin{aligned} &-\mathcal{L}_{\mathrm{DPO}}(p_\theta; p_0, \beta, \mathcal{D}_{\mathrm{DPO}}) \\ =&\mathbb{E}\left[\log \sigma\left(\beta\left(\log \tfrac{p_\theta(y_1|\boldsymbol{x})}{p_0(y_1|\boldsymbol{x})} - \log \tfrac{p_\theta(y_2|\boldsymbol{x})}{p_0(y_2|\boldsymbol{x})}\right)\right)\right]. \end{aligned} \tag{5}$$

For completeness, we provide the proofs of the claims above in App. B.1.

2.2 LEARNING-TO-RANK AND LISTWISE PREFERENCE OPTIMIZATION

In LTR tasks, we are given a ranking dataset

$$\mathcal{D}_{\mathrm{LTR}} = \{(\boldsymbol{x}^{(k)}, \boldsymbol{y}_1^{(k)} \succ \cdots \succ \boldsymbol{y}_n^{(k)})\}_{k\in[N]},$$

where $\boldsymbol{y}_1^{(k)} \succ \cdots \succ \boldsymbol{y}_n^{(k)}$ denotes the ranking of the proposals in the context of $\boldsymbol{x}^{(k)}$. As the listwise counterpart of the BTL model, the *Plackett-Luce* (PL) model [Plackett, 1975] postulates that the probability of the ranking $\boldsymbol{\pi}$ is given by:

$$\mathbb{P}(\boldsymbol{y}_{\pi_1} \succ \cdots \succ \boldsymbol{y}_{\pi_n}|\boldsymbol{x}) = \prod_{i=1}^{n} \frac{e^{s(\boldsymbol{y}_{\pi_i}|\boldsymbol{x})}}{\sum_{k=i}^{n} e^{s(\boldsymbol{y}_{\pi_k}|\boldsymbol{x})}}, \tag{6}$$

with $s(\boldsymbol{y}|\boldsymbol{x})$ being a score function, and thus the top-one probability is given by the $\mathrm{softmax}$ function:

$$\mathbb{P}(\boldsymbol{y}_i \succ \boldsymbol{y}_{i'},\ \forall i' \neq i|\boldsymbol{x}) = \frac{e^{s(\boldsymbol{y}_i|\boldsymbol{x})}}{\sum_{i'=1}^{n} e^{s(\boldsymbol{y}_{i'}|\boldsymbol{x})}}.$$

In many scenarios, the ranking in $\mathcal{D}_{\mathrm{LTR}}$ is given by a label vector $\boldsymbol{z}$, with $z_1 \geq \cdots \geq z_n$, indicating the preference tendency of proposals. The goal is to learn the score $s_\theta(\boldsymbol{y}|\boldsymbol{x})$ parameterized by a neural network with parameters θ. One of the most popular loss functions is the *ListNet* loss [Cao et al., 2007], which aligns an appropriate normalized version $\overline{\boldsymbol{z}}$ of the labels $\boldsymbol{z}$ with the top-one probabilities:

$$-\mathcal{L}_{\mathrm{LN}}(s_\theta; \mathcal{D}_{\mathrm{LTR}}) = \mathbb{E}\left[\sum_{i=1}^{n} \overline{z}_i \log \frac{e^{s_\theta(\boldsymbol{y}_i|\boldsymbol{x})}}{\sum_{i'=1}^{n} e^{s_\theta(\boldsymbol{y}_{i'}|\boldsymbol{x})}}\right]. \tag{7}$$

Common choices include the $\mathrm{softmax}$ function for dense labels and L_1 normalization for sparse labels, corresponding to different modeling of the ranking data.

Similarly, the DPO framework can also be generalized from preference to ranking datasets. Suppose the base model is given in the form of a score function $s_0(\boldsymbol{y}|\boldsymbol{x})$, the *listwise preference optimization* (LiPO) Liu et al. [2024] proposes the following loss function to obtain a fine-tuned model $s_\theta(\boldsymbol{y}|\boldsymbol{x})$:

$$\begin{aligned} &-\mathcal{L}_{\mathrm{LiPO}}(s_\theta; s_0, \beta, \mathcal{D}_{\mathrm{LTR}}) \\ =&\mathbb{E}\left[\sum_{i=1}^{n} \overline{z}_i \log \frac{e^{\beta(s_\theta(\boldsymbol{y}_i|\boldsymbol{x}) - s_0(\boldsymbol{y}_i|\boldsymbol{x}))}}{\sum_{i'=1}^{n} e^{\beta(s_\theta(\boldsymbol{y}_{i'}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'}|\boldsymbol{x}))}}\right]. \end{aligned} \tag{8}$$

A justification for this loss is provided in App. B.1. One should notice that when adopting the L_1 normalization, the ListNet loss (7) applied to the preference dataset $\mathcal{D}_{\mathrm{DPO}}$ in the form of binary labels is equivalent to the DPO loss (5).

2.3 MULTI-OBJECTIVE OPTIMIZATION

MOO considers an optimization problem with multiple objectives $\min_{\theta\in\Theta} \boldsymbol{\mathcal{L}}(\theta) = (\mathcal{L}_1(\theta), \ldots, \mathcal{L}_m(\theta))$, where Θ is the feasible region. The goal is to profile the Pareto front, the set of trade-off solutions that cannot be improved in one objective without worsening another, or formally, the set of all θ such that for all $\theta' \in \Theta$, (1) $\mathcal{L}_i(\theta) \leq \mathcal{L}_i(\theta')$ for all $i \in [m]$, and (2) $\mathcal{L}_j(\theta) < \mathcal{L}_j(\theta')$ for some $j \in [m]$. This concept is motivated by the possible conflicts between objectives, and one may observe the trade-offs from the Pareto front and make informed decisions accordingly.

For many machine learning applications, the MOO problem can be formulated as follows. Given a dataset

$$\mathcal{D}_{\mathrm{MOO}} = \{\mathcal{D}_{\mathrm{MOO}}^{j}\}_{j\in[m]} = \{\{\boldsymbol{y}^{(k)}, z^{j,(k)}\}_{k\in[N]}\}_{j\in[m]},$$

where $\boldsymbol{y}^{(k)}$ is the feature vector and $z^{j,(k)}$ is the j-th label of the k-th data point, one learns a model $f_\theta(\boldsymbol{y})$ optimizing:

$$\min_{\theta\in\Theta} \boldsymbol{\mathcal{L}}(f_\theta; \mathcal{D}_{\mathrm{MOO}}) := (\mathcal{L}_j(f_\theta; \mathcal{D}_{\mathrm{MOO}}^{j}))_{j\in[m]}, \tag{9}$$

where $\mathcal{L}_j(f_\theta; \mathcal{D}_{\mathrm{MOO}}^{j})$ is the loss function for the model f_θ w.r.t. the j-th objective, and the feasible region Θ is over all possible model parameters.

2.4 MULTI-OBJECTIVE FINE-TUNING

We now introduce the MOFT problem as a generalization of the LLM alignment problem to the multi-objective setting with ranking datasets, where the goal is to fine-tune an existing base model $p_0(\boldsymbol{y}|\boldsymbol{x})$ (or in the form of scores $s_0(\boldsymbol{y}|\boldsymbol{x})$) w.r.t. multiple *auxiliary* objectives simultaneously while maintaining its performance on the *main* objective(s) the base model was optimized for. Similar settings are studied by multiple concurrent works [Wang et al., 2024a, Mukherjee et al., 2024, Wang et al., 2024b, Guo et al., 2024].

In this work, we formulate the MOFT problem as follows. Given a set of item groups, each of which contains a list of items and corresponding labels w.r.t. m different objectives. The dataset is of the form $\mathcal{D}_{\mathrm{MOFT}} = \{\mathcal{D}^j_{\mathrm{MOFT}}\}_{j\in[m]}$, with

$$\mathcal{D}^j_{\mathrm{MOFT}} = \left\{\boldsymbol{x}^{(k)}, (\boldsymbol{y}_i^{(k)})_{i\in[n^{(k)}]}, (z_i^{j,(k)})_{i\in[n^{(k)}]}\right\}_{k\in[N]}, \tag{10}$$

where $n^{(k)}$ is the number of items, $\boldsymbol{x}^{(k)} \in \mathbb{R}^D$ the context, $\boldsymbol{y}_i^{(k)} \in \mathbb{R}^d$ the feature vector of the i-th item, and $z_i^{j,(k)} \in \mathbb{R}^{n^{(k)}}$ the preference tendency of the i-th item w.r.t. the j-th aspect, in the k-th item group.

Relation to LLM Alignment. The preference dataset $\mathcal{D}_{\mathrm{DPO}}$ (1) in LLM alignment can be viewed as a special case of the MOFT problem, where $m = 1$, $n^{(k)} \equiv 2$, and the label $z_i^{1,(k)}$ is binary, being 1 if the i-th item is preferred over the other, and 0 otherwise.

Relation to MOO. MOFT is a generalization of the MOO problem (9) to the fine-tuning setting, where we aim to obtain all possible trade-offs of aligning the base model $p_0(\boldsymbol{y}|\boldsymbol{x})$ (or $s_0(\boldsymbol{y}|\boldsymbol{x})$) to m additional datasets $\mathcal{D}^j_{\mathrm{MOFT}}$.

Relation to LTR. When $m = 1$, the MOFT problem reduces to the task of fine-tuning a LTR model by viewing $\mathcal{D}_{\mathrm{MOFT}}$ as an additional listwise ranking dataset. This setting will be further discussed in Sec. 4.1 as we apply the COS-DPO framework to this task. We refer to Liu et al. [2024], Song et al. [2024] for more discussions on LLM alignment with listwise data.

We thus aim to design a framework as a versatile solution to the MOFT problem that can not only address the LTR fine-tuning, LLM alignment, and MOO tasks simultaneously but also synergize the state-of-the-art practices in each of these areas to achieve the best performance.

3 METHODOLOGY

In this section, we present the COS-DPO framework, a conditioned one-shot multi-objective fine-tuning framework that generalizes the DPO framework for LLM alignment to the MOFT setting and profiles the Pareto front. Below, we consider the following MOFT problem:

$$\min_{\theta\in\Theta} \boldsymbol{\mathcal{L}}_{\mathrm{LiPO}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}}), \tag{11}$$

where the vector of the loss functions $\boldsymbol{\mathcal{L}}_{\mathrm{LiPO}}$ consists of the LiPO loss functions $(\mathcal{L}_{\mathrm{LiPO}}(s_\theta; s_0, \beta_j, \mathcal{D}^j_{\mathrm{MOFT}}))_{j\in[m]}$ (8) for each auxiliary objective, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ is the vector of temperatures that control the trade-off between the main objective and each auxiliary objective.

Linear Scalarization. The most straightforward way to solve this MOO problem is to train the model s_θ with a linear combination of the preference data [Zhou et al., 2023]:

$$\mathcal{L}_{\boldsymbol{w}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}}) = \boldsymbol{w}^\top \boldsymbol{\mathcal{L}}_{\mathrm{LiPO}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}}),$$

where $\boldsymbol{w} = (w_1, \ldots, w_m)^\top \in \Delta^m$ is the weight vector that reflects the importance we assign over the objectives, and with Δ^m being the m-dimensional probability simplex.

As $\boldsymbol{w}$ iterates over Δ^m, the model s_θ will be optimized over a specific trade-off between the main objective and the auxiliary objectives and possibly land on the Pareto front. This approach is able to obtain the complete Pareto front when it is convex [Jakob and Blume, 2014].

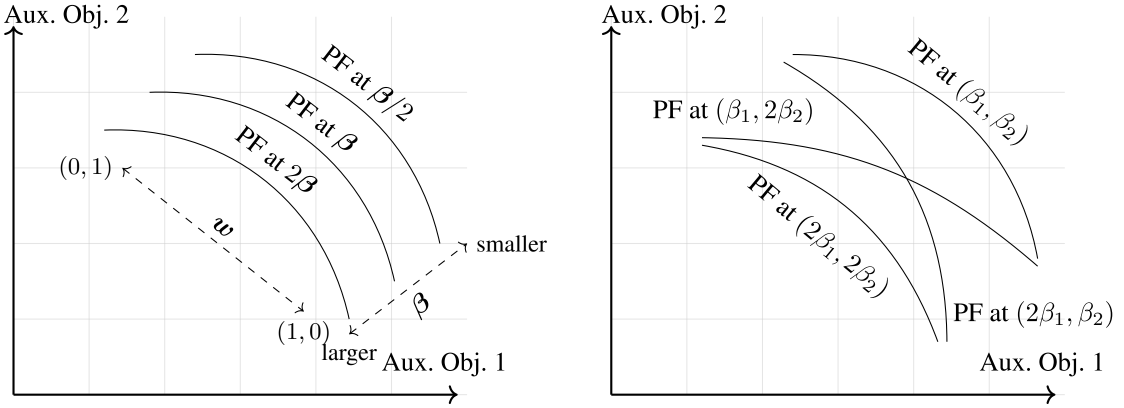
Conditioned One-Shot Networks. An efficient way to profile the Pareto front of this MOFT problem is to use *hypernetworks* [Navon et al., 2020, Hoang et al., 2023], *i.e.*, additional neural networks that generate the model parameters according to the weight vector $\boldsymbol{w}$. As an efficient and robust alternative to hypernetworks, Ruchte and Grabocka [2021] proposes *conditioned one-shot networks* that directly input the weight vector $\boldsymbol{w}$ to the model, with successful applications to multiple MOO tasks.

3.1 WEIGHT-CONDITIONED NETWORKS

COS-DPO generalizes the idea of conditioned one-shot networks to the MOFT setting. To be specific, we propose *Weight-Conditioned One-Shot* (Weight-COS) networks $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ that not only take in the data but also condition on the weight $\boldsymbol{w}$ over objectives. Intuitively, it formulates the MOFT problem into a "meta-learning" problem, and the model $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ is trained to optimize the objectives over a distribution of weight vectors $\boldsymbol{w}$.

Since $\boldsymbol{w}$ is supported on Δ^m, we sample $\boldsymbol{w}$ from a Dirichlet distribution $\mathrm{Dir}(\boldsymbol{\alpha})$ during each epoch of the training process, where $\boldsymbol{\alpha}$ is the concentration parameter. The Weight-COS-DPO method is equivalent to optimize the Weight-COS network $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ w.r.t. the following loss function:

$$\begin{aligned}&\mathcal{L}_{\text{W-COS}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}}, \boldsymbol{\alpha})\\ =&\mathbb{E}_{\boldsymbol{w}\sim\mathrm{Dir}(\boldsymbol{\alpha})}[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}})].\end{aligned} \tag{12}$$



(a) Weight-COS-DPO

(b) Temperature-COS-DPO

Figure 1: Conceptual Illustration of Post-Training Controls in the COS-DPO Framework with 2 auxiliary objectives.

Ideally, *i.e.*, when the model has sufficient capacity and the Pareto front is smooth and convex, the optimized model $s_{\theta,\boldsymbol{\beta}}(\cdot, \boldsymbol{w}|\boldsymbol{x})$ would be Pareto optimal w.r.t. the auxiliary objectives for each $\boldsymbol{w} \in \Delta^m$ and thus form the Pareto front.

The Weight-COS-DPO method is summarized in Alg. 1.

Algorithm 1: Weight-COS-DPO

```
Data: Base model $s_0(\boldsymbol{y}|\boldsymbol{x})$, dataset $\mathcal{D}_{\mathrm{MOFT}}$
      concentration parameter $\boldsymbol{\alpha}$, temperature $\boldsymbol{\beta}$.
Result: Fine-Tuned model $s_{\theta,\boldsymbol{\beta}}(\cdot,\cdot|\boldsymbol{x})$.
for $e = 1$ to $N_{\mathrm{steps}}$ do
    Sample $\boldsymbol{w}' \sim \mathrm{Dir}(\boldsymbol{\alpha})$;
    $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}'|x); s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}})$;
end
```

3.2 LINEAR TRANSFORMATION PROPERTY

Due to the linearity of the Weight-COS-DPO method, we underscore the following linear transformation property:

Proposition 3.1 (Linear Transformation Property)**.** *For any $\boldsymbol{\beta} \in \mathbb{R}_+^m$ and $\boldsymbol{w} \in \Delta^m$, we denote the model obtained by optimizing the Weight-COS-DPO loss (12) with temperature $\boldsymbol{\beta}$ as $s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})$.*

Then $s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})$ should satisfy the following linear transformation property *that for any $c > 0$, we have*

$$s_{\theta,c\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) = \left(1 - \tfrac{1}{c}\right) s_0(\boldsymbol{y}|\boldsymbol{x}) + \tfrac{1}{c} s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) \quad (13)$$

is also an optimal solution to the Weight-COS-DPO loss (12) with temperature $c\boldsymbol{\beta}$.

The proof of this proposition is provided in App. B.2 and will be empirically validated with experiments as shown in Fig. 12. Powered by Prop. 3.1, Weight-COS networks offer post-training controls over the trade-offs between the main and auxiliary objectives. To be specific, once we have trained a model $s_\theta(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})$ with a specific temperature $\boldsymbol{\beta}$, we may also obtain the Pareto front under a different temperature $c\boldsymbol{\beta}$ by simply scaling the output as:

$$s_\theta(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) \leftarrow \left(1 - \tfrac{1}{c}\right) s_0(\boldsymbol{y}|\boldsymbol{x}) + \tfrac{1}{c} s_\theta(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}). \quad (14)$$

Consequently, as illustrated in Fig. 1a, after only one training process with a specific choice of temperature $\boldsymbol{\beta}$, Weight-COS-POS allows post-training controls over two kinds of trade-offs: (1) those between the auxiliary objectives by adjusting the weight vector $\boldsymbol{w}$, and (2) those between the fidelity to the base model and its performance on the auxiliary objectives by scaling temperature $\boldsymbol{\beta}$ with (14).

3.3 TEMPERATURE-CONDITIONED NETWORKS

Although Prop. 3.1 provides a flexible way to control the trade-offs between the main and auxiliary objectives for the Weight-COS-DPO method, it covers only one specific degree of freedom in the space of the space of the second type of trade-offs. Generally speaking, the model should exhibit different Pareto fronts for different temperature parameters $\boldsymbol{\beta} \in \mathbb{R}_+^m$, and thus one may consider using different temperatures across objectives and also a disproportionate post-training scaling of $\boldsymbol{\beta}$ to achieve more flexible control over the Pareto front (*cf.* Fig. 1b vs. Fig. 1a).

To this end, as a generalized version of the Weight-COS networks, we propose the *Temperature-Conditioned One-Shot* (Temperature-COS) networks by further incorporating the temperature parameter $\boldsymbol{\beta}$ into the model input in a similar manner as the weight vector $\boldsymbol{w}$, denoted by $s_\theta(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x})$. Prop. 3.1 implies that the temperature $\boldsymbol{\beta} \in \mathbb{R}_+^m$ is actually of $m - 1$ degrees of freedom, and thus we propose to use the following reparametrization by projecting $\boldsymbol{\beta}$ to its L^1-

normalization $\overline{\boldsymbol{\beta}} = \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_1} \in \Delta^m$, *i.e.*,

$$\begin{aligned} &s_\theta(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x}) \\ =&\big(1 - \tfrac{1}{\|\boldsymbol{\beta}\|_1}\big)s_0(\boldsymbol{y}|\boldsymbol{x}) + \tfrac{1}{\|\boldsymbol{\beta}\|_1}s_\theta(\boldsymbol{y}, \boldsymbol{w}, \overline{\boldsymbol{\beta}}|\boldsymbol{x}). \end{aligned} \tag{15}$$

The training is thus conducted similarly to the Weight-COS-DPO method by randomly sampling $\boldsymbol{\beta} \in \mathbb{R}_+^m$ over a certain distribution $\mathcal{D}_{\boldsymbol{\beta}}$ valued in $\mathbb{R}_+^m$ besides $\boldsymbol{w}$ at each epoch. The loss of Temperature-COS-DPO can be written as

$$\begin{aligned} &\mathcal{L}_{\text{T-COS}}(s_\theta; s_0, \mathcal{D}_{\text{MOFT}}, \boldsymbol{\alpha}) \\ =&\mathbb{E}_{\boldsymbol{\beta}\sim\mathcal{D}_{\boldsymbol{\beta}}, \boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x}); s_0, \mathcal{D}_{\text{MOFT}})]. \end{aligned} \tag{16}$$

The Temperature-COS-DPO method is provided in Alg. 2.

Algorithm 2: Temperature-COS-DPO

```
Data: Base model $s_0(\boldsymbol{y}|\boldsymbol{x})$, dataset $\mathcal{D}_{\text{MOFT}}$,
      concentration parameter $\boldsymbol{\alpha}$, temperature
      distribution $\mathcal{D}_{\boldsymbol{\beta}}$.
Result: Fine-Tuned Model $s_\theta(\cdot, \cdot, \cdot|\boldsymbol{x})$.
for $e = 1$ to $N_{\text{steps}}$ do
    Sample $\boldsymbol{w}' \sim \text{Dir}(\boldsymbol{\alpha})$, $\boldsymbol{\beta}' \sim \mathcal{D}_{\boldsymbol{\beta}}$;
    $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}', \boldsymbol{\beta}'|\boldsymbol{x}); s_0, \mathcal{D}_{\text{MOFT}})$;
end
```

We remark that both Weight-COS-DPO and Temperature-COS-DPO can be implemented with penalization terms to foster the exploration of the Pareto front without affecting the validity of the linear transformation property (*cf.* Prop. 3.1). We refer to App. A.2 for more details.

4 EXPERIMENTS

In this section, we provide the detailed experiment design and results of the COS-DPO framework for different applications, including the LTR fine-tuning and LLM alignment task. We compare the Weight-COS-DPO (W-COS-DPO in below) method with the following existing baselines, including the DPO Linear Scalarization (DPO-LS) method, the DPO Soup method [Rame et al., 2024], and the MO-DPO method [Zhou et al., 2023]. For details and further discussion of these baselines, we refer to App. A.1. For each baseline, we use the same number of weight vectors $\boldsymbol{w}$ for a fair comparison. The *Hypervolume (HV)* indicator [Zitzler and Künzli, 2004] is adopted for evaluating the performance of MOFT methods (*cf.* App. A.3). We also carry out preliminary experiments on the proposed Temperature-COS-DPO (T-COS-DPO in below) method on the LTR fine-tuning task to demonstrate its feasibility.

4.1 LEARNING-TO-RANK FINE-TUNING

We first test the COS-DPO framework on the task of fine-tuning LTR models. We adopt the Microsoft Learning-to-Rank Web Search (MSLR-WEB10K) dataset [Qin and Liu, 2013] for the LTR task. The MSLR-WEB10K dataset consists of 10,000 groups ($N = 10^4$), with 5 auxiliary objectives ($m = 5$). We refer to App. A.4 for more details.

In MSLR-WEB10K dataset, the information $\boldsymbol{x}$ has been incorporated into the feature vectors $\boldsymbol{y}$ by upstream data processing. We thus use a 2-layer transformer architecture of hidden dimension 128 for the base model $s_0(\boldsymbol{y})$, and the model $s_\theta(\cdot, \boldsymbol{w})$ is designed as a 2-layer transformer architecture of hidden dimension 64 with $\boldsymbol{w}$ concatenated to the input of the first layer.

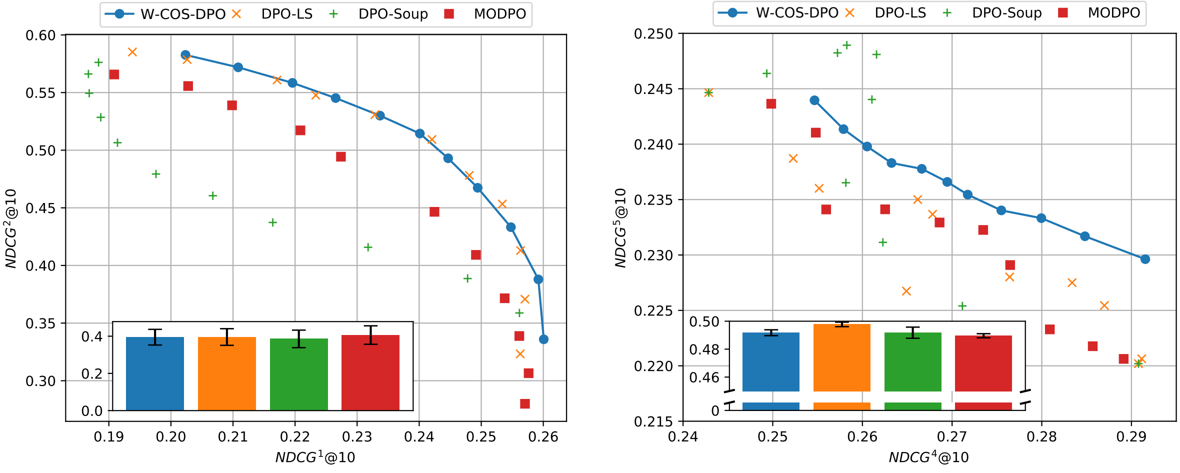
We first apply W-COS-DPO to the case $m = 2$ for better visualization. Fig. 2a presents the Pareto front of two sparse labels ($\overline{\boldsymbol{z}} = \boldsymbol{z}/\|\boldsymbol{z}\|_1$ in (8)) with a convex Pareto front, while Fig. 2b presents that of two dense labels ($\overline{\boldsymbol{z}} = \text{softmax}(\boldsymbol{z})$ in (8)) with an ill-posed Pareto front. W-COS-DPO obtains comprehensive Pareto fronts that dominate those of the baselines in both pairs. Notably, our method obtains a smooth Pareto front in Fig. 2b while other baselines fail. With a common temperature parameter $\boldsymbol{\beta}$ used across all methods, the inset plots demonstrate that the superior performance of our method is not at the cost of the main objective, as the NDCG@10 of the main objective is comparable or even slightly better to baselines.

Metric	DPO-LS	DPO Soup	MO-DPO	W-COS-DPO
Aux. HV	1.648e-3	1.468e-3	1.263e-3	**2.039e-3**
Avg. M. Scr.	0.355	0.382	0.360	**0.432**
(w/±Std)	(± 0.029)	(± 0.032)	(± 0.024)	(± 0.028)
Time (s)	14649.15	6061.69	27059.70	**4043.47**
# Params.	551,232	250,615	801,792	**50,432**

Table 1: HV metric and training time of COS-DPO and the baselines on the MSLR-WEB10K dataset with 5 auxiliary objectives. The reference point is set to $(0, 0)$. 11 points are produced for computing HV. Avg. M. Scr. (w/±Std) refers to the average NDCG@10 (with standard deviation) of the main objective across the 11 points.

We also test W-COS-DPO on a more complicated case where we have 5 auxiliary objectives ($m = 5$), as shown in Tab. 1. Our results demonstrate W-COS-DPO achieves a higher hypervolume metric with significantly less training time and number of parameters compared to the baselines and comparably good preservation of the performance on the main objective (see also App. A.5). While the computational cost of DPO-LS and MO-DPO, grows exponentially with the number of objectives, our method maintains a linear growth with almost intact performance, indicating the efficiency of our W-COS-DPO method in handling high-dimensional MOFT problems in the LTR task.

We present an example of the post-training control over the trade-offs between the main and auxiliary objectives by both W-COS-DPO and T-COS-DPO in Fig. 3. Detailed



(a) Objective I vs Objective II.

(b) Objective IV vs Objective V.

Figure 2: Comparison of Pareto fronts obtained by W-COS-DPO and baselines on the MSLR-WEB10K dataset with 2 auxiliary objectives. Two axes denote the NDCG@10 of the two auxiliary objectives (the higher, the better). The inset plot shows the average NDCG@10 of the main objective, with the error bar denoting the standard deviation across the 11 sampled points.

experimental settings are deferred to Apps. A.7 and A.8. While the model obtained by W-COS-DPO in Fig. 3a is only able to apply linear translation to the Pareto front, the model obtained by the T-COS-DPO method in Fig. 3b demonstrates its capability of scaling the Pareto front in an objective-specific manner. This is in exact accordance with their expected performance as illustrated in Fig. 1 and validates the feasibility of the proposed T-COS-DPO method. Our theoretical findings (Prop. 3.1) are also confirmed by Fig. 13, where similar Pareto fronts are obtained by training with different temperatures and scaling the output with (14).

We study the hyperparameter robustness of W-COS-DPO on the LTR fine-tuning task in App. A.6. Specifically, we evaluate its sensitivity to the concentration parameter $\boldsymbol{\alpha}$ (*cf.* App. A.6.1) and the model depth (capacity) (*cf.* App. A.6.2). Furthermore, we introduce and compare the performance of two different parametrizations of $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ in App. A.6.3, namely (a) *Training-from-Scratch* and (b) *Augmentation Network*, which exhibit different trade-offs between the performance and the computational cost and thus may serve different purposes in practice.

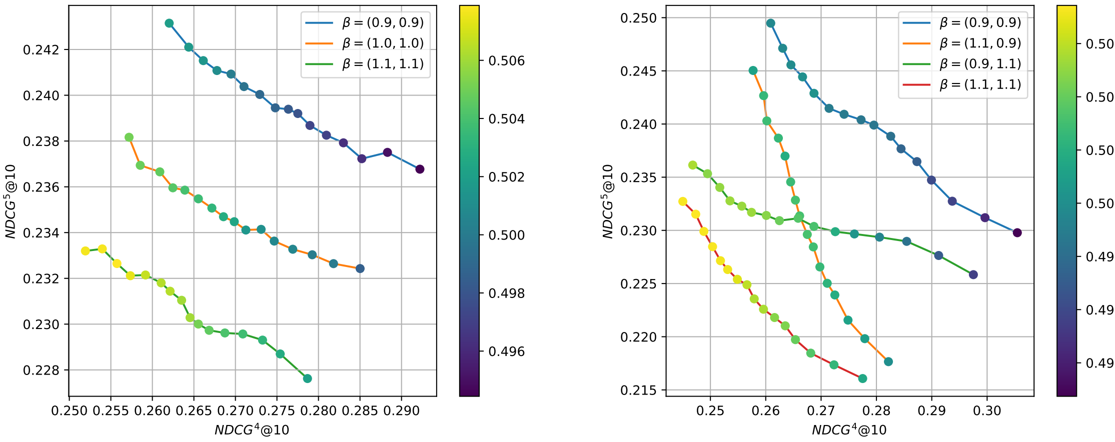
4.2 LLM ALIGNMENT TASK

We then apply the COS-DPO framework to the LLM alignment task. The PKU-SafeRLHF dataset [Ji et al., 2024a] is adopted for experiments, which consists of 83.4k entries, with 2 auxiliary objectives ($m = 2$). We refer to App. A.4 for more details on the dataset.

In contrast to the LTR task, where we directly concatenate the weight $\boldsymbol{w}$ to the input of the W-COS network $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$. In LLM alignment, this strategy is generally infeasible, since the input of transformers is tokenized prompts, and the weight vector $\boldsymbol{w}$ is a real vector that is not considered in tokenization and thus a direct concatenation may cause the model to fail to generate meaningful responses. To address this issue and incorporate the information of the weight vector $\boldsymbol{w}$ into the LLM with the least modification to the model and the training process, we propose a novel design, called *Hyper Prompt Tuning (HPT)*.

The mechanism of HPT is shown in Fig. 4. Inspired by Prompt Tuning [Lester et al., 2021], HPT augments the input embedding obtained post token embedding and positional encoding with a trainable prefix embedding block that is controlled by the weight vector $\boldsymbol{w}$. Specifically, HPT follows the following steps: (1) HPT takes in a weight vector $\boldsymbol{w} \in \Delta^m$ that indicates the importance across auxiliary objectives and, through simple trainable MLPs, produces two matrices, the matrix product of which forms the mask, (2) the mask is multiplied entrywise with a trainable prefix embedding block with k virtual tokens, and (3) the prefix embedding block is then concatenated to the input embedding as a prefix and fed into the transformer blocks of the LLM. In contrast to Multi-Task Prompt Tuning [Wang et al., 2023], which permits only a finite number of tasks, one can pass a continuum of weight by HPT into the LLM, offering both flexibility and versatility.

We thus perform fine-tuning to the GPT-2 model [Radford et al., 2019] and the Alpaca-7B-Reproduced model [Dai et al., 2023], following the practice by Zhou et al. [2023]



(a) W-COS-DPO.

(b) T-COS-DPO.

Figure 3: Examples of post-training control over temperature β of both W-COS-DPO and T-COS-DPO.

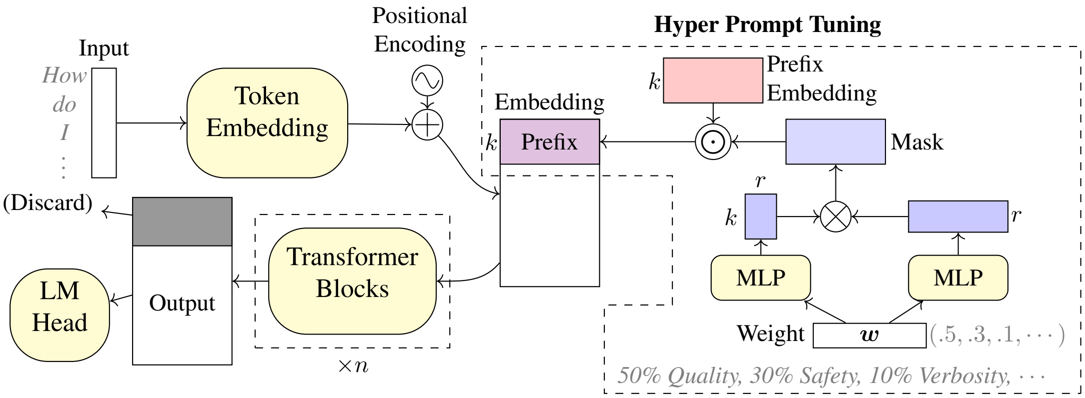
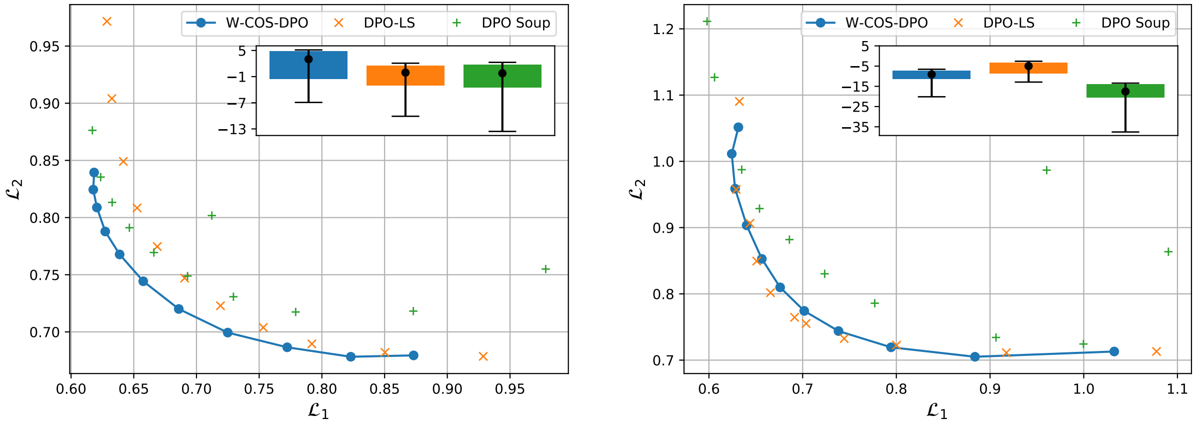


Figure 4: Illustration of the implementation of the W-COS-DPO framework for the LLM alignment task. The proposed **Hyper Prompt Tuning** method, highlighted within the dashed box, transforms the weight vector $\boldsymbol{w}$ into a mask and passes it to the LLM via prompt tuning. k denotes the number of virtual tokens, and r is the rank of the weight mask.



(a) GPT-2

(b) Alpaca-7B-Reproduced

Figure 5: Comparison of Pareto fronts obtained by W-COS-DPO and baselines on the PKU-SafeRLHF dataset. Two axes denote the expected cross-entropy error of two auxiliary objectives (the lower, the better). The inset plot shows the interquartile range (IQR) of the log-likelihood deviation of the response from the reference model across the test dataset.

Method	GPT-2		Alpaca-7B-Rep.	
	HV	**Time (s)**	**HV**	**Time (s)**
DPO-LS	0.1767	15148.5	0.1687	94156.1
DPO Soup	0.1840	2755.5	0.1427	17138.7
W-COS-DPO	**0.1942**	**1396.8**	**0.1689**	**8520.2**

Table 2: HV metric and training time of W-COS-DPO and the baselines on the PKU-SafeRLHF dataset. The reference point is set to $(1.1, 1.1)$, and 11 points are produced for computing HV.

via Parameter-Efficient Fine-Tining (PEFT) with $\alpha = 8$ and $r = 4$ in the low-rank adaptions (LoRA) to the modules within the model. For W-COS-DPO, we adopt the Hyper Prompt Tuning technique with $k = 8$ and $r = 4$. To ensure a fair comparison, baseline methods will also be augmented with the prompt tuning of $k = 8$ on top of LoRA. Our method is built upon the TRL package [von Werra et al., 2020], and the implementation of the HPT is compatible with the PEFT package [Mangrulkar et al., 2022], allowing easy integration with existing LLMs. All the experiments are conducted on 8× NVIDIA A100 GPUs.

In this task, we compare the results of W-COS-DPO with those of DPO-LS and DPO Soup and we refer readers to discussions in App. A.1 for the comparison with MO-DPO. For all experiments, we have chosen a common temperature $\beta = 0.1$ to balance the trade-offs between the main and auxiliary objectives. W-COS-DPO achieves smooth and comprehensive Pareto fronts (*cf.* Fig. 5) with higher hypervolume metrics and less training time (*cf.* Table 2) for both LLM architectures compared to the baselines, demonstrating the effectiveness of our method in the large-scale LLM alignment tasks. Notably, as W-COS-DPO tackles a intrinsically more challenging "meta-learning" problem and thus demands more expressive power, our method is less prone to overfitting and more robust to the choice of the hyperparameters compared to the baselines. Several studies on the hyperparameters are provided in App. A.6.

5 DISCUSSION

In this work, we propose the COS-DPO framework for multi-objective fine-tuning, which is inspired by the DPO framework to profile the Pareto front for a wide range of multi-objective fine-tuning problems. Our method enjoys an efficient one-shot training process by conditioning the model on the importance weights $\boldsymbol{w}$ across auxiliary objectives (Weight-COS-DPO), and further also on the temperature $\boldsymbol{\beta}$ (Temperature-COS-DPO) to achieve the desired trade-offs between the main and auxiliary objectives.

We demonstrated the effectiveness and efficiency of Weight-COS-DPO in handling high-dimensional MOFT problems in both the LTR fine-tuning and large-scale LLM alignment tasks, displayed the post-training control by the linear transformation property, and empirically validated the feasibility of Temperature-COS-DPO. Our newly proposed Hyper Prompt Tuning technique also provides a novel way to incorporate continuous information into the LLM.

We expect to incorporate other possible MOO and LTR techniques in the Weight-COS-DPO method and further explore the potential of the Temperature-COS-DPO method in various, more complicated multi-objective fine-tuning problems in future works.

Acknowledgements

This work was conducted during Yinuo Ren's internship at Amazon. We thank Subhajit Sanyal, Yeshwant Dattatreya, and the whole Amazon Search team for their support and constructive feedback.

References

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

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

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

Christopher Burges, Robert Ragno, and Quoc Le. Learning to rank with nonsmooth cost functions. *Advances in neural information processing systems*, 19, 2006.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

David Carmel, Elad Haramaty, Arnon Lazerson, and Liane Lewin-Eytan. Multi-objective ranking optimization for product search using stochastic label aggregation. In *Proceedings of The Web Conference 2020*, pages 373–383, 2020.

Angelica Chen, Sadhika Malladi, Lily H Zhang, Xinyi Chen, Qiuyi Zhang, Rajesh Ranganath, and Kyunghyun Cho. Preference learning algorithms do not learn preference rankings. *arXiv preprint arXiv:2405.19534*, 2024.

Sirui Chen, Yuan Wang, Zijing Wen, Zhiyu Li, Changshuo Zhang, Xiao Zhang, Quan Lin, Cheng Zhu, and Jun Xu. Controllable multi-objective re-ranking with policy hypernetworks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3855–3864, 2023.

Weiyu Chen and James Kwok. Multi-objective deep learning with adaptive reference vectors. *Advances in Neural Information Processing Systems*, 35:32723–32735, 2022.

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

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*, 2023.

Na Dai, Milad Shokouhi, and Brian D Davison. Multi-objective optimization in learning to rank. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 1241–1242, 2011.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Adrian Gambier and Essameddin Badreddin. Multi-objective optimal control: An overview. In *2007 IEEE international conference on control applications*, pages 170–175. IEEE, 2007.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv e-prints*, pages arXiv–2310, 2023.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Jiexin Wang, Huimin Chen, Bowen Sun, Ruobing Xie, Jie Zhou, Yankai Lin, et al. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*, 2024.

Long P Hoang, Dung D Le, Tran Anh Tuan, and Tran Ngoc Thang. Improving pareto front learning via multi-sample hypernetworks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7875–7883, 2023.

Jun Hu and Ping Li. Collaborative multi-objective ranking. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1363–1372, 2018.

Wilfried Jakob and Christian Blume. Pareto optimization or cascaded weighted sum: A comparison of concepts. *Algorithms*, 7(1):166–185, 2014.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. Pku-saferlhf: A safety alignment preference dataset for llama family models. *arXiv preprint arXiv:2406.15513*, 2024a.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36, 2024b.

Marco Laumanns and Jiri Ocenasek. Bayesian optimization algorithms for multi-objective optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 298–307. Springer, 2002.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qing-Fu Zhang, and Sam Kwong. Pareto multi-task learning. *Advances in neural information processing systems*, 32, 2019.

Xi Lin, Zhiyuan Yang, Qingfu Zhang, and Sam Kwong. Controllable pareto multi-task learning. *arXiv preprint arXiv:2010.06313*, 2020.

Suyun Liu and Luis Nunes Vicente. The stochastic multi-gradient algorithm for multi-objective optimization and its application to supervised machine learning. *Annals of Operations Research*, pages 1–30, 2021.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, et al. Lipo: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*, 2024.

Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3): 225–331, 2009.

Debabrata Mahapatra and Vaibhav Rajan. Multi-task learning with user preferences: Gradient descent with controlled ascent in pareto optimization. In *International Conference on Machine Learning*, pages 6597–6607. PMLR, 2020.

Debabrata Mahapatra, Chaosheng Dong, Yetian Chen, and Michinari Momma. Multi-label learning to rank through multi-objective optimization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4605–4616, 2023a.

Debabrata Mahapatra, Chaosheng Dong, and Michinari Momma. Querywise fair learning to rank through multi-objective optimization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1653–1664, 2023b.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

Subhojyoti Mukherjee, Anusha Lalitha, Sailik Sengupta, Aniket Deshmukh, and Branislav Kveton. Multi-objective alignment of large language models through hypervolume maximization. *arXiv preprint arXiv:2412.05469*, 2024.

Aviv Navon, Aviv Shamsian, Gal Chechik, and Ethan Fetaya. Learning the pareto front with hypernetworks. *arXiv preprint arXiv:2010.04104*, 2020.

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

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Tao Qin and Tie-Yan Liu. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Mike Bendersky, and Marc Najork. Are neural rankers still outperformed by gradient boosted decision trees? In *International Conference on Learning Representations (ICLR)*, 2021.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9, 2019.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024a.

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*, 36, 2024b.

Alexandre Rame, Guillaume Couairon, Corentin Dancette, Jean-Baptiste Gaya, Mustafa Shukor, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36, 2024.

Yinuo Ren, Tesi Xiao, Tanmay Gangwani, Anshuka Rangi, Holakou Rahmanian, Lexing Ying, and Subhajit Sanyal. Multi-objective optimization via wasserstein-fisher-rao gradient flow. In *International Conference on Artificial Intelligence and Statistics*, pages 3862–3870. PMLR, 2024.

Michael Ruchte and Josif Grabocka. Scalable pareto front approximation for deep multi-objective learning. In *2021 IEEE international conference on data mining (ICDM)*, pages 1306–1311. IEEE, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31, 2018.

Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*, 2023.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998, 2024.

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

Robin Swezey, Aditya Grover, Bruno Charron, and Stefano Ermon. Pirank: Scalable learning to rank via differentiable sorting. *Advances in Neural Information Processing Systems*, 34:21644–21654, 2021.

Jie Tang, Huiji Gao, Liwei He, and Sanjeev Katariya. Multi-objective learning to rank by model distillation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5783–5792, 2024a.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024b.

Ma Guadalupe Castillo Tapia and Carlos A Coello Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *2007 IEEE congress on evolutionary computation*, pages 532–539. IEEE, 2007.

Michael Taylor, John Guiver, Stephen Robertson, and Tom Minka. Softrank: optimizing non-smooth rank metrics. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, pages 77–86, 2008.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`, 2020.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024a.

Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Avinava Dubey, et al. Conditional language policy: A general framework for steerable multi-objective finetuning. *arXiv preprint arXiv:2407.15762*, 2024b.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR, 2013.

Zhen Wang, Rameswar Panda, Leonid Karlinsky, Rogerio Feris, Huan Sun, and Yoon Kim. Multitask prompt tuning enables parameter-efficient transfer learning. *arXiv preprint arXiv:2303.02861*, 2023.

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*, 2024c.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-dpo: Direct preference optimization with dynamic β. *arXiv preprint arXiv:2407.08639*, 2024a.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36, 2024b.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024.

Rui Yang, Xiaoman Pan, Feng Luo, Shuang Qiu, Han Zhong, Dong Yu, and Jianshu Chen. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. *arXiv preprint arXiv:2402.10207*, 2024.

Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.

Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and evolutionary computation*, 1 (1):32–49, 2011.

Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization. *arXiv preprint ArXiv:2310.03708*, 2023.

Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *International conference on parallel problem solving from nature*, pages 832–842. Springer, 2004.

COS-DPO: Conditioned One-Shot Multi-Objective Fine-Tuning Framework (Supplementary Material)

Yinuo Ren[1] **Tesi Xiao**[2] **Michael Shavlovsky**[2] **Lexing Ying**[3,1] **Holakou Rahmanian**[2]

[1]Institute for Computational and Mathematical Engineering (ICME), Stanford University
[2]Amazon
[3]Department of Mathematics, Stanford University

A ADDITIONAL EXPERIMENT DETAILS

In this section, we present additional details and results of the experiments conducted in the main text, including further descriptions of the baseline implementations, the penalization terms in the pratical implementation of Weight-COS-DPO and Temperature-COS-DPO. We will also provide a brief discussion on the hypervolume metric as the evaluation metric for the multi-objective fine-tuning (MOFT) methods, and detailed descriptions of the datasets used in the experiments. Additional experimental results of post-training control over trade-offs and the Temperature-COS-DPO method are also provided.

A.1 BASELINE IMPLEMENTATIONS

In the following, we will introduce and discuss the baseline methods used in the experiments in detail.

DPO Linear Scalarization (DPO-LS). Given the base model s_0, for each weight vector $\boldsymbol{w} \in \mathbb{R}^m$, the DPO-LS method trains the new model s_θ with the loss function $\mathcal{L}_{\boldsymbol{w}}$ and obtain $s_{\theta,\boldsymbol{w}}$ defined as

$$\begin{aligned} s_{\theta,\boldsymbol{w}} &= \arg\min_{s_\theta} \mathcal{L}_{\boldsymbol{w}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}}) \\ &= \arg\min_{s_\theta} \boldsymbol{w}^\top \boldsymbol{\mathcal{L}}_{\mathrm{LiPO}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\mathrm{MOFT}}). \end{aligned}$$

This model is a naïve generalization from the linear scalarization method in the MOO literature to the MOFT problem, and the main drawback is that it needs as many training jobs and models as the number of sampled weight vectors, which is computationally expensive.

DPO Soup. The DPO Soup [Rame et al., 2024] model first trains m models $s_{\theta,\boldsymbol{e}_i}$ for each unit vector $\boldsymbol{e}_i$ in the m-dimensional space, *i.e.*, m DPO models w.r.t. the m auxiliary objectives, respectively, and then linearly combines the m models to obtain the final model with the weight vector $\boldsymbol{w}$ in the parameter space.

The DPO Soup method offers a more efficient way to combine the models trained with different auxiliary objectives, but it still requires m training jobs and models for each auxiliary objective, and the performance of this model is largely dependent on the landscape of the parameter space of the neural network architecture.

As depicted in Fig. 2, the Pareto front obtained by the DPO Soup method may present unexpected curves, and Fig. 5 shows that the DPO Soup method may even exhibit mode collapse for certain combinations.

MO-DPO. The MO-DPO method also starts with the training of m models $s_{\theta,\boldsymbol{e}_i}$ for each unit vector $\boldsymbol{e}_i$ in the m-dimensional space, and then instead of linearly combining the parameters, MO-DPO conducts a new training job for each weight vector $\boldsymbol{w} \in \mathbb{R}^m$ with the following loss function:

$$-\mathcal{L}_{\text{MO-DPO}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}}) = \mathbb{E}\left[\sum_{i=1}^{n} \bar{z}_i^j \log \frac{\exp\left(\beta_j r_{\theta,\boldsymbol{w}}^{\text{MO-DPO}}\right)}{\sum_{i'=1}^{n} \exp\left(\beta_j r_{\theta,\boldsymbol{w}}^{\text{MO-DPO}}\right)}\right],$$

where, for an arbitrary $i \in [m]$, $r_{\theta,\boldsymbol{w}}^{\text{MO-DPO}}$ is defined as

$$r_{\theta,\boldsymbol{w}}^{\text{MO-DPO}} := \frac{1}{w_i}\left(s_\theta(\boldsymbol{y}|\boldsymbol{x}) - s_0(\boldsymbol{y}|\boldsymbol{x}) - \sum_{i' \neq i} w_{i'}\left(s_{\theta,\boldsymbol{e}_i'}(\boldsymbol{y}|\boldsymbol{x}) - s_0(\boldsymbol{y}|\boldsymbol{x})\right)\right). \tag{17}$$

As MO-DPO requires m training jobs and one additional training job for each weight vector, it may require more training time and computational resources compared to the DPO-LS and DPO Soup methods.

For the LLM alignment task, we observe MO-DPO suffers from unstable training caused by the $1/w_i$ vector in the expression (17) especially when w_i is close to zero and exhibits less competitive performance. The results are shown in Fig. 6. We suspect that the conflict between the prompt tuning and the MO-DPO method may lead to the suboptimal performance of MO-DPO in the LLM alignment task and thus do not present the results in the main text (*cf.* Fig. 5).

The COS-DPO framework is designed to address the limitations of the existing methods and provide a more efficient and effective way to profile the Pareto front of the MOFT problems.

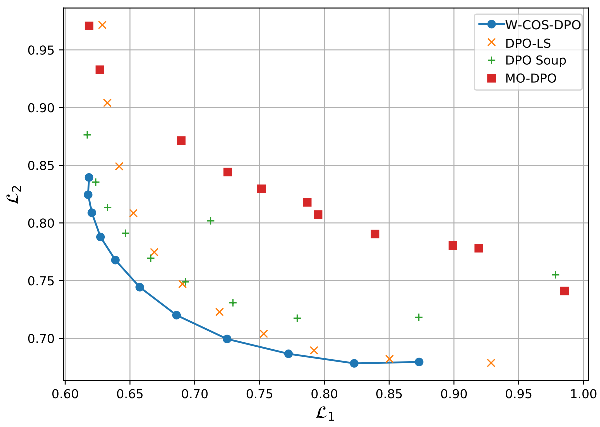


Figure 6: Comparison of Pareto fronts obtained by Weight-COS-DPO and the baselines on the PKU-SafeRLHF dataset with the GPT-2 model, including the MO-DPO method.

A.2 PENALIZATION

In practice, in order to foster the exploration of the Pareto front, one may also incorporate artificial penalization terms to the loss function, such as the cosine similarity between the loss vector $\boldsymbol{\mathcal{L}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})$ of the model and the weight vector [Ruchte and Grabocka, 2021]:

$$\begin{aligned} & -\mathcal{G}_{\boldsymbol{w}}(s_\theta; s_0, \boldsymbol{\beta}) \\ =& \cos\langle \boldsymbol{w}, -\boldsymbol{\mathcal{L}}_{\text{LiPO}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})\rangle \\ =& -\frac{\boldsymbol{w}^\top \boldsymbol{\mathcal{L}}_{\text{LiPO}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})}{\|\boldsymbol{w}\| \|\boldsymbol{\mathcal{L}}_{\text{LiPO}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})\|}, \end{aligned} \tag{18}$$

where $\langle \cdot, \cdot \rangle$ denotes the angle between two vectors.

This penalization term intuitively confines the loss vector $\boldsymbol{\mathcal{L}}_{\text{LiPO}}$ to converging along the direction of the weight vector $\boldsymbol{w}$, empowering possible profiling of concave Pareto fronts [Lin et al., 2019].

We present the practical implementation of the penalization terms in the Weight-COS-DPO and Temperature-COS-DPO methods in the following.

- For Weight-COS-DPO, the penalized loss function, modified from (12), is thus defined as

$$\begin{aligned} & \mathcal{L}_{\text{I-COS}}(s_\theta; s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}}, \boldsymbol{\alpha}, \lambda) \\ :=& \mathbb{E}_{\boldsymbol{w} \sim \text{Dir}(\boldsymbol{\alpha})}[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}}) + \lambda \mathcal{G}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta})], \end{aligned} \tag{19}$$

  and the corresponding algorithm is presented in Alg. 3.

- For Temperature-COS-DPO, the penalized loss function, modified from (16), is defined as

$$
\begin{aligned}
&\mathcal{L}_{\text{T-COS}}(s_\theta; s_0, \mathcal{D}_{\text{MOFT}}, \boldsymbol{\alpha}, \lambda) \\
:=&\mathbb{E}_{\boldsymbol{\beta}\sim\mathcal{D}(\boldsymbol{\beta}), \boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x}); s_0, \mathcal{D}_{\text{MOFT}}) + \lambda \mathcal{G}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x}); s_0)],
\end{aligned}
$$

  and the corresponding algorithm is presented in Alg. 4.

Algorithm 3: Weight-COS-DPO with Penalization

```
Data: Base model $s_0(\boldsymbol{y}|\boldsymbol{x})$, dataset $\mathcal{D}_{\text{MOFT}}$ concentration parameter $\boldsymbol{\alpha}$, temperature $\boldsymbol{\beta}$, penalization coefficient
      $\lambda$(Training); scale $c$, weight vector $\boldsymbol{w}$ (Post-Training Control).
Result: Fine-Tuned model $s_{\theta,\boldsymbol{\beta}}(\cdot, \cdot|\boldsymbol{x})$.
  // Training
  for $e = 1$ to $N_{\text{steps}}$ do
      Sample $\boldsymbol{w}' \sim \text{Dir}(\boldsymbol{\alpha})$;
      $\theta \leftarrow \theta - \eta\nabla_\theta[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}'|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}}) + \lambda\mathcal{G}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}'|\boldsymbol{x}); s_0, \boldsymbol{\beta})]$;
  end
  // Post-Training Control
  $s_{\theta,c\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) \leftarrow (1 - 1/c)\, s_{\text{base}}(\boldsymbol{y}|\boldsymbol{x}) + s_\theta(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})/c$.
```

Algorithm 4: Temperature-COS-DPO with Penalization

```
Data: Base model $s_0(\boldsymbol{y}|\boldsymbol{x})$, dataset $\mathcal{D}_{\text{MOFT}}$, concentration parameter $\boldsymbol{\alpha}$, temperature distribution $\mathcal{D}_{\boldsymbol{\beta}}$, penalization
      coefficient $\lambda$; temperature $\boldsymbol{\beta}$, weight vector $\boldsymbol{w}$ (Post-Training Control).
Result: Fine-Tuned Model $s_\theta(\cdot, \cdot, \cdot|\boldsymbol{x})$.
  // Training
  for $e = 1$ to $N_{\text{steps}}$ do
      Sample $\boldsymbol{w}' \sim \text{Dir}(\boldsymbol{\alpha})$, $\boldsymbol{\beta}' \sim \mathcal{D}$;
      $\theta \leftarrow \theta - \eta\nabla_\theta[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}', \boldsymbol{\beta}'|\boldsymbol{x}); s_0, \mathcal{D}_{\text{MOFT}}) + \lambda\mathcal{G}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}', \boldsymbol{\beta}'|\boldsymbol{x}); s_0)]$;
  end
  // Post-Training Control
  $s_\theta(\boldsymbol{y}, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x}) \leftarrow \big(1 - \frac{1}{\|\boldsymbol{\beta}\|_1}\big)s_0(\boldsymbol{y}|\boldsymbol{x}) + \frac{1}{\|\boldsymbol{\beta}\|_1}s_\theta\left(\boldsymbol{y}, \boldsymbol{w}, \frac{\boldsymbol{\beta}}{\|\boldsymbol{\beta}\|_1}|\boldsymbol{x}\right)$.
```

We have also included the post-training control in both algorithms to scale the output of the model with the temperature $\boldsymbol{\beta}$ and the weight vector $\boldsymbol{w}$.

A.3 HYPERVOLUME METRIC

In MOO, the quality of an approximate Pareto front $\hat{\mathcal{P}}$ is often evaluated using the *hypervolume (HV)* metric, which measures the volume of the region dominated by $\hat{\mathcal{P}}$ relative to a predefined reference point $\boldsymbol{r}$. This reference point is typically chosen to be a point that is worse than all solutions in the objective space. The hypervolume provides an aggregate measure of performance by capturing how well $\hat{\mathcal{P}}$ extends toward optimal trade-offs among objectives.

The definition of the hypervolume differs based on whether the objective functions are being maximized or minimized:

- For maximization problems, where higher values are preferred, the hypervolume is computed as:

$$\text{HV}(\hat{\mathcal{P}}, \boldsymbol{r}) = \int_{\boldsymbol{x}\succeq\boldsymbol{r}} \mathbf{1}_{\exists \boldsymbol{p}\in\hat{\mathcal{P}}, \boldsymbol{p}\succeq\boldsymbol{x}}\, \mathrm{d}\boldsymbol{x}. \tag{20}$$

  In this case, the hypervolume measures the volume of the region above the reference point $\boldsymbol{r}$ that is dominated by the approximate Pareto front $\hat{\mathcal{P}}$.

- For minimization problems, where lower values are preferred, the hypervolume is defined as:

$$\mathrm{HV}(\hat{\mathcal{P}}, \boldsymbol{r}) = \int_{\boldsymbol{x} \preceq \boldsymbol{r}} \mathbf{1}_{\exists \boldsymbol{p} \in \hat{\mathcal{P}}, \boldsymbol{p} \preceq \boldsymbol{x}} \, \mathrm{d}\boldsymbol{x}. \tag{21}$$

Here, the hypervolume represents the volume of the region below the reference point $\boldsymbol{r}$ that is dominated by $\hat{\mathcal{P}}$.

Intuitively, a larger hypervolume value indicates a better approximation of the true Pareto front $\mathcal{P}$, as it suggests that $\hat{\mathcal{P}}$ spans a larger and more favorable region in the objective space, and the optimal Pareto front $\mathcal{P}$ possesses the maximum hypervolume.

A.4 DATASETS

We provide additional details on the datasets used in the experiments.

LTR Fine-Tuning Task. In this task, $\boldsymbol{x}^{(k)}$ in $\mathcal{D}_{\mathrm{MOFT}}$ denotes a query, and $\boldsymbol{y}_i^{(k)}$ denotes the feature vector of the i-th document, and $z_i^{j,(k)}$ denotes the score of the i-th document w.r.t. the j-th aspect.

The goal of LTR is to provide a ranking $\boldsymbol{\pi}$ of the documents w.r.t. the scores $z_i^{j,(k)}$ for each query $\boldsymbol{x}^{(k)}$, that maximizes the *Normalized Discounted Cumulative Gain (NDCG)* [Wang et al., 2013] metric, defined as

$$\mathrm{NDCG}^j@\mathrm{k}(\boldsymbol{\pi}) = \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}^j)} \left[\frac{\mathrm{DCG}@\mathrm{k}(\boldsymbol{\pi}, \boldsymbol{z}^j)}{\max_{\boldsymbol{\pi}'} \mathrm{DCG}@\mathrm{k}(\boldsymbol{\pi}', \boldsymbol{z}^j)} \right], \tag{22}$$

where the DCG@k, the discounted cumulative gain for the first k items, is defined as

$$\mathrm{DCG}@\mathrm{k}(\boldsymbol{\pi}, \boldsymbol{z}^j) = \sum_{i=1}^{k} \frac{z^j_{\pi_i}}{\log_2(i+1)}.$$

This metric intuitively measures the quality of the ranking $\boldsymbol{\pi}$ of the documents w.r.t. the scores $z_i^{j,(k)}$ for each query $\boldsymbol{x}^{(k)}$ by assigning higher weights to the top-ranked documents, normalized by the ideal ranking.

We adopt the Microsoft Learning-to-Rank Web Search (MSLR-WEB10K) dataset [Qin and Liu, 2013] for the LTR task. The MSLR-WEB10K dataset consists of 10,000 groups ($N = 10^4$), each containing a list of webpages retrieved by the search engine in response to the query $\boldsymbol{x}^{(k)}$ and the corresponding features extracted from the webpage. Following the practice by Mahapatra et al. [2023b], we treat the first 131 features as the feature vector ($\boldsymbol{y}_i^{(k)} \in \mathbb{R}^{131}$). We also identify the relevance label $\in [0 : 4]$ as the main objective used to train the base model, and the last 5 features, *viz.* (I) Query-URL Click Count, (II) URL Click Count, (III) URL Dwell Time, (IV) Quality Score 1, (V) Quality Score 2, with the relevance label, as 5 different auxiliary objectives ($m = 5$) for fine-tuning. The dataset is split into training (60%), validation (20%), and test (20%) datasets, and all results shown are on the test split.

We first train w.r.t. the relevance label sufficiently and treat it as our base model $s_0(\boldsymbol{y})$.

LLM Alignment Task. In this task, $\boldsymbol{x}^{(k)}$ in $\mathcal{D}_{\mathrm{MOFT}}$ denotes a prompt, and $\boldsymbol{y}_i^{(k)}$ denotes the response generated by the LLM, and $z_i^{j,(k)}$ denotes the score of the i-th response w.r.t. the j-th aspect. The goal is to align the LLM to generate responses that satisfy the auxiliary objectives (*e.g.*, verboseness, harmlessness, *etc.*) while maintaining its performance on general tasks (*e.g.*, fluency, relevance, *etc.*).

We adopt the PKU-SafeRLHF dataset [Ji et al., 2024a] for experiments, which consists of 83,400 entries, each containing a prompt and a pair of responses ($n = 2$) annotated with preferences w.r.t. both harmlessness and helpfulness ($m = 2$). When the k-th response is annotated as more helpful, we assign $z_1^{(k)} = 1$; otherwise, $z_1^{(k)} = 0$. Similarly, when the k-th response is annotated as more harmless, we assign $z_2^{(k)} = 1$; otherwise, $z_2^{(k)} = 0$. The goal is to fine-tune the model to generate responses that are both harmless and helpful as a multi-objective optimization problem.

A.5 REMARK ON TRAINING TIME

The training time in Table 1 refers to the duration of all training jobs required for computing the 11-point Pareto front. As described in Alg. 1 or its penalized version Alg. 3, in each epoch during the Weight-COS-DPO training, we first sample a single weight vector $\boldsymbol{w}$ and then compute the loss $\mathcal{L}_{\text{W-COS}}$ and back-propagate the gradients. Therefore, the training does not introduce additional computational cost compared to the training w.r.t. a single objective.

However, Weight-COS-DPO may require more training epochs to converge due to the exploration of the Pareto front. In practice, we find that the Weight-COS-DPO framework converges rapidly, and the training time may only be slightly longer than that of a single model training.

A.6 HYPERPARAMETER STUDIES

In this section, we provide the studies of the hyperparameters of the COS-DPO framework, including the sensitivity of the concentration parameter $\boldsymbol{\alpha}$ in the Dirichlet distribution, the depth of the model, and the performance of two different NN parametrizations $s_{\theta,\boldsymbol{w},\boldsymbol{\beta}}(\cdot,\cdot|\boldsymbol{x})$, namely training-from-scratch and augmentation network.

A.6.1 Concentration Parameter $\boldsymbol{\alpha}$

The concentration parameter $\boldsymbol{\alpha}$ controls the span of the Dirichlet distribution from which the weight vector $\boldsymbol{w}$ is sampled and is the key parameter affecting the performance of the COS-DPO framework that should be carefully selected and validated. By the basic properties of the Dirichlet distribution, suppose $\boldsymbol{w} \sim \mathrm{Dir}(\boldsymbol{\alpha})$, then we have

$$\mathbb{E}[\boldsymbol{w}] = \frac{\boldsymbol{\alpha}}{\|\boldsymbol{\alpha}\|_1} := \overline{\boldsymbol{\alpha}}, \quad \mathrm{var}(\boldsymbol{w}) = \frac{\mathrm{diag}(\overline{\boldsymbol{\alpha}}) - \overline{\boldsymbol{\alpha}}\overline{\boldsymbol{\alpha}}^\top}{\|\boldsymbol{\alpha}\|_1 + 1}.$$

In general, the smaller the $\boldsymbol{\alpha}$, the more likely the weight vector $\boldsymbol{w}$ is close to the boundary of the simplex, and the larger the $\boldsymbol{\alpha}$, the more likely the weight vector $\boldsymbol{w}$ is concentrated around the expectation $\overline{\boldsymbol{\alpha}}$.

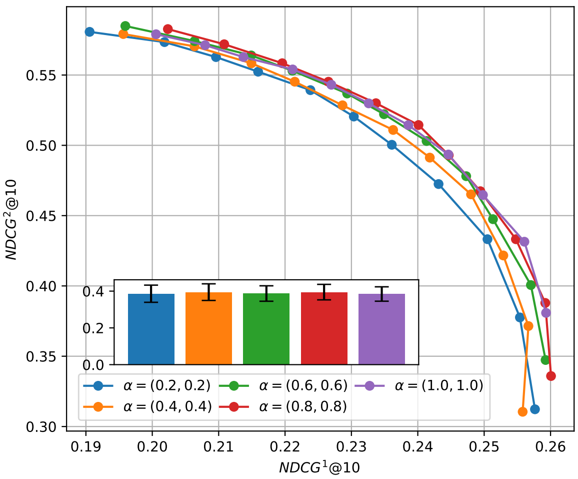
As the COS-DPO framework is generally robust to the choice of the concentration parameter $\boldsymbol{\alpha}$, we conduct ablation studies to investigate the impact of the concentration parameter $\boldsymbol{\alpha}$ on the performance of the COS-DPO framework in different settings. We first conduct experiments on the MSLR-WEB10K dataset with 2 auxiliary objectives (Query-URL Click Count vs URL Click Count) to investigate the impact of the concentration parameter $\boldsymbol{\alpha}$ on the performance of the COS-DPO framework. The results are shown in Fig. 7. The experiment settings and plotting details are the same as in the main text.

As shown in Fig. 7a, as the concentration parameter $\boldsymbol{\alpha}$ decreases, COS-DPO obtains a visually more comprehensive Pareto front thanks to more samples close to the boundary of the simplex. However, it is at the cost of a slightly undertrained model across the simplex, indicated by a lower hypervolume metric. It turns out that the choice of $\boldsymbol{\alpha}$ faces a trade-off between the diversity of the samples and the overall quality of the fine-tuning, given a fixed training budget. Similar trade-offs are observed in Fig. 7b and 7c when only one dimension of the concentration parameter $\boldsymbol{\alpha}$ is varied.

We also conducted experiments on the PKU-SafeRLHF dataset to investigate the impact of the concentration parameter $\boldsymbol{\alpha}$ on the performance of the COS-DPO framework on the LLM alignment task. The results are shown in Fig. 8. A similar pattern is observed in this large-scale task, where a smaller choice of the concentration parameter $\boldsymbol{\alpha}$ leads to a more comprehensive Pareto front. However, it does not necessarily lead to a worse hypervolume metric, suggesting that the performance of COS-DPO here is less hindered by the expressive power of the model, which has already been abundant in the LLM, but rather by the diversity of the samples.

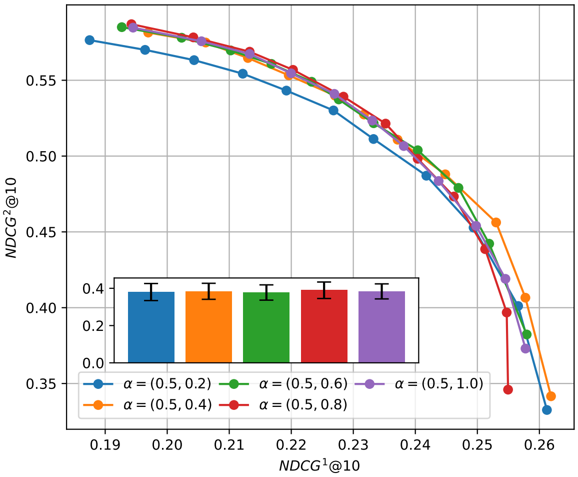
A.6.2 Model Depth

The depth of the neural network architecture is also crucial for the performance of the COS-DPO framework, as it determines the complexity and the expressiveness of the model. We also use the MSLR-WEB10K dataset with 2 auxiliary objectives (Query-URL Click Count vs URL Click Count) to investigate the impact of *the model depth* on the performance of the COS-DPO framework. The results are shown in Fig. 10a, where the depth, referring to the number of transformer layers in the model, is varied from 1 to 5. As shown in the figure, the performance of the COS-DPO framework is first significantly improved and gradually saturated with the increase of depth. Besides, while the hypervolume metric improves, the coverage of the Pareto front does not change significantly with the increase in depth. This suggests that the concentration parameter $\boldsymbol{\alpha}$ may have a more significant impact on the diversity of the samples than the model depth.



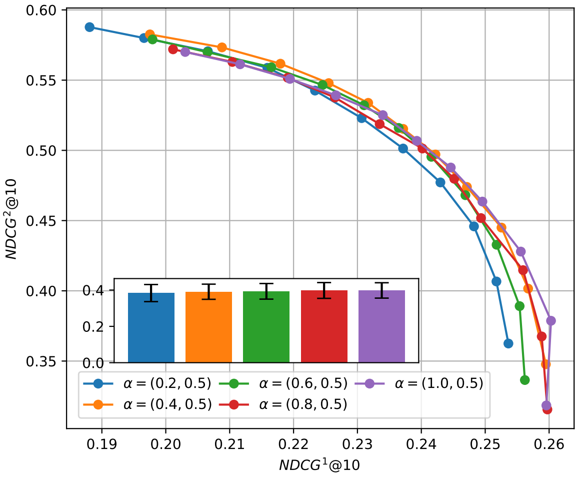
$\boldsymbol{\alpha}$	**Hypervolume**
$(0.2, 0.2)$	1.446×10^{-1}
$(0.4, 0.4)$	1.445×10^{-1}
$(0.6, 0.6)$	1.471×10^{-1}
$(0.8, 0.8)$	$\mathbf{1.473 \times 10^{-1}}$
$(1.0, 1.0)$	1.463×10^{-1}

(a) $\boldsymbol{\alpha} = (\alpha, \alpha)$ for $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.



$\boldsymbol{\alpha}$	**Hypervolume**
$(0.5, 0.2)$	1.451×10^{-1}
$(0.5, 0.4)$	$\mathbf{1.474 \times 10^{-1}}$
$(0.5, 0.6)$	1.466×10^{-1}
$(0.5, 0.8)$	1.458×10^{-1}
$(0.5, 1.0)$	1.464×10^{-1}

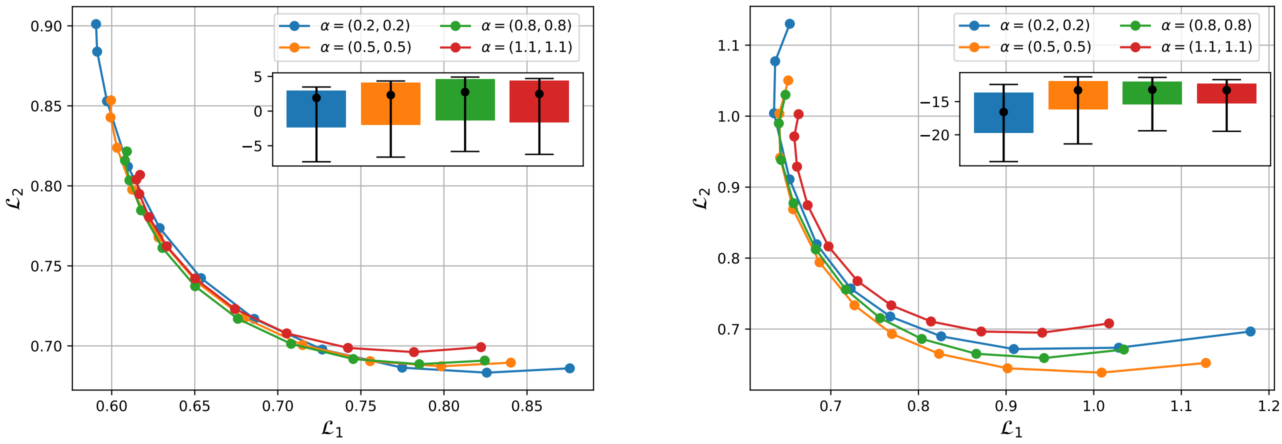
(b) $\boldsymbol{\alpha} = (0.5, \alpha)$ for $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.



$\boldsymbol{\alpha}$	**Hypervolume**
$(0.2, 0.5)$	1.447×10^{-1}
$(0.4, 0.5)$	$\mathbf{1.468 \times 10^{-1}}$
$(0.6, 0.5)$	1.445×10^{-1}
$(0.8, 0.5)$	1.444×10^{-1}
$(1.0, 0.5)$	1.445×10^{-1}

(c) $\boldsymbol{\alpha} = (\alpha, 0.5)$ for $\alpha \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$.

Figure 7: Hyperparameter study on the impact of concentration parameter $\boldsymbol{\alpha}$ on the Pareto fronts obtained by Weight-COS-DPO on the MSLR-WEB10K dataset (Objective I vs Objective II) with different settings of $\boldsymbol{\alpha}$. The hypervolume metric is shown in the table beside each figure.



(a) GPT2

(b) Alpaca-7B-Reproduced

Figure 8: Hyperparameter study on the impact of the concentration parameter $\boldsymbol{\alpha}$ on the Pareto fronts obtained by Weight-COS-DPO on the PKU-SafeRLHF dataset.

A.6.3 Model Parametrization

In general, one could adopt one of the two different parametrizations of $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ (or $s_\theta(\cdot, \boldsymbol{w}, \boldsymbol{\beta}|\boldsymbol{x})$, respectively) in the COS-DPO framework. For simplicity, we will only discuss the case of Weight-COS-DPO in the following, and the discussion can be easily extended to the Temperature-COS-DPO method naturally.

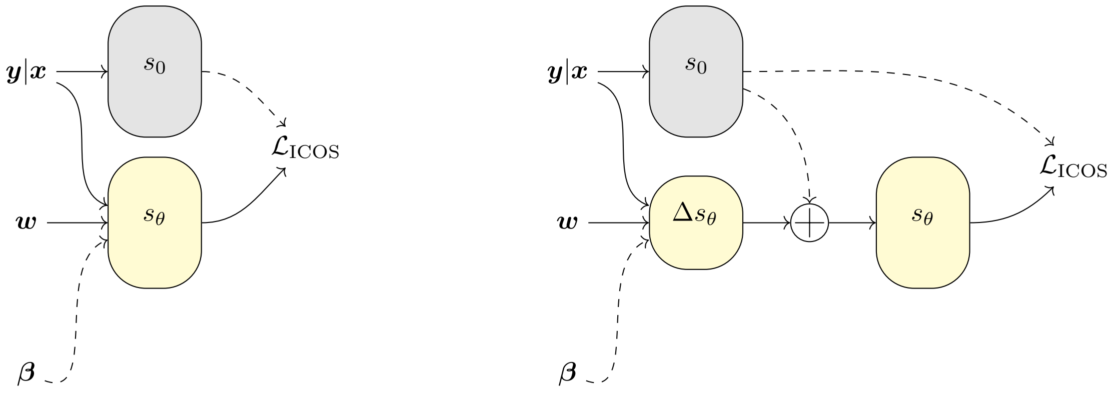
- *Training-from-Scratch*: The model $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ is a completely separate neural network from the base model $s_0(\boldsymbol{y}|\boldsymbol{x})$. Depending on the specific design of the additional inputs $\boldsymbol{w}$, the new model may or may not share the same architecture as the base model. The main advantage of this design is that it requires less memory and computation resources [Rafailov et al., 2024b], and thus is more suitable for large-scale applications, *e.g.*, LLMs.
- *Augmentation Network*: As several works [Chen et al., 2024, Xu et al., 2024] argue that DPO is prone to overfitting, one may curb the complexity of the model for the score function $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ by only adding a first-order correction term to the base model $s_0(\boldsymbol{y}|\boldsymbol{x})$ as:
$$s_\theta(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) = s_0(\boldsymbol{y}|\boldsymbol{x}) + \Delta s_\theta(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}),$$
where the parameters in the base model are fixed, and the importance-conditioned design is only applied to the correction term $\Delta s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$. This design allows limited modification and reversibility to the base model and is thus suitable for applications where the fine-tuning is limited in budget, frequent, or expected to be minor.

The two parametrizations are illustrated in Fig. 9a and 9b, respectively. Both parametrizations can be seamlessly applied to the COS-DPO framework and easily switch between each other. In all the experiments presented in the main text, we have adopted the training-from-scratch design for the COS-DPO framework. Fig. 10b shows the results of the COS-DPO framework with the augmentation training design on the same task as the previous ablation studies. Compared with Fig. 10a, the augmentation training achieves a roughly better performance than the training-from-scratch design with the same depth, coinciding with the intuition that the augmentation training benefited from the information provided by the base model and instead of learning the entire score function $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ from scratch, it only needs to learn the correction term $\Delta s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$. When the model depth is increased, the performance of the augmentation training is also improved, sharing the same trend as the training-from-scratch design.

A.7 ADDITIONAL RESULTS ON LINEAR TRANSFORMATION PROPERTY

As discussed in Sec. 3.2, the linear transformation property implies that the model can be scaled proportionally by a constant factor c by a simple linear transformation of the output scores. We provide two relevant experiments on the post-training controls of the Weight-COS networks obtained by Weight-COS-DPO based on this property.

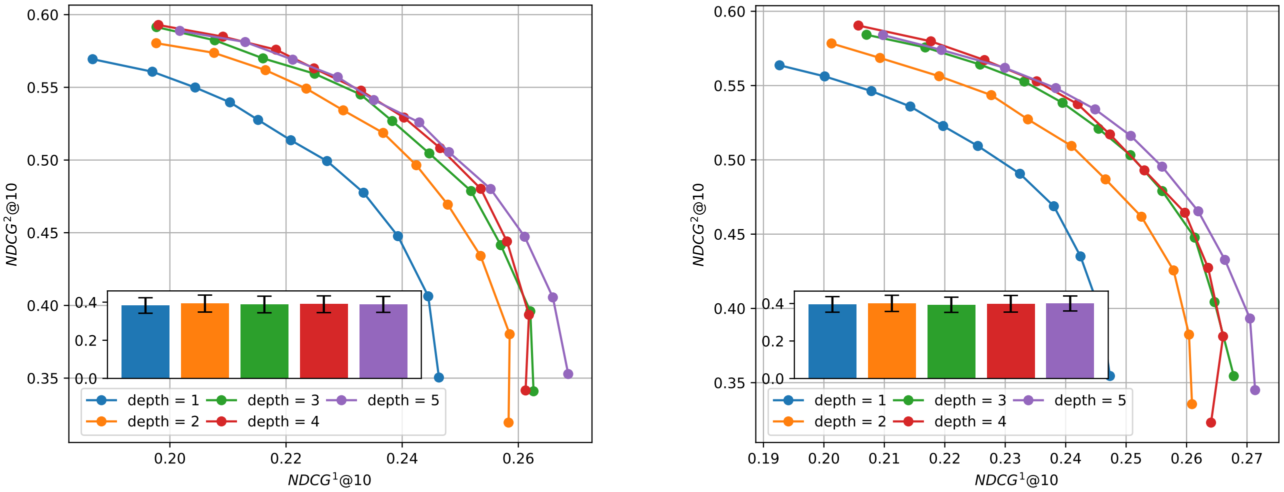
Fig. 11 gives examples of the post-training control over the temperature $\boldsymbol{\beta}$ on the MSLR-WEB10K dataset with 2 auxiliary



(a) Training-from-Scratch

(b) Augmentation Network

Figure 9: Illustration of two different parametrizations of the model $s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x})$ in the COS-DPO framework. Dashed lines denote that backpropagation is not applied.



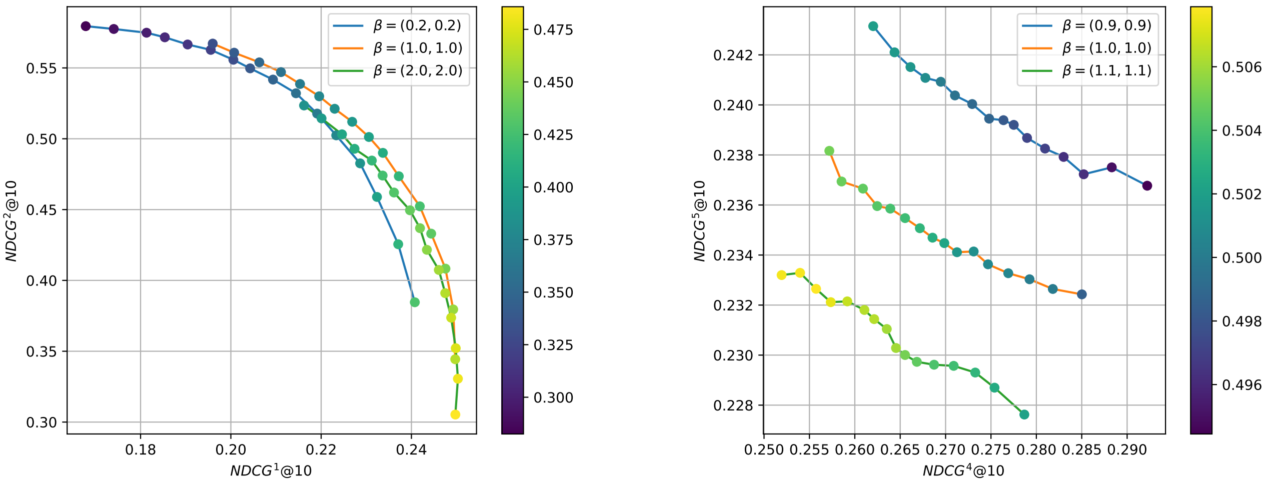
(a) Training-from-Scratch

(b) Augmentation Network

Figure 10: Study on the impact of the model depth and the model parametrizations on the Pareto fronts obtained by Weight-COS-DPO on the MSLR-WEB10K dataset (Objective I vs Objective II).

objectives. As the temperature $\boldsymbol{\beta}$ increases, the Pareto front shifts towards the direction where the main objective is more emphasized, which is consistent with our expectations. In Fig. 11b, the two auxiliary objectives are in balance, and thus, the shifts of the Pareto fronts resemble that depicted in Fig. 1. However, in Fig. 11a, the unexpected shifting pattern is observed, which may reflect the complex interactions between the main and auxiliary objectives.

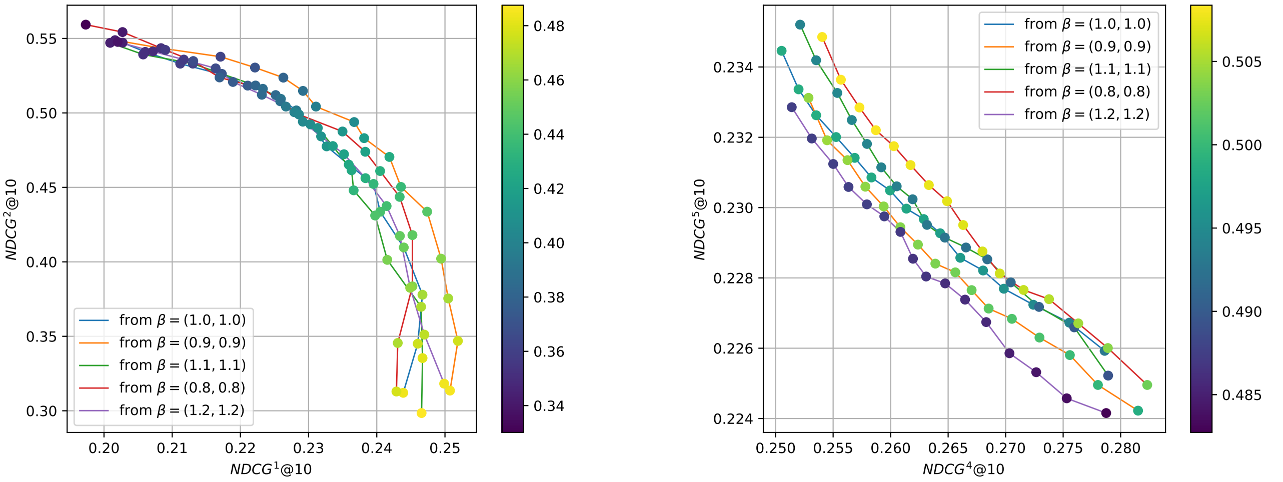
Fig. 12 provides empirical validation of the linear transformation property on the MSLR-WEB10K dataset with 2 auxiliary objectives. The methodology is that we first train a Weight-COS network with the different temperatures $\boldsymbol{\beta}$ ranging from $(0.8, 0.8)$ to $(1.2, 1.2)$, and then transform the Pareto fronts obtained by the trained models to the same temperature $\boldsymbol{\beta} = (1, 1)$ using the post-training control as indicated in (26) and Alg. 3. The penalization coefficient λ is set to 0.05 in the training. The results show that the transformed Pareto fronts are roughly aligned with each other, which validates the linear transformation property of the model. The slight deviation may be caused by the noises in the training process and the non-uniqueness of the optimal solutions of the Weight-COS-DPO loss.



(a) Objective I vs Objective II.

(b) Objective IV vs Objective V.

Figure 11: Examples of post-training control of Weight-COS networks over temperature $\boldsymbol{\beta}$ on the MSLR-WEB10K dataset with 2 auxiliary objectives. Two axes denote the NDCG@10 of the two auxiliary objectives (the higher, the better). The colorbar denotes the NDCG@10 of the main objective.



(a) Objective I vs Objective II.

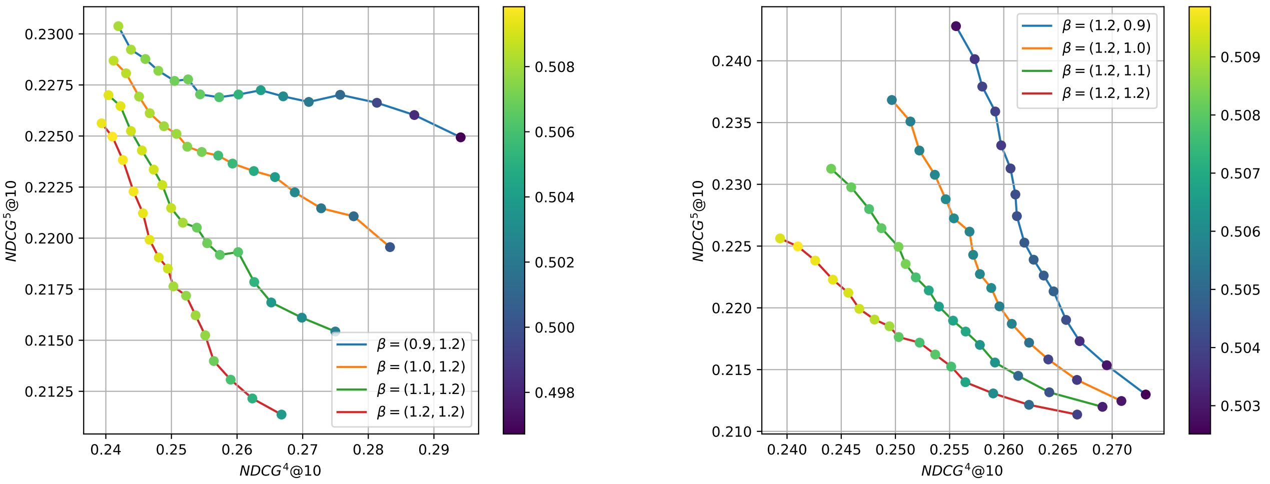
(b) Objective IV vs Objective V.

Figure 12: Empirical validation of the linear transformation property of Weight-COS networks on the MSLR-WEB10K dataset with 2 auxiliary objectives. The Pareto fronts in the figures are obtained by first training a model with the temperature $\boldsymbol{\beta}$ in the legend and then transforming to the same temperature $\boldsymbol{\beta} = (1, 1)$ using post-training controls. Two axes denote the NDCG@10 of the two auxiliary objectives (the higher, the better). The colorbar denotes the NDCG@10 of the main objective.

A.8 ADDITIONAL RESULTS OF TEMPERATURE-COS-DPO

All experiments of the Temperature-COS-DPO method are conducted on the MSLR-WEB10K dataset with 2 auxiliary objectives (Quality Score vs Quality Score 2) to investigate the performance of the Temperature-COS-DPO, as it provides better visualization and comparisons of the Pareto fronts with different temperature parameters $\boldsymbol{\beta}$. In particular, we adopt the augmentation network design for Temperature-COS networks for better expressive power and stability.

We provide the results of the Temperature-COS-DPO method on the LTR fine-tuning task in Fig. 13. The model depth is chosen to be 5, and the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ is set to be $\mathrm{Unif}([0.67, 1.5]^2)$. The results demonstrate the Temperature-COS-



(a) $\boldsymbol{\beta} = (\beta, 1.2)$ for $\beta \in \{0.9, 1.0, 1.1, 1.2\}$.

(b) $\boldsymbol{\beta} = (1.2, \beta)$ for $\beta \in \{0.9, 1.0, 1.1, 1.2\}$.

Figure 13: Preliminary results of Temperature-COS-DPO on the MSLR-WEB10K dataset (Objective IV vs Objective V). The colorbar denotes the NDCG@10 of the main objective.

DPO method is capable of capturing the trade-off between the main objective and the auxiliary objectives for all kinds of temperature configurations $\boldsymbol{\beta}$, and the Pareto fronts exhibit expected behaviors with different $\boldsymbol{\beta}$. These results suggest that temperature-conditioned one-shot fine-tuning is a promising direction for the COS-DPO framework to achieve more flexible control over the Pareto front.

Given the choices of the temperature parameters, the Pareto fronts in both **??** and **??** should merge into one single point, which refers to the solution of the single-objective fine-tuning task with certain temperature parameter β. Although the results are roughly in accordance with the theoretical expectations, there are still small gaps that may be accounted for by the limit of the expressive power of the model and insufficient exploration over the weight vector $\boldsymbol{w}$.

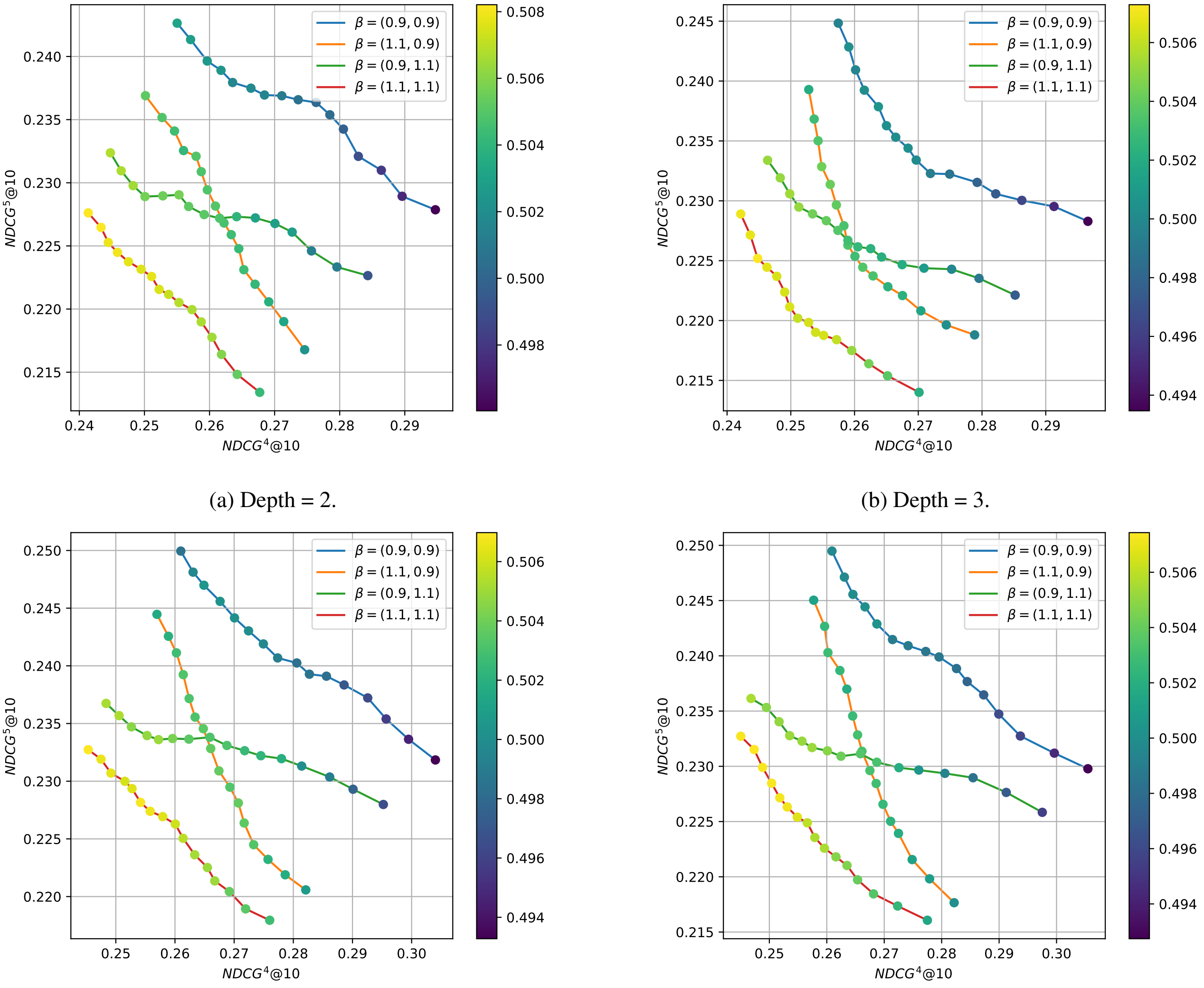
To explain this, we present ablation studies to investigate the effect of the expressiveness of the model on the performance of the Temperature-COS-DPO. We applied models with 2 to 5 layers of transformer architecture, and the results show that the performance, indicated by the expected behaviors of the Pareto front, is drastically improved with the increase of the number of layers. While swallower models yield Pareto fronts with less expected behaviors and more noise, *e.g.*, the concavity of the Pareto fronts in Fig. 14b partially indicates the insufficiency of the training process, the model with 5 layers of transformer architecture in Fig. 14d exhibits improved scores and more expected behaviors according to different temperature configurations. This suggests and confirms the intuition that Temperature-COS-DPO requires more expressive structures to capture the complex trade-offs between the main and auxiliary objectives.

The choice of the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ also affects the performance of Temperature-COS-DPO. Fig. 15 shows the impact of the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ on the Pareto fronts obtained by the Temperature-COS-DPO on the MSLR-WEB10K dataset. When the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ only covers a small range, the Pareto fronts exhibit less expected behaviors and more noise, *e.g.*, the Pareto fronts with $\mathcal{D}_{\boldsymbol{\beta}} = \mathrm{Unif}([0.83, 1.2]^2)$ in Fig. 15a are less concave and more scattered. As the range of the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ increases, the Pareto fronts become more concave and exhibit less desired behaviors. The results suggest that the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ should cover a larger range than those interested to ensure sufficient training.

Given the results shown above and our studies on several hyperparameters, we conclude that despite requiring more expressive structures and more training resources, Temperature-COS-DPO is a feasible and promising direction for the COS-DPO framework to achieve more flexible control over the Pareto front and we expect to further investigate the validity of Temperature-COS networks and apply them to more complex multi-objective optimization tasks in the future.

B MISSING PROOFS

In this section, we provide the proofs of the propositions and theorems mentioned in the main text.



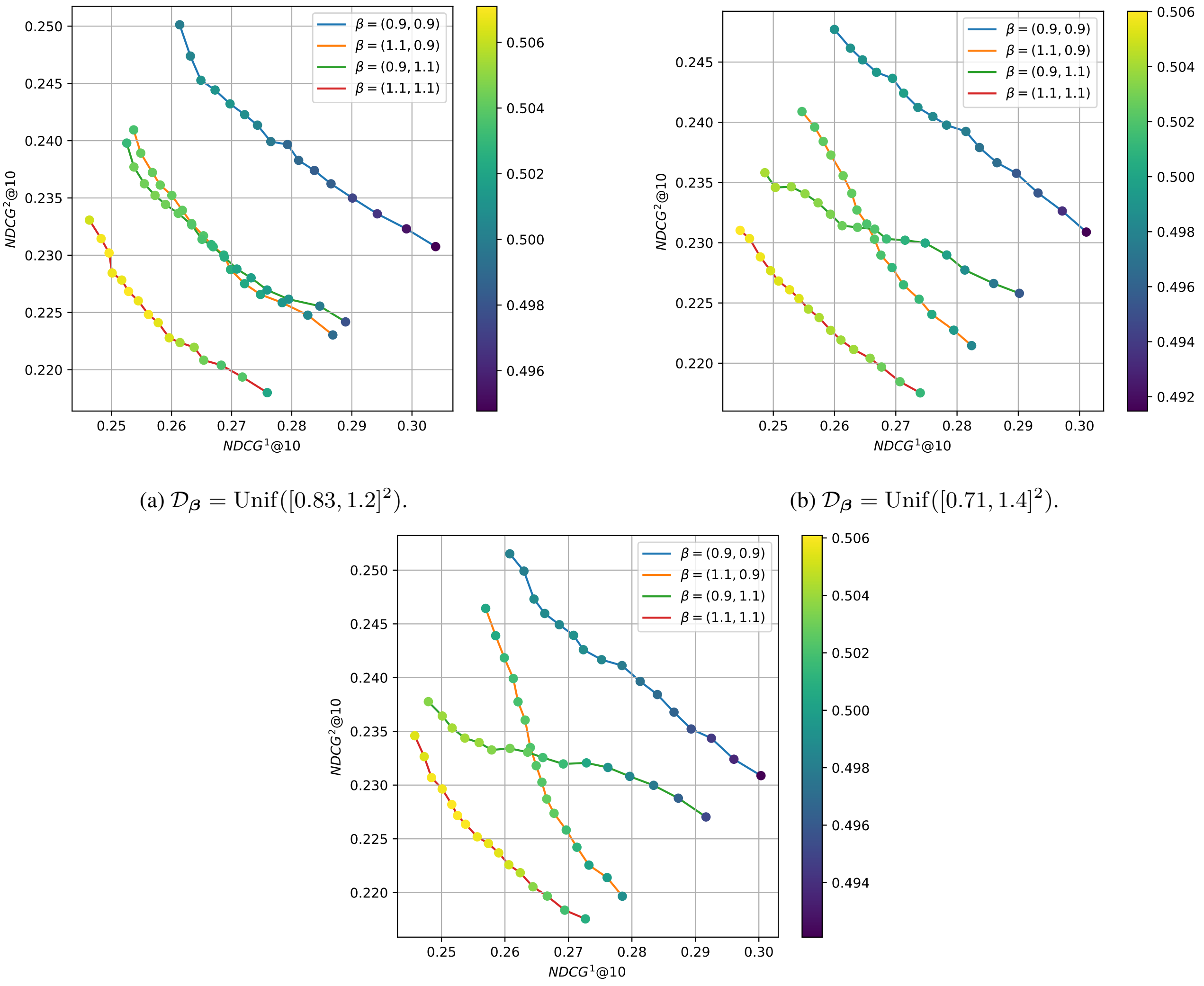
(a) Depth = 2.

(b) Depth = 3.

(c) Depth = 4.

(d) Depth = 5.

Figure 14: Ablation study of the impact of model depth on the Pareto fronts obtained by the Temperature-COS-DPO on the MSLR-WEB10K dataset (Objective IV vs Objective V). The colorbar denotes the NDCG@10 of the main objective.



(a) $\mathcal{D}_{\boldsymbol{\beta}} = \mathrm{Unif}([0.83, 1.2]^2)$.

(b) $\mathcal{D}_{\boldsymbol{\beta}} = \mathrm{Unif}([0.71, 1.4]^2)$.

(c) $\mathcal{D}_{\boldsymbol{\beta}} = \mathrm{Unif}([0.63, 1.6]^2)$.

Figure 15: Ablation study of the impact of the distribution $\mathcal{D}_{\boldsymbol{\beta}}$ on the Pareto fronts obtained by the Temperature-COS-DPO on the MSLR-WEB10K dataset (Objective IV vs Objective V). The colorbar denotes the NDCG@10 of the main objective.

B.1 PROOFS OF REPARAMETRIZATION-RELATED ARGUMENTS

We prove the reparametrization of the DPO loss (5) and the LiPO loss (8) below.

Proof of (5). Recall that in the second step of PPO, we consider the loss function (4) as follows:

$$\begin{aligned} -\mathcal{L}(p_\theta; p_0, r_\phi, \beta) &= \mathbb{E}_{(\boldsymbol{x},y)} \left[r_\phi(y|\boldsymbol{x}) - \beta \log \frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})} \right] \\ &= \int \left(r_\phi(y|\boldsymbol{x}) - \beta \log \frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})} \right) p_\theta(y|\boldsymbol{x}) \mathrm{d}y, \end{aligned}$$

we calculate the functional derivative of the loss w.r.t. the density function $p_\theta(y|\boldsymbol{x})$:

$$
\begin{aligned}
&\frac{\delta\mathcal{L}(p_\theta;p_0,r_\phi,\beta)}{\delta p_\theta(y|\boldsymbol{x})} = \lim_{\epsilon\to 0}\frac{\mathcal{L}(p_\theta+\epsilon\delta p_\theta;p_0,r_\phi,\beta)-\mathcal{L}(p_\theta;p_0,r_\phi,\beta)}{\epsilon}\\
&=\lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[\int\left(r_\phi(y|\boldsymbol{x})-\beta\log\frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})}-\beta\frac{\epsilon\delta p_\theta(y|\boldsymbol{x})}{p_\theta(y|\boldsymbol{x})}\right)(p_\theta(y|\boldsymbol{x})+\epsilon\delta p_\theta(y|\boldsymbol{x}))\mathrm{d}y\right.\\
&\qquad\left.-\int\left(r_\phi(y|\boldsymbol{x})-\beta\log\frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})}\right)p_\theta(y|\boldsymbol{x})\mathrm{d}y\right]\\
&=\int\left(r_\phi(y|\boldsymbol{x})-\beta\log\frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})}-\beta\right)\delta p_\theta(y|\boldsymbol{x})\mathrm{d}y.
\end{aligned}
$$

Let the functional derivative vanish, we obtain

$$
r_\phi(y|\boldsymbol{x})=\beta\log\frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})}+\beta,
$$

i.e.,

$$
p_\theta(y|\boldsymbol{x})\ \propto\ p_0(y|\boldsymbol{x})\exp\left(\frac{r_\phi(y|\boldsymbol{x})}{\beta}\right).
$$

Since the likelihood $\mathbb{P}(y_1\succ y_2|\boldsymbol{x})$ (2) in the BTL model only depends on the difference of the reward functions, $r_\phi(y|\boldsymbol{x})$ admits an arbitrary constant shift, and thus we assume $r_\phi(y|\boldsymbol{x})$ to be normalized in a way such that

$$
\mathbb{E}\left[p_0(y|\boldsymbol{x})\exp\left(\frac{r_\phi(y|\boldsymbol{x})}{\beta}\right)\right]=1,
$$

which leads to the reparametrization $r_\theta(y|\boldsymbol{x})\ =\ \beta\log\frac{p_\theta(y|\boldsymbol{x})}{p_0(y|\boldsymbol{x})}$, plugging which into the PPO loss (4) yields the DPO loss (5). □

Proof of (8). As in the derivation of the DPO loss (5) under the BTL model, we first consider the PPO algorithm for the PL model:

Step 1. Find the optimal score function $s_\phi(\boldsymbol{y}|\boldsymbol{x})$ that minimizes the ListNet loss (7):

$$
-\mathcal{L}_{\mathrm{LN}}(s_\theta;\mathcal{D}^j_{\mathrm{LTR}})=\mathbb{E}\left[\sum_{i=1}^n\bar{z}^j_i\log\frac{\exp(s_\phi(\boldsymbol{y}_i|\boldsymbol{x}))}{\sum_{i'=1}^n\exp(s_\phi(\boldsymbol{y}_{i'}|\boldsymbol{x}))}\right];\tag{23}
$$

Step 2. Fine-tune the base model s_0 with the optimal score function s_ϕ by maximizing the expected score value while penalizing the KL divergence between the new model and the base model:

$$
-\mathcal{L}(p_\theta;p_0,r_\phi,\beta)=\mathbb{E}\left[s_\phi(\boldsymbol{y}|\boldsymbol{x})\right]-\beta D_{\mathrm{KL}}(p_\theta||p_0)=\mathbb{E}\left[s_\phi(\boldsymbol{y}|\boldsymbol{x})-\beta\log\frac{p_\theta(\boldsymbol{y}|\boldsymbol{x})}{p_0(\boldsymbol{y}|\boldsymbol{x})}\right].\tag{24}
$$

For the optimization problem in the second step (24), following the same procedure as in the proof of (5), we solve the optimal p_θ by letting the functional derivative of the loss w.r.t. the density function $p_\theta(y|\boldsymbol{x})$ vanish and obtain

$$
p_\theta(\boldsymbol{y}|\boldsymbol{x})\ \propto\ p_0(\boldsymbol{y}|\boldsymbol{x})\exp\left(\frac{s_\phi(\boldsymbol{y}|\boldsymbol{x})}{\beta}\right).\tag{25}
$$

By the assumption of the PL model and the ListNet loss, we have $p_\theta(\boldsymbol{y}|\boldsymbol{x})$ modeled as the top-1 probability of the PL model and thus related to the score function $s_\theta(\boldsymbol{y}|\boldsymbol{x})$ via

$$
p_\theta(\boldsymbol{y}|\boldsymbol{x})=\frac{\exp(s_\theta(\boldsymbol{y}|\boldsymbol{x}))}{\sum_{i'=1}^n\exp(s_\theta(\boldsymbol{y}_{i'}|\boldsymbol{x}))}.
$$

Let

$$
p_0(\boldsymbol{y}|\boldsymbol{x})=\frac{\exp(s_0(\boldsymbol{y}|\boldsymbol{x}))}{\sum_{i'=1}^n\exp(s_0(\boldsymbol{y}_{i'}|\boldsymbol{x}))},
$$

(25) can be rewritten as

$$\exp(s_\theta(\boldsymbol{y}|\boldsymbol{x})) \ \propto\ \exp\left(s_0(\boldsymbol{y}|\boldsymbol{x}) + \beta s_\phi(\boldsymbol{y}|\boldsymbol{x})\right),$$

i.e.,

$$s_\theta(\boldsymbol{y}|\boldsymbol{x}) = s_0(\boldsymbol{y}|\boldsymbol{x}) + \beta s_\phi(\boldsymbol{y}|\boldsymbol{x}) + C,$$

where C is a constant shift. By noticing that the softmax function in (23) is invariant to the constant shift of the score function $s_\phi(\boldsymbol{y}|\boldsymbol{x})$, we may choose certain normalization such that

$$s_\theta(\boldsymbol{y}|\boldsymbol{x}) = s_0(\boldsymbol{y}|\boldsymbol{x}) + \beta s_\phi(\boldsymbol{y}|\boldsymbol{x})$$

holds, plugging which into the loss (23) yields the reparametrized ListNet loss (8). □

B.2 PROOFS OF LINEAR TRANSFORMATION PROPERTY

In this section, we provide the proof of the linear transformation property of the Weight-COS-DPO loss. Instead of Prop. 3.1 in the main text, we provide a more general proposition that considers the penalization terms in the Weight-COS-DPO loss introduced in App. A.2. The takeaway of this generalization is that the linear transformation property still holds whenever the penalization term is a function of the normalized loss function $\mathcal{L}_{\text{LiPO}}$.

Proposition B.1 (Linear Transformation Property with Penalization Terms)**.** *For any $\boldsymbol{\beta} \in \mathbb{R}_+^m$ and $\boldsymbol{w} \in \Delta^m$, we denote the model obtained by optimizing the Weight-COS-DPO loss (12) with temperature $\boldsymbol{\beta}$ as $s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})$, and suppose the penalization term $\mathcal{G}_{\boldsymbol{w}}(s_\theta; s_0, \boldsymbol{\beta})$ is a function of $\boldsymbol{\mathcal{L}}_{\text{LiPO}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})$.*

Then $s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})$ should satisfy the linear transformation that for any $c > 0$, we have that

$$s_{\theta,c\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) = \left(1 - \frac{1}{c}\right) s_0(\boldsymbol{y}|\boldsymbol{x}) + \frac{1}{c} s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) \tag{26}$$

is also an optimal solution to the Weight-COS-DPO loss (12) with temperature $c\boldsymbol{\beta}$.

Proof of Prop. B.1. For clarity, we first remove the penalization, *i.e.* to consider the case where $\lambda = 0$.

Then the Weight-COS-DPO loss (12) is of the following form:

$$\begin{aligned}
&\mathcal{L}_{\text{W-COS}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}}) \\
=&\mathbb{E}_{\boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}\left[\mathcal{L}_{\boldsymbol{w}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})\right] \\
=&\mathbb{E}_{\boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}\left[\sum_{j=1}^m w_j \mathcal{L}_{\text{LiPO}}(s_\theta(\cdot, \boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}^j_{\text{MOFT}})\right] \\
=&\mathbb{E}_{\boldsymbol{w}\sim\text{Dir}(\boldsymbol{\alpha})}\left[\sum_{j=1}^m w_j \mathbb{E}\left[\sum_{i=1}^n \bar{z}_i^j \log\left(\frac{\exp\left(\beta_j(s_\theta(\boldsymbol{y}_i, \boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_i, \boldsymbol{w}|\boldsymbol{x}))\right)}{\sum_{i'=1}^n \exp\left(\beta_j(s_\theta(\boldsymbol{y}_{i'}, \boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'}, \boldsymbol{w}|\boldsymbol{x}))\right)}\right)\right]\right] \\
=&\mathbb{E}\left[\sum_{j=1}^m \sum_{i=1}^n w_j \bar{z}_i^j \log\left(\frac{\exp\left(\beta_j(s_\theta(\boldsymbol{y}_i, \boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_i, \boldsymbol{w}|\boldsymbol{x}))\right)}{\sum_{i'=1}^n \exp\left(\beta_j(s_\theta(\boldsymbol{y}_{i'}, \boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'}, \boldsymbol{w}|\boldsymbol{x}))\right)}\right)\right],
\end{aligned}$$

where the expectation in the second to last equality is taken over the data distribution $\mathcal{D}_{\text{MOFT}}$, and the expectation in the last equality is taken over both the data distribution $\mathcal{D}_{\text{MOFT}}$ and the weight distribution $\text{Dir}(\boldsymbol{\alpha})$ as a shorthand notation.

By the definition of the model $s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x})$, we have that

$$s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y}, \boldsymbol{w}|\boldsymbol{x}) = \underset{s_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x})}{\arg\max}\, \mathbb{E}\left[\sum_{j=1}^m \sum_{i=1}^n w_j \bar{z}_i^j \log\left(\frac{\exp\left(\beta_j(s_\theta(\boldsymbol{y}_i, \boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_i, \boldsymbol{w}|\boldsymbol{x}))\right)}{\sum_{i'=1}^n \exp\left(\beta_j(s_\theta(\boldsymbol{y}_{i'}, \boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'}, \boldsymbol{w}|\boldsymbol{x}))\right)}\right)\right].$$

We now consider the following reparametrized optimization problem:

$$\begin{aligned}
&\max_{s'_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x})} \mathbb{E}\Bigg[\sum_{j=1}^{m}\sum_{i=1}^{n} w_j \overline{z}_i^j \log\left(\frac{\exp\left(\beta_j(cs'_\theta(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}) + (1-c)s_0(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}))\right)}{\sum_{i'=1}^{n}\exp\left(\beta_j(cs'_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) + (1-c)s_0(\boldsymbol{y}_{i'},\boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'},\boldsymbol{w}|\boldsymbol{x}))\right)}\right)\Bigg] \\
&= \mathbb{E}\left[\sum_{j=1}^{m}\sum_{i=1}^{n} w_j \overline{z}_i^j \log\left(\frac{\exp\left(c\beta_j(s'_\theta(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}))\right)}{\sum_{i'=1}^{n}\exp\left(c\beta_j(s'_\theta(\boldsymbol{y}_{i'},\boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'},\boldsymbol{w}|\boldsymbol{x}))\right)}\right)\right],
\end{aligned} \tag{27}$$

obtained by reparametrizing $s_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x})$ as

$$s_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) = cs'_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) + (1-c)s_0(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}), \tag{28}$$

and thus by solving

$$s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) = cs'_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) + (1-c)s_0(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}),$$

we have

$$s'_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) = \frac{1}{c}s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) - \frac{1-c}{c}s_0(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) \tag{29}$$

is an optimal solution to the reparametrized optimization problem.

Notice that the function in the optimization problem (27) is exactly the Weight-COS-DPO loss (12) with the temperature $c\boldsymbol{\beta}$, we have that the $s_{\theta,c\boldsymbol{\beta}}(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x})$ as defined in (26) coincides with the optimal solution (29). Thus we have proved the linear transformation property for the Weight-COS-DPO loss with $\lambda = 0$.

For the case with penalization, we assume the penalization term $\mathcal{G}_{\boldsymbol{w}}(s_\theta; s_0, \boldsymbol{\beta})$ is a function of the vector of LiPO losses $\boldsymbol{\mathcal{L}}_{\text{LiPO}}(s_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})$, which is satisfied for the cosine similarity penalization loss (18) as proposed by Ruchte and Grabocka [2021]. And in turn, the vector of LiPO losses $\boldsymbol{\mathcal{L}}_{\text{LiPO}}(s_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}); s_0, \boldsymbol{\beta}, \mathcal{D}_{\text{MOFT}})$ depends on $s_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x})$ only in the form of $s_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}) - s_0(\cdot,\boldsymbol{w}|\boldsymbol{x})$, and therefore, we could write the Weight-COS-DPO loss in an abstract form as

$$s_{\theta,\boldsymbol{\beta}}(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x}) = \underset{s_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x})}{\arg\max}\,\mathbb{E}\left[\Phi\left(s_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}) - s_0(\cdot,\boldsymbol{w}|\boldsymbol{x})\right)\right],$$

e.g., for the case where $\lambda = 0$, Φ is of the following form:

$$\begin{aligned}
&\Phi(s_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}) - s_0(\cdot,\boldsymbol{w}|\boldsymbol{x})) \\
&= \sum_{j=1}^{m}\sum_{i=1}^{n} w_j \overline{z}_i^j \log\left(\frac{\exp\left(\beta_j(s_\theta(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_i,\boldsymbol{w}|\boldsymbol{x}))\right)}{\sum_{i'=1}^{n}\exp\left(\beta_j(s_\theta(\boldsymbol{y}_{i'},\boldsymbol{w}|\boldsymbol{x}) - s_0(\boldsymbol{y}_{i'},\boldsymbol{w}|\boldsymbol{x}))\right)}\right).
\end{aligned}$$

Apply the same reparametrization as in (28), we have that the reparametrized optimization problem is of the form

$$\begin{aligned}
&\max_{s'_\theta(\boldsymbol{y},\boldsymbol{w}|\boldsymbol{x})} \mathbb{E}\left[\Phi\left(cs'_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}) + (1-c)s_0(\cdot,\boldsymbol{w}|\boldsymbol{x}) - s_0(\cdot,\boldsymbol{w}|\boldsymbol{x})\right)\right] \\
&= \mathbb{E}\left[\Phi\left(cs'_\theta(\cdot,\boldsymbol{w}|\boldsymbol{x}) - cs_0(\cdot,\boldsymbol{w}|\boldsymbol{x})\right)\right],
\end{aligned}$$

with an optimal solution in the form of (29). Therefore, the linear transformation property also holds for the Weight-COS-DPO loss with the penalization term. □