Orthogonal Gradient Descent: Learning from Preferences with Minimal Forgetting

Yunjae Won [1] **Doyoung Kim** [1] **Geewook Kim** [1]

Abstract

Direct Preference Optimization (DPO) is a widely used method for aligning language models to human preferences in an offline, supervised manner. However, DPO often suffers from *alignment tax*, where the aligned model forgets critical knowledge from the initial supervised fine-tuned (SFT) policy. In this work, we empirically identify a direct relationship between alignment tax and reductions in the log-likelihood of winning samples. To address this issue, we propose **Orthogonal Gradient Descent** (OGD), a novel approach that aligns models to preferences with minimal forgetting. OGD employs projected gradient descent to increase the log-likelihood of winning samples while decreasing or maintaining the log-likelihood of rejected ones. Unlike DPO, OGD requires no additional hyper-parameters and can function without access to the original SFT policy. Experiments show that OGD is able to learn the preference reward comparable to DPO ($\beta = 0.1$), while reducing forgetting by up to **99.15%**. These findings highlight OGD's potential as a robust method for AI alignment, especially in critical domains like healthcare and legal systems, where preserving foundational knowledge is essential.

1. Introduction

Aligning language model (LM) to human preferences has been demonstrated to be effective in various applications (Stiennon et al., 2020; Ouyang et al., 2022). Direct Preference Optimization (Rafailov et al. 2023; DPO) has been widely used as a method of directly aligning LMs in an offline supervised fashion (Bai et al., 2023; Ivison et al., 2023; Tunstall et al., 2023).

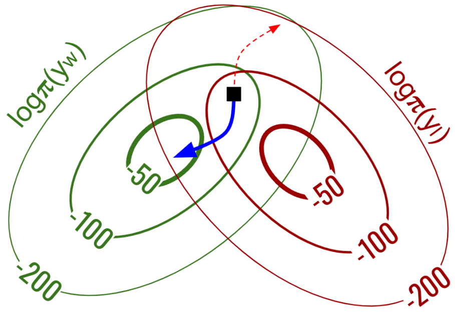

Figure 1. An illustration of the policy's trajectory in the parametric space during alignment training. While the DPO model follows the red path where the likelihood of both the preferred and dispreferred responses decrease, our proposed method follows the blue path which increases the preferred response likelihood while decreasing the dispreferred responses' likelihood. This mitigates forgetting of knowledge contained in the preferred responses.

DPO trains a LM to maximize the log-likelihood margin between the preferred and dispreferred[1] responses (Rafailov et al., 2024b). Intuitively, one may expect that DPO increases the log-likelihood of the preferred response $\mathbf{y}_w$ while decreasing the log-likelihood of the dispreferred response $\mathbf{y}_l$. However, various recent works have reported the *likelihood displacement* phenomena (Razin et al., 2024) in which DPO training causes the model to *decrease* the log-likelihood of $\mathbf{y}_w$, while increasing the log-likelihood margin (Pal et al., 2024; Rafailov et al., 2024a; Yuan et al., 2024b; Tajwar et al., 2024; Pang et al., 2024; Liu et al., 2024b). While there has been conflicting viewpoints regarding the favorableness of this phenomena, there lacks understanding in exactly what kind of real-life side-effects it entails when the log-likelihood of $\mathbf{y}_w$ decreases throughout the training process.

In this paper, we first empirically show that the decrease in the log-likelihood of $\mathbf{y}_w$ directly correlates to alignment tax. We measure the aligned policy's accuracy degradation in various benchmarks, with respect to its initial SFT policy under different training settings. We observe a clear trend between the log-likelihood of $\mathbf{y}_w$ and the degree of overall

[1] KAIST AI, Seoul, Republic of Korea. Correspondence to: Yunjae Won <yunjae.won@kaist.ac.kr>, Doyoung Kim <doyoungkim@kaist.ac.kr>, Geewook Kim <geewook@kaist.ac.kr>.

Project for KAIST AI611 *Deep Reinforcement Learning* Fall 2024

[1] We use the term 'winning', 'chosen', and 'preferred' interchangably.

performance degradation. Next, we present **Orthogonal Gradient Descent** (OGD), a projected gradient descent method that trains a model to increase the log-likelihood of $\mathbf{y}_w$, while maintaining or decreasing the log-likelihood of $\mathbf{y}_l$. We demonstrate the effectiveness of OGD on two preference datasets (Xu et al., 2024b) using `Mistral-7B-v0.3`. Experimental results suggest that OGD can align a LM to human preferences comparable to DPO, while significantly reducing forgetting by up to **99.15%**.

Our contribution is as follows:

- We demonstrate that the decrease in the log-likelihood of $\mathbf{y}_w$ directly correlates to an increased alignment tax (Forgetting).
- We present Orthogonal Gradient Descent (OGD), which increases the winning responses' log-likelihood while decreasing or maintaining the rejected responses' log-likelihood.
- OGD does not introduce any new hyper-parameters and does not require the reference policy. OGD also does not require the strict pairing between the chosen and rejected responses.
- We demonstrate that OGD can align a LM to human preferences comparable to DPO, while exhibiting significantly less alignment tax.

2. Related Work

2.1. Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) has become a popular method for aligning LMs to human preferences. In general, RLHF's training process consists of three phases: Supervised Fine-tuning (SFT), reward model (RM) training, and policy optimization to maximize the KL-regularized reward, typically using PPO (Schulman et al., 2017). Despite its success in various domains (Chaudhari et al., 2024; Wang et al., 2024b; Bai et al., 2022), the requirement of RM training complicates the training process, with the addition of careful hyper-parameter training and increased computational costs.

2.2. Direct Preference Optimization

Direct Preference Optimization (DPO) presents a simple, yet effective alternative to RLHF. By identifying a mapping between a language model and an implicit reward model, DPO proposes to directly minimize the KL-divergence between the empirical preference distribution and the implicit RM represented by the policy. This simplifies the overall RLHF process by eliminating the RM training step, and DPO has been widely adopted as a method for directly training a LM to learn human preferences (Bai et al., 2023; Ivison et al., 2023; Tunstall et al., 2023).

With increasing attempts to apply DPO in various domains, various recent works have also reported the *likelihood-displacement* phenomena (Razin et al., 2024) in which the policy decreases the log-likelihood of $\mathbf{y}_w$ throughout the training process. This inspired research on understanding the underlying cause of such phenomena (Razin et al., 2024; Tajwar et al., 2024; Pal et al., 2024; Rafailov et al., 2024a; Feng et al., 2024; Yuan et al., 2024a), where notably Razin et al. identified the gradient condition for which likelihood-displacement occurs.

The research community has seen conflicting perspectives on how this phenomena should be addressed; One line of research (Rafailov et al., 2024a; Shi et al., 2024) argued that such phenomena are a natural consequence of DPO, while another line of research (Pal et al., 2024; Feng et al., 2024; Yuan et al., 2024a) viewed it as a failure mode and attempted to mitigate it. Most notably, several recent works (Pal et al., 2024; Yuan et al., 2024b; Pang et al., 2024; Meng et al., 2024) have observed the phenomena where DPO training leads to degradation in math and reasoning benchmarks, for which Razin et al. has viewed it as an instance of the catastrophic likelihood displacement phenomena. Other works (Razin et al., 2024; Yuan et al., 2024a) have also argued that likelihood displacement itself can hinder the process of properly learning the preference distribution. However, such works were often validated in simplified and synthetic preference datasets. Our work expands upon this view by investigating the relationship between likelihood displacement and forgetting in real-world resembling complex instruction preference datasets (Cui et al., 2023; Xu et al., 2024b).

2.3. Alignment Tax

Efforts to reduce forgetting during alignment of LMs has been an active field of research. Often recognized as the *alignment tax* phenomena (Ouyang et al., 2022), there has been numerous reports where aligning LMs to human preferences lead to degraded performance on other downstream tasks (Pal et al., 2024; Yuan et al., 2024b; Pang et al., 2024; Meng et al., 2024). This is crucial for real-life applications of LMs in domains where forgetting pre-trained knowledge can lead to critical consequences, such as medical domains.

While previous works have studied ways to reduce alignment tax, several short-comings has remained. First, works either relied on the assumption that pre-training data is available (Ouyang et al., 2022), or relied on resource intensive methods, such as weight merging where one needs to train multiple models and carefully adjust the merging hyper-parameters (Lu et al., 2024; Lin et al., 2023; Fu et al., 2024). Our work aims to address the alignment tax problem in

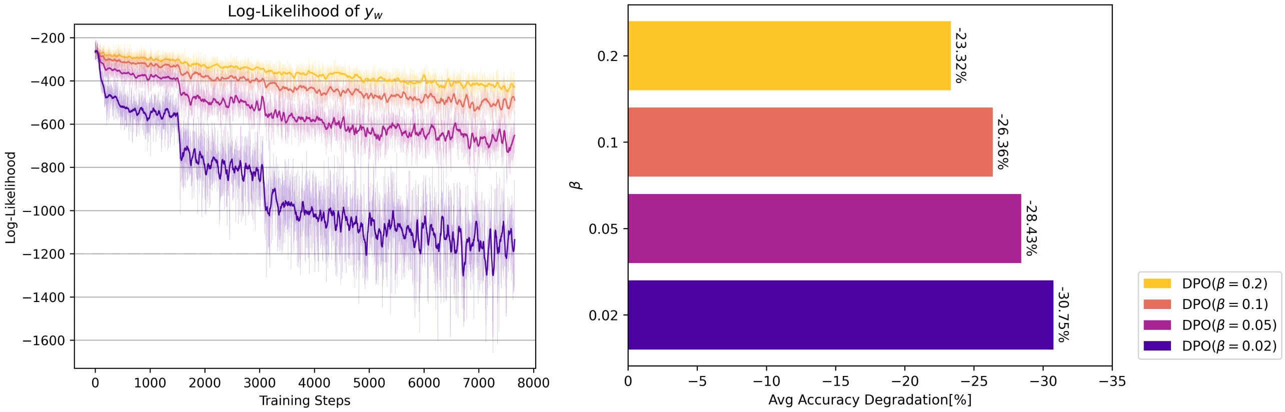

Figure 2. Comparison between the decrease in log-likelihood of $\mathbf{y}_w$ with the overall accuracy degradation in various downstream tasks (TAX$(\cdot)$). We train `Mistral-7B-v0.3` on the `Magpie-Pro` preference dataset using DPO with $\beta \in \{0.2, 0.1, 0.05, 0.02\}$ for 5 epochs. The model with the most decreased log-likelihood of $\mathbf{y}_w$ exhibits the most severe degree of forgetting.

a model-agnostic manner, without relying on any weight-merging based methods or data augmentation techniques.

2.4. Continual Learning

Our work proposes Orthogonal Gradient Descent (OGD) for addressing alignment tax. A methodology proposed under the same name in (Farajtabar et al., 2020) attempts to mitigate catastrophic forgetting by updating the model's parameters in the direction of not increasing the loss of the previous task. OGD can be understood as a projected gradient descent based method (Rosen, 1960) that mitigates forgetting during preference optimization of language models. In particular, we realize that the conventional two-step SFT training and preference optimization pipeline can be viewed as a continual learning setup. As such, we aim to mitigate forgetting of the SFT model's knowledge by forcing the preference optimization step to decrease the negative log-likelihood loss of $\mathbf{y}_w$. Our method differs from (Farajtabar et al., 2020) by utilizing the mini-batch gradients on the fly instead of using instance-wise gradients, due to the intensive memory requirements of training LMs. Most notably, instead of treating the constraint as a regularization method for alleviating forgetting as in conventional continual learning works (Li & Hoiem, 2017; Kirkpatrick et al., 2017; Farajtabar et al., 2019; Saha et al., 2021; Lin et al., 2022), we flip the constraint's role to *increase* the negative log-likelihood loss of $\mathbf{y}_l$. We show that this allows policies to provably optimize a large class of direct alignment loss functions (Gheshlaghi Azar et al., 2023), even without introducing new hyper-parameters or utilizing a reference SFT policy.

3. Preliminaries

In this section, we investigate the relation between likelihood-displacement and alignment tax. To measure alignment tax, we utilize the metric TAX$(\cdot)$ which computes the average percentage drop in accuracy over several benchmarks with respect to its initial SFT policy. For an aligned policy θ and its initial policy θ_{SFT}, we measure its degree of performance degradation for some task T_i by $100 \times \min(0, \frac{T_i(\theta) - T_i(\theta_{\text{SFT}})}{T_i(\theta_{\text{SFT}})})[\%]$, where $T_i(\theta)$ denotes the accuracy of LM θ on task T_i. For a collection of tasks $\mathcal{T} = \{T_1, T_2, ...\}$ we measure the average degradation to compute $\text{TAX}(\theta) = \frac{100}{|\mathcal{T}|}\Sigma_{T_i \in \mathcal{T}} \min(0, \frac{T_i(\theta) - T_i(\theta_{\text{SFT}})}{T_i(\theta_{\text{SFT}})})[\%]$. Its upper-bound is 0%, indicating no alignment tax has occurred, and has a lower bound of -100%, where the aligned policy has achieved 0% accuracy across all tasks. We compute TAX$(\cdot)$ in a zero-shot manner for a total of 8 multi-choice QA benchmarks: **(A) Commonsense QA:** PIQA (Bisk et al., 2020), SIQA (Sap et al., 2019), HellaSwag (Zellers et al., 2019), ARC-Easy and ARC-Challenge (Clark et al., 2018), **(B) Knowledge:** MMLU (Hendrycks et al., 2020), **(C) Math:** GSM8k (Cobbe et al., 2021), and **(D) Reading Comprehension:** BoolQ (Clark et al., 2019).

We train `Mistral-7B-v0.3` (Albert Jiang, 2024) on the `Magpie-Pro`[2] (Xu et al., 2024b; Wang et al., 2024a) preference dataset using the DPO objective with various $\beta \in \{0.2, 0.1, 0.05, 0.02\}$ values for 5 epochs. See appendix for further training details. We plot the log-likelihood of $\mathbf{y}_w$ and TAX$(\cdot)$ at Figure 2. We are able to observe an overall trend between the log-likelihood of $\mathbf{y}_w$ and TAX$(\cdot)$. In particular, the more the log-likelihood of $\mathbf{y}_w$ decreases, the more the TAX$(\cdot)$ metric drops. We hypothesize that the decrease in

[2] Dataset available at Magpie-Align/Magpie-Llama-3.1-Pro-DPO-100K-v0.1

the log-likelihood of $\mathbf{y}_w$ encourages the model to forget the factual knowledge contained in $\mathbf{y}_w$.

4. Method

The findings of section 3 motivate a method for aligning language models without decreasing the log-likelihood of $\mathbf{y}_w$. In this section, we present Orthogonal Gradient Descent (OGD), a projected gradient descent based method that can optimize the DPO loss without decreasing the log-likelihood of $\mathbf{y}_w$.

Orthogonal Gradient Descent (OGD) is based on the following simple intuition: If decreasing the negative log-likelihood (NLL) loss of $\mathbf{y}_w$ increases the NLL loss of $\mathbf{y}_l$, we can simply perform gradient descent on the NLL loss of $\mathbf{y}_w$. On the other hand, if decreasing the NLL loss of $\mathbf{y}_w$ decreases the NLL loss of $\mathbf{y}_l$ too, we update the parameters in such a way that we decrease only the NLL loss of $\mathbf{y}_w$, while maintaining the NLL loss of $\mathbf{y}_l$. In particular, we update the model parameters in the direction of the orthogonal projection of the preferred responses' gradient against the dispreferred responses' gradient. This direction is acute to the winning samples' gradient, forming a descent direction for the winning samples' NLL loss. Meanwhile, this update direction is orthogonal to the losing samples gradient, and thus ensures that the losing samples' NLL loss doesn't change, in principle.

Consider a batch of M training samples $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^M$ where $y_w^{(i)}, y_l^{(i)}$ are the preferred and dispreferred completions on the prompt $x^{(i)}$. Assuming standard NLL loss $L(\cdot)$, we denote the gradient of winning samples $\nabla L(y_w) = \frac{1}{M}\Sigma_{i=1}^M \nabla_\theta L(y_w^{(i)}|x^{(i)})$ and the losing samples $\nabla L(y_l) = \frac{1}{M}\Sigma_{i=1}^M \nabla_\theta L(y_l^{(i)}|x^{(i)})$. The first case, in which decreasing the NLL loss of winning samples increases the NLL loss of losing samples, corresponds to the gradient condition: $\nabla L(y_w)\cdot\nabla L(y_l) < 0$. In such cases we simply update θ in the direction of $-\nabla L(y_w)$. Otherwise, if we have $\nabla L(y_w)\cdot\nabla L(y_l) \geq 0$, we update θ in the direction of the orthogonal projection $-(\nabla L(y_w) - \frac{\nabla L(y_w)\cdot\nabla L(y_l)}{||\nabla L(y_l)||_2^2}\nabla L(y_l))$.

This leads to the following parameter update rule of OGD.

Definition 4.1. OGD: $\theta_{k+1} = \theta_k - \eta(\nabla L(y_w) - \frac{\alpha}{||\nabla L(y_l)||_2^2}\nabla L(y_l))$, where θ_k denotes the policy at training step k, $\eta > 0$ denotes the learning rate, and $\alpha = \max(0, \nabla L(y_w)\cdot\nabla L(y_l))$.

In practice, when using an optimizer (e.g., Adam (Kingma & Ba, 2015), RMSprop (Tieleman & Hinton, 2012), etc.), we set the parameters' gradient as $\nabla L(y_w) - \frac{\alpha}{||\nabla L(y_l)||_2^2}\nabla L(y_l)$ and update its parameters following the optimizer's algorithm. Similarly, when applying gradient-clipping, we clip the L2 norm of $\nabla L(y_w) - \frac{\alpha}{||\nabla L(y_l)||_2^2}\nabla L(y_l)$.

5. Theoretical Analysis

In this section, we analyze the theoretical properties of OGD. Without any introduction of new hyper-parameters, OGD *provably* increases the log-likelihood of $\mathbf{y}_w$, while *provably* decreasing or maintaining the log-likelihood of $\mathbf{y}_l$. This further allows the model to learn human preferences represented by a general family of offline preference optimization methods (Tang et al., 2024).

Definition 5.1. For some function $f : \mathbb{R}^D \to \mathbb{R}$, and a point $\theta \in \mathbb{R}^D$, a direction $\Delta\theta \in \mathbb{R}^D$ is called a descent direction if there exists $\bar{\alpha} > 0$ such that $f(\theta + \alpha\Delta\theta) < f(\theta), \forall\alpha \in (0, \bar{\alpha})$.

The following well-known lemma allows one to easily verify whether a direction is a descent direction by computing its dot product with the gradient of the function.

Lemma 5.2. *Consider a point $\theta \in \mathbb{R}^D$. Any direction $\Delta\theta \in \mathbb{R}^D$ satisfying $\Delta\theta \cdot \nabla f(\theta) < 0$ is a descent direction.*

Many existing offline preference optimization methods can be characterized by solving the following objective (Tang et al., 2024):

$$\arg\min_\theta \mathbb{E}_{(y_w,y_l)\sim\mu}[f(\beta\cdot(\log\frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \log\frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}))]$$

where f denotes any valid supervised binary classification loss function (Hastie, 2009).

We now analyze the properties of the update direction of OGD: $\Delta\theta = \theta_{k+1} - \theta_k = -\eta\{\nabla L(y_w) - \frac{\max(0, \nabla L(y_w)\cdot\nabla L(y_l))}{||\nabla L(y_l)||_2^2}\nabla L(y_l)\}$. The following theorem states that OGD increases the log-likelihood of $\mathbf{y}_w$.

Theorem 5.3. *$\Delta\theta$ is a descent direction of the negative log-likelihood of the preferred responses $\frac{1}{M}\Sigma_{i=1}^M L(y_w^{(i)}|x^{(i)})$*

Proof. See A.1.1. In a nutshell, regardless of the sign value of $\nabla L(y_w)\cdot\nabla L(y_l)$, we can show that $\Delta\theta\cdot\nabla L(y_w) < 0$. □

Conversely, we can show that OGD decreases or maintains the log-likelihood of $\mathbf{y}_l$.

Theorem 5.4. *$\Delta\theta$ is **not** a descent direction of the negative log-likelihood of the dispreferred responses: $\frac{1}{M}\Sigma_{i=1}^M L(y_l^{(i)}|x^{(i)})$*

Proof. $\Delta\theta \cdot \nabla L(y_l) = -\eta\{\nabla L(y_w)\cdot\nabla L(y_l) - \max(0, \nabla L(y_w)\cdot\nabla L(y_l))\} \geq 0$ In other words, $\Delta\theta$ is either an ascent direction or orthogonal to the log-likelihood of the dispreferred responses $\mathbf{y}_l$. □

As a consequence of 5.3 and 5.4, OGD is able to train a policy to learn the KL-regularized Bradley-Terry preference reward by optimizing various direct preference optimization objectives.

Corollary 5.5. *For any valid supervised binary classification loss function f with $f'(\cdot) < 0$, $\Delta\theta$ is a descent direction to the loss $f(\beta \cdot (\log \frac{\pi_\theta(y_w)}{\pi_{ref}(y_w)} - \log \frac{\pi_\theta(y_l)}{\pi_{ref}(y_l)}))$ where $\beta > 0$.*

Proof. See A.1.2. □

Meanwhile, we note that OGD can be considered as a hyper-parameter free variant of the unlikelihood training method (Welleck et al., 2019). In particular, OGD can be seen as dynamically adjusting the unlikelihood coefficient according to the arrangement of the log-likelihood gradients. Under this perspective, OGD can be understood as performing some form of knowledge unlearning (Jang et al., 2023), where the policy unlearns the knowledge required to generate $\mathbf{y}_l$ while retaining the knowledge contained in $\mathbf{y}_w$.

It is worth mentioning the implicit assumptions that the aforementioned theoretical properties rely on. First, Lemma 5.2 assumes that the first-order Taylor's approximation holds. Thus, taking a step size that violates this condition may not guarantee Theorem 5.4 and 5.3. In addition, both theorems assume the usage of full-batch gradients, which is not the case in conventional training setups. Utilizing mini-batch gradients may not ensure the proper optimization of the log-likelihoods of $\mathbf{y}_w$ and $\mathbf{y}_l$ outside the mini-batch. Finally, the usage of modern optimizers (Kingma & Ba, 2015; Tieleman & Hinton, 2012) that modify the parameter update direction may not guarantee the statements of Theorem 5.4 and 5.3. However, we demonstrate in Section 6 that OGD can successfully optimize the respective log-likelihoods and the DPO loss in realistic training settings, using stochastic mini-batch gradients and the RMSprop optimizer (Tieleman & Hinton, 2012).

Overall, OGD is a method which can provably optimize a general class of supervised direct alignment loss functions, without any new additional hyper-parameters. In addition, OGD does not require the reference SFT policy, similar to (Hong et al., 2024; Meng et al., 2024; Zhao et al., 2023b). As with (Ethayarajh et al., 2024; Mao et al., 2024), our method doesn't require the strict pairing between the preferred and dispreferred responses. Most importantly, OGD fits well with previous theoretical frameworks for learning prefernces, as OGD can provably optimize a wide class of direct preference optimization loss objectives (Tang et al., 2024).

6. Experiments

We empirically demonstrate that OGD is very effective in mitigating alignment tax, by simply changing the parameter update direction without any careful data-augmentation or complex weight merging techniques. In particular, we show that OGD is able to learn preferences comparable to that of DPO, while exhibiting significantly less alignment tax.

We mainly utilize the DPO loss with varying KL-regularization strengths $\beta \in \{0.2, 0.1, 0.05, 0.02\}$ as the baseline methods. Despite works exploring variants of DPO (Liu et al., 2024a; Hong et al., 2024; Ethayarajh et al., 2024; Meng et al., 2024; Xu et al., 2024a; Gheshlaghi Azar et al., 2023), the original DPO (Rafailov et al., 2024b) objective remains the standard training objective for training LMs to directly learn preferences (Dubey et al., 2024; Jiang et al., 2024; Tunstall et al., 2023). In addition, it has been suggested in (Tang et al., 2024) that varying the β KL-regularization strength is more influential to the final performance than changing the training objective (e.g., IPO, cDPO, ORPO, etc.). Due to non-negligible training and evaluation costs, we mainly consider varying the KL-regularization strength and early-stopping as the baseline methods.

We evaluate our method on two high-quality preference datasets (`Magpie-Pro` and `Magpie-Air-Gemma27B`[3]) using an open-source language model `Mistral-7B-v0.3`. To measure the preference reward learnability, we utilize two *LLM-As-a-Judge* benchmarks (Lin et al., 2024; Tianle Li*, 2024). In particular, we define that model `A` is able to learn preferences comparable to model `B` when `A` either outperforms `B` or has an overlap between its 95% confidence intervals. To compare the degree of forgetting, we utilize the TAX$(\cdot)$ metric using the same multi-choice QA benchmarks in Section 3. See Appendix A.2 for specific experimental details.

We summarize our results on Table 5. OGD consistently learns the preference reward comparable to the best performing DPO models, while paying significantly less alignment tax. In particular, the 95% confidence interval on the WildBench benchmark overlaps between OGD and the best performing DPO trained models. Meanwhile, OGD alleviates forgetting, up to 99.15% of that of DPO ($\beta = 0.1$) on the `Magpie-Pro` preference dataset.

We track the change in TAX$(\cdot)$ at each epoch for all training configurations in Figure 3. OGD consistently exhibits significantly less forgetting compared to DPO trained models. At the last fifth epoch, OGD consistently ranks as the best model in alleviating forgetting. On average, OGD is able to

[3] Dataset available at https://huggingface.co/datasets/yjwon/Magpie-Air-Gemma2-DPO-100K

Table 1. Overall evaluation results for Section 6. We specify the 95% confidence interval inside the parenthesis for the Arena-Hard and Wild-Bench benchmark performances. Overall, OGD is able to learn the preference reward comparable to the best DPO trained models, while paying significantly less alignment tax.

Method	**Best Epoch**	**Arena-Hard SC Win-Rate**	**Wild-Bench v0.2 ELO Score**	TAX$(\cdot)$
		`Mistral-7B-v0.3` on the `Magpie-Air-Gemma27B` dataset		
DPO ($\beta = 0.2$)	3	30.2 (27.82, 32.3)	1134.79 (1121.15, 1146.72)	-12.7355%
DPO ($\beta = 0.1$)	4	30.5 (28.44, 32.67)	1136.45 (1124.85, 1147.87)	-19.4960%
DPO ($\beta = 0.05$)	1	32.6 (31.13, 35.11)	1142.00 (1127.45, 1153.95)	-12.5871%
DPO ($\beta = 0.02$)	1	30.9 (29.1, 32.74)	1132.48 (1119.39, 1149.92)	-19.1050%
OGD (*Ours*)	5	27.1 (24.56, 29.18)	1122.92 (1111.20, 1135.67)	**-2.6340%**
		`Mistral-7B-v0.3` on the `Magpie-Pro` dataset		
DPO ($\beta = 0.2$)	1	23.8 (22.02, 25.55)	1142.27 (1131.90, 1152.78)	-14.6284%
DPO ($\beta = 0.1$)	1	26.8 (24.98, 28.81)	1138.75 (1127.15, 1151.52)	-19.2612%
DPO ($\beta = 0.05$)	1	27.4 (25.49, 29.56)	1141.01 (1128.93, 1151.62)	-22.6019%
DPO ($\beta = 0.02$)	1	27.2 (24.9, 28.96)	1138.17 (1122.19, 1152.72)	-23.3374%
OGD (*Ours*)	5	26 (24.03, 28.26)	1124.37 (1108.08, 1137.55)	**-0.1638%**

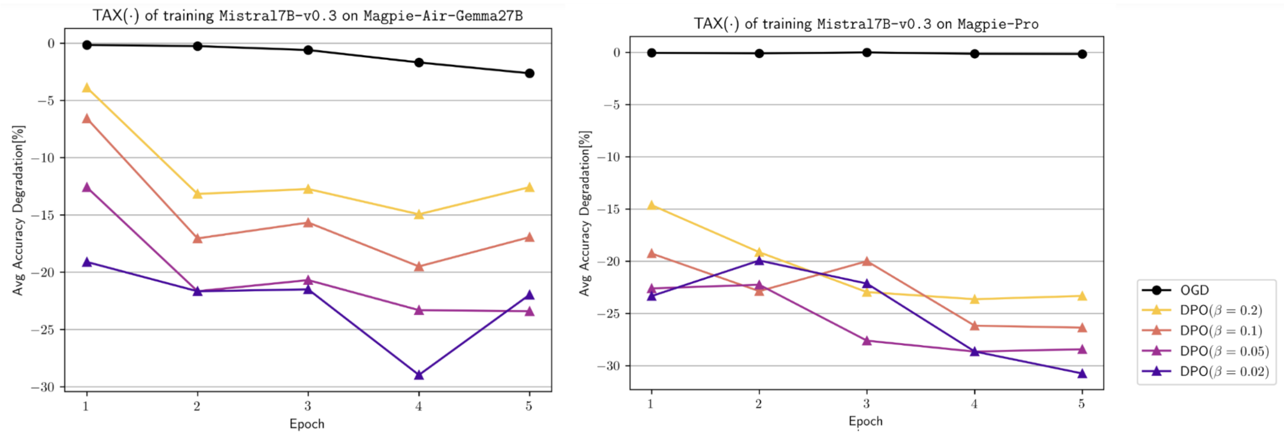

Figure 3. Change in TAX$(\cdot)$ at each epoch. A point closer to 0% indicates less forgetting has occurred. Overall, OGD consistently exhibits significantly less forgetting compared to other baseline methods.

reduce forgetting by **92.82%** of that of DPO ($\beta = 0.1$) at epoch 5.

7. Discussion

7.1. Why is OGD effective at alleviating forgetting?

OGD trained models consistently exhibit significantly less forgetting compared to DPO trained models. Comparing Figure 4 and 5, we find that the primary difference between DPO and OGD is the change in the log-likelihood of $\mathbf{y}_w$. Consequently, we hypothesize that the increase of the log-likelihood of $\mathbf{y}_w$ allows the model to retain the knowledge contained in the $\mathbf{y}_w$. This aligns with the observation made in (Shi et al., 2024) in which ”higher likelihood correlates with better memorisation of factual knowledge patterns”. As acknowledged in Section 2.4, OGD can be viewed as a continual learning method that aims to retain the knowledge seen during the previous SFT phase. Therefore, as long as $\mathbf{y}_w$ offers sufficient coverage of factual knowledge, we hypothesize that OGD can show strong performance in alleviating forgetting.

7.2. Does OGD work in realistic training settings?

The properties of OGD mentioned in Section 5 can be summarized as the following:

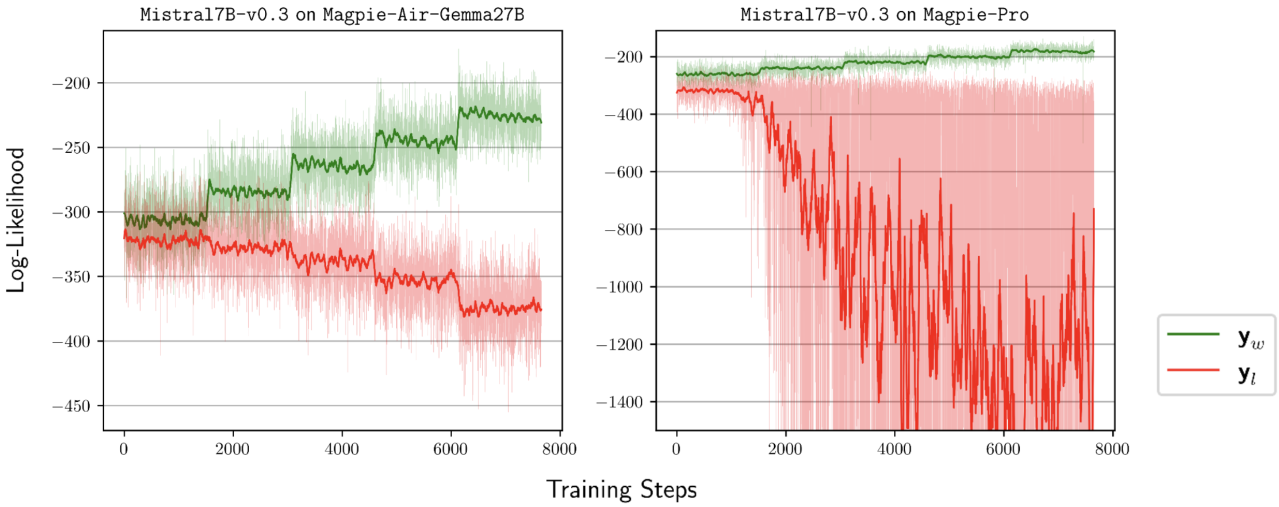

Figure 4. Log-likelihood change for OGD across all experimental configurations. Overall, OGD consistently increases the log-likelihood of $\mathbf{y}_w$, while decreasing or maintaining the log-likelihood of $\mathbf{y}_l$.

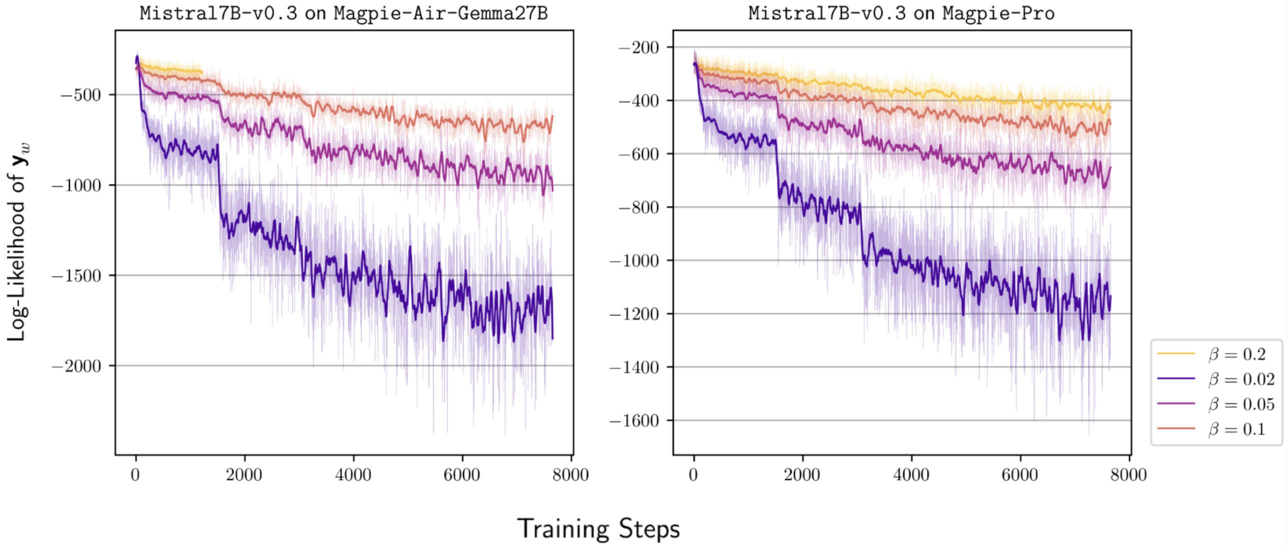

Figure 5. Log-likelihood change of $\mathbf{y}_w$ for DPO across all experimental configurations.

- OGD increases the log-likelihood of $\mathbf{y}_w$.
- OGD either decreases or maintains the log-likelihood of $\mathbf{y}_l$.
- OGD can optimize a wide class of offline preference optimization objectives, including the DPO loss.

However, as noted in Section 5, these are not guaranteed when utilizing stochastic mini-batch gradients and non-SGD optimizers (e.g., Adam, RMSprop, etc.) We demonstrate that OGD can satisfy these properties even in realistic training setting. The experiments in Section 6 were conducted using batch size of 64 and RMSprop optimizer with learning rate 1e-6. Under such settings, Figure 4 shows that OGD can optimize the respective log-likelihoods for all experiments. Furthermore, Figure 6 demonstrates that OGD can successfully optimize the DPO loss with $\beta = 0.1$. This suggests that the properties of OGD listed above holds in realistic training settings, too.

8. Limitations and Future Works

Despite the strong empirical performance of OGD and favorable theoretical properties, there remains some limitations of OGD that may be addressed in future works. First, OGD requires an additional memory requirement proportional to

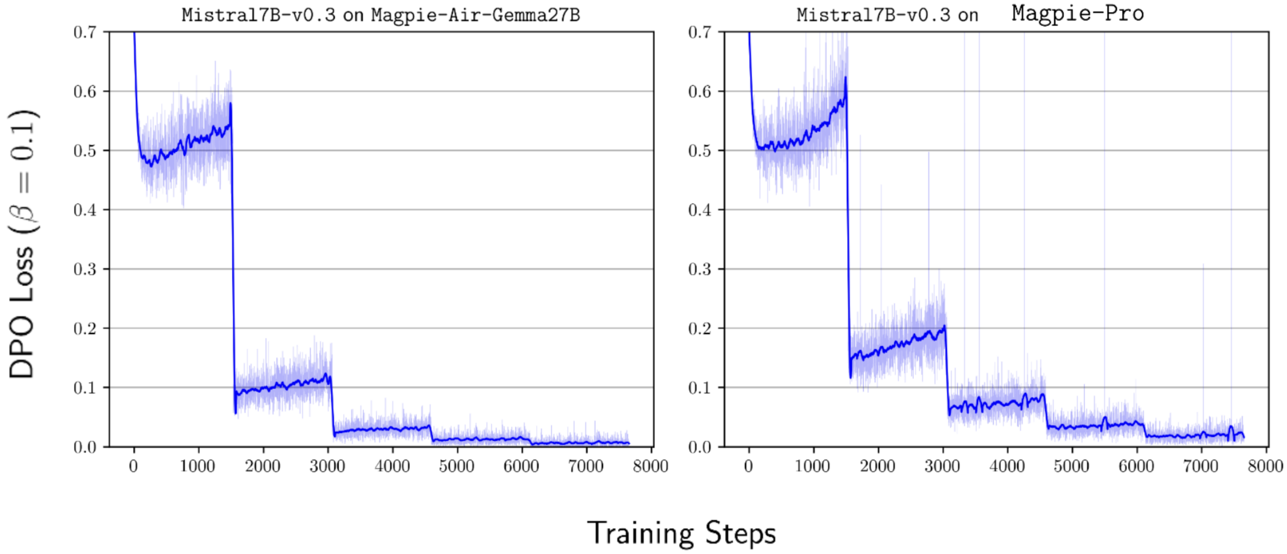

Figure 6. DPO loss for OGD across all experimental configurations. The DPO loss is measured using $\beta = 0.1$. Overall, OGD is able to optimize the DPO loss regardless of the dataset.

the size of the model's parameters. This is due to the method requiring the separate storage of $\nabla L(y_w)$ and $\nabla L(y_l)$. Future works could explore combining parameter efficient training methods (Hu et al., 2021; Dettmers et al., 2023; Chavan et al., 2023) with OGD. However, it should also be acknowledged that OGD doesn't require the storage and forward pass of the initial SFT policy, which is not the case for DPO. Second, OGD relies on the assumption that the $\mathbf{y}_w$ contains sufficiently favorable responses. Since OGD is designed to increase the log-likelihood of $\mathbf{y}_w$, if $\mathbf{y}_w$ contains suboptimal completions, the model may fail to learn an optimal policy. Likewise, if $\mathbf{y}_w$ fails to offer a sufficient coverage of factual knowledge, OGD may overfit the model on $\mathbf{y}_w$, forgetting knowledge not covered by $\mathbf{y}_w$. However, it should also be noted that OGD can easily incorporate mixing different training data, including paired preference datasets and instruction tuning datasets without dispreferred responses. This is because OGD does not assume the strict pairing between the preferred and dispreferred responses. Thus, future works can explore methods of mixing different data to tackle overfitting on $\mathbf{y}_w$.

9. Conclusion

Forgetting pre-trained knowledge during AI alignment can be critical in various knowledge-intensive domains (e.g., medical domains). In this paper, we have investigated the relationship between the log-likelihood of preferred responses and the degree of alignment tax (forgetting). Preliminary experiments discussed in Section 3 suggest that decreasing the log-likelihood of preferred responses leads to in-domain forgetting, eventually leading to degraded downstream task performances. We present a novel preference optimization method, Orthogonal Gradient Descent (OGD), that increases the log-likelihood of preferred responses, while decreasing or maintaining the log-likelihood of dispreferred responses. OGD enjoys several favorable theoretical properties, most notably where it can also optimize a wide class of offline preference optimization losses, including the DPO objective. We demonstrate the effectiveness of OGD on two complex preference datasets. Experimental results suggest that OGD is able to learn the preference reward comparable to the best performing DPO models, while exhibiting significantly less forgetting. We hope that this work paves a new way for effectively alleviating forgetting during AI alignment.

References

Albert Jiang, Alexandre Sablayrolles, A. T. A. R. A. M. A. H.-S. B. B. B. d. M. B. S. B. C. F. D. S. C. D. d. l. C. E. A. E. B. H. E. M. G. L. G. B. G. L. H. R. J.-M. D. J. L. J. M. L. M. L. T. L. S. L. R. L. M. J. M. P. M. T. M.-A. L. N. S. P. v. P. P. S. S. S. S. Y. S. A. T. L. S. T. L. T. L. T. G. T. W. V. N. W. E. S. W. M. mistralai/mistral-7b-v0.3, 2024. URL https://huggingface.co/mistralai/Mistral-7B-v0.3.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., and Zhu, T. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

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

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Chaudhari, S., Aggarwal, P., Murahari, V., Rajpurohit, T., Kalyan, A., Narasimhan, K., Deshpande, A., and da Silva, B. C. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *arXiv preprint arXiv:2404.08555*, 2024.

Chavan, A., Liu, Z., Gupta, D., Xing, E., and Shen, Z. One-for-all: Generalized lora for parameter-efficient fine-tuning, 2023. URL https://arxiv.org/abs/2306.07967.

Clark, C., Lee, K., Chang, M., Kwiatkowski, T., Collins, M., and Toutanova, K. Boolq: Exploring the surprising difficulty of natural yes/no questions. *CoRR*, abs/1905.10044, 2019. URL http://arxiv.org/abs/1905.10044.

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

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

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

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. Qlora: Efficient finetuning of quantized llms, 2023. URL https://arxiv.org/abs/2305.14314.

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

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning, 2019. URL https://arxiv.org/abs/1910.07104.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Feng, D., Qin, B., Huang, C., Zhang, Z., and Lei, W. Towards analyzing and understanding the limitations of dpo: A theoretical perspective. *arXiv preprint arXiv:2404.04626*, 2024.

Fu, T., Cai, D., Liu, L., Shi, S., and Yan, R. Disperse-then-merge: Pushing the limits of instruction tuning via alignment tax reduction. *arXiv preprint arXiv:2405.13432*, 2024.

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

Gheshlaghi Azar, M., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv e-prints*, pp. arXiv–2310, 2023.

Hastie, T. The elements of statistical learning: data mining, inference, and prediction, 2009.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hong, J., Lee, N., and Thorne, J. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*, 2024.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Ivison, H., Wang, Y., Pyatkin, V., Lambert, N., Peters, M., Dasigi, P., Jang, J., Wadden, D., Smith, N. A., Beltagy, I., and Hajishirzi, H. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023.

Jang, J., Yoon, D., Yang, S., Cha, S., Lee, M., Logeswaran, L., and Seo, M. Knowledge unlearning for mitigating privacy risks in language models. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805.

Jiang, A. Q., Sablayrolles, A., Roux, A., Mensch, A., Savary, B., Bamford, C., Chaplot, D. S., de las Casas, D., Hanna, E. B., Bressand, F., Lengyel, G., Bour, G., Lample, G., Lavaud, L. R., Saulnier, L., Lachaux, M.-A., Stock, P., Subramanian, S., Yang, S., Antoniak, S., Scao, T. L., Gervet, T., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Lin, B. Y., Deng, Y., Chandu, K., Brahman, F., Ravichander, A., Pyatkin, V., Dziri, N., Bras, R. L., and Choi, Y. Wildbench: Benchmarking llms with challenging tasks from real users in the wild, 2024. URL https://arxiv.org/abs/2406.04770.

Lin, S., Yang, L., Fan, D., and Zhang, J. Beyond not-forgetting: Continual learning with backward knowledge transfer. *Advances in Neural Information Processing Systems*, 35:16165–16177, 2022.

Lin, Y., Tan, L., Lin, H., Zheng, Z., Pi, R., Zhang, J., Diao, S., Wang, H., Zhao, H., Yao, Y., et al. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. *arXiv preprint arXiv:2309.06256*, 2023.

Liu, T., Qin, Z., Wu, J., Shen, J., Khalman, M., Joshi, R., Zhao, Y., Saleh, M., Baumgartner, S., Liu, J., et al. Lipo: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*, 2024a.

Liu, Z., Lu, M., Zhang, S., Liu, B., Guo, H., Yang, Y., Blanchet, J., and Wang, Z. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer. *arXiv preprint arXiv:2405.16436*, 2024b.

Lu, K., Yu, B., Huang, F., Fan, Y., Lin, R., and Zhou, C. Online merging optimizers for boosting rewards and mitigating tax in alignment. *arXiv preprint arXiv:2405.17931*, 2024.

Mao, X., Li, F.-L., Xu, H., Zhang, W., Chen, W., and Luu, A. T. As simple as fine-tuning: Llm alignment via bidirectional negative feedback loss, 2024. URL https://arxiv.org/abs/2410.04834.

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

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

Pal, A., Karkhanis, D., Dooley, S., Roberts, M., Naidu, S., and White, C. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

Pang, R. Y., Yuan, W., Cho, K., He, H., Sukhbaatar, S., and Weston, J. Iterative reasoning preference optimization. *arXiv preprint arXiv:2404.19733*, 2024.

Rafailov, R., Hejna, J., Park, R., and Finn, C. From r r to qˆ qˆ* : Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024a.

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

Razin, N., Malladi, S., Bhaskar, A., Chen, D., Arora, S., and Hanin, B. Unintentional unalignment: Likelihood displacement in direct preference optimization, 2024. URL https://arxiv.org/abs/2410.08847.

Rosen, J. B. The gradient projection method for nonlinear programming. part i. linear constraints. *Journal of the Society for Industrial and Applied Mathematics*, 8(1): 181–217, 1960. ISSN 03684245. URL http://www.jstor.org/stable/2098960.

Saha, G., Garg, I., and Roy, K. Gradient projection memory for continual learning, 2021. URL https://arxiv.org/abs/2103.09762.

Sap, M., Rashkin, H., Chen, D., LeBras, R., and Choi, Y. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

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

Shi, Z., Land, S., Locatelli, A., Geist, M., and Bartolo, M. Understanding likelihood over-optimisation in direct alignment algorithms, 2024. URL https://arxiv.org/abs/2410.11677.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021, 2020.

Tajwar, F., Singh, A., Sharma, A., Rafailov, R., Schneider, J., Xie, T., Ermon, S., Finn, C., and Kumar, A. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Tang, Y., Guo, Z. D., Zheng, Z., Calandriello, D., Munos, R., Rowland, M., Richemond, P. H., Valko, M., Ávila Pires, B., and Piot, B. Generalized preference optimization: A unified approach to offline alignment, 2024. URL https://arxiv.org/abs/2402.05749.

Tianle Li*, Wei-Lin Chiang*, E. F. L. D. B. Z. J. E. G. I. S. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024. URL https://lmsys.org/blog/2024-04-19-arena-hard/.

Tieleman, T. and Hinton, G. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4 (2):26–31, 2012.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Wang, H., Xiong, W., Xie, T., Zhao, H., and Zhang, T. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024a.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024b.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training, 2019. URL https://arxiv.org/abs/1908.04319.

Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Van Durme, B., Murray, K., and Kim, Y. J. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024a.

Xu, Z., Jiang, F., Niu, L., Deng, Y., Poovendran, R., Choi, Y., and Lin, B. Y. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *ArXiv*, abs/2406.08464, 2024b. URL https://api.semanticscholar.org/CorpusID:270391432.

Yuan, H., Zeng, Y., Wu, Y., Wang, H., Wang, M., and Leqi, L. A common pitfall of margin-based language model alignment: Gradient entanglement, 2024a. URL https://arxiv.org/abs/2410.13828.

Yuan, L., Cui, G., Wang, H., Ding, N., Wang, X., Deng, J., Shan, B., Chen, H., Xie, R., Lin, Y., et al. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*, 2024b.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. Hellaswag: Can a machine really finish your sentence? *CoRR*, abs/1905.07830, 2019. URL http://arxiv.org/abs/1905.07830.

Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.-C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., Desmaison, A., Balioglu, C., Damania, P., Nguyen, B., Chauhan, G., Hao, Y., Mathews, A., and Li, S. Pytorch fsdp: Experiences on scaling fully sharded data parallel, 2023a. URL https://arxiv.org/abs/2304.11277.

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

A. Appendix

A.1. Proofs for Section 5

A.1.1. Proof for Theorem 5.3

Proof. **Case 1:** If we have $\nabla L(y_w) \cdot \nabla L(y_l) > 0$, it follows that

$$\Delta\theta \cdot \nabla L(y_w) = -\eta\{||\nabla L(y_w)||_2^2 - \frac{\nabla L(y_w) \cdot \nabla L(y_l)}{||\nabla L(y_l)||_2^2}\nabla L(y_l) \cdot \nabla L(y_w)\}$$

$$= -\frac{\eta}{||\nabla L(y_l)||_2^2}\{||\nabla L(y_w)||_2^2 \cdot ||\nabla L(y_l)||_2^2 - (\nabla L(y_l) \cdot \nabla L(y_w))^2\} < 0$$

where the last inequality follows from the Cauchy-Schwarz inequality: $||\nabla L(y_w)|| \cdot ||\nabla L(y_l)|| > ||\nabla L(y_w) \cdot \nabla L(y_l)|| > 0$

Case 2: Otherwise, we have $\nabla L(y_w) \cdot \nabla L(y_l) \leq 0$ and it follows that

$$\Delta\theta \cdot \nabla L(y_w) = -\eta||\nabla L(y_w)||_2^2 < 0$$

.

□

A.1.2. Proof for Corollary 5.5

Proof.

$$\Delta\theta \cdot \nabla f(\beta \cdot (\log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}))$$

$$= \Delta\theta \cdot \{\beta f'(\beta(\log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}))(\nabla L(y_w) - \nabla L(y_l))\}$$

$$= \beta f'(\beta(\log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)}))(\Delta\theta \cdot \nabla L(y_w) - \Delta\theta \cdot \nabla L(y_l))$$

From Lemma 5.3, we have $\Delta\theta \cdot \nabla L(y_w) > 0$, and from Lemma 5.4, we have $\Delta\theta \cdot \nabla L(y_l) \leq 0$. Thus, we have $(\Delta\theta \cdot \nabla L(y_w) - \Delta\theta \cdot \nabla L(y_l)) > 0$. Since $\beta > 0$ and $\beta f'(\beta(\Delta\theta \cdot \nabla L(y_w) - \Delta\theta \cdot \nabla L(y_l))) < 0$, it follows that $\Delta\theta \cdot \nabla f(\beta \cdot (\log \frac{\pi_\theta(y_w)}{\pi_{\text{ref}}(y_w)} - \log \frac{\pi_\theta(y_l)}{\pi_{\text{ref}}(y_l)})) < 0$. □

A.2. Experimental Setting

In this section, we list the specific setting for the experiments conducted in Section 6. Unless further specified, the following hyper-parameters apply to all experiments.

We utilize the pre-trained base models instead of the instruction tuned versions to study the sole effects of the alignment method. We only utilize the chat template offered by the official instruction tuned corresponding models. For the initial SFT phase, we train the model for 1 epoch with effective batch size 256 using the Adam optimizer (Kingma & Ba, 2015) with default β_0, β_1 values and weight decay value of 0. We utilize a constant 5e-6 learning rate schedule with a linear warm-up for the first 10% training steps. The training objective consists of standard Cross Entropy Loss on the entire sequence including the prompt and special chat template tokens. During the subsequent preference learning phase, we train for 5 epochs with effective batch size 64 using the RMSprop optimizer (Tieleman & Hinton, 2012) without any weight decay. We employ a constant 1e-6 learning rate schedule with a linear warm-up of 150 steps. For all alignment methods, we compute the loss on the completions only. We fix the seed to 0 for the SFT stage and 1 for the preference learning stage. We save the model checkpoint every epoch, and all training is done in `bfloat16` precision. We fix the prompt token sequence length to 2,048 and entire token sequence length to 4,096. We train all models including OGD on A100/H100 gpus using Pytorch FSDP (Zhao et al., 2023a).

After training, we choose the best model based on the win-rate on the Arena-Hard benchmark using `gpt-4o-mini-2024-07-18` due to low evaluation costs. We report the final performance on **Arena-Hard** using `gpt-4-1106-preview` as recommended in (Tianle Li*, 2024). For **Wild-Bench v0.2**, we utilize `gpt-4o-2024-08-06` as recommended in the official github repository[4]. During evaluation, we greedy decode a response with maximum token length of 4,096 using `vLLM` (Kwon et al., 2023) as the inference engine. We evaluate TAX$(\cdot)$ using the official `lm-evaluation-harness` library (Gao et al., 2024), with a minor modification to prefix the system prompt message as the first sentence of the user query.

[4] https://github.com/allenai/WildBench