Human Alignment of Large Language Models through Online Preference Optimisation

Daniele Calandriello [*1] **Zhaohan Daniel Guo** [*1] **Remi Munos** [*1] **Mark Rowland** [*1] **Yunhao Tang** [1] **Bernardo Avila Pires** [1] **Pierre Harvey Richemond** [1] **Charline Le Lan** [1] **Michal Valko** [1] **Tianqi Liu** [1] **Rishabh Joshi** [1] **Zeyu Zheng** [1] **Bilal Piot** [*1]

Abstract

Ensuring alignment of language models' outputs with human preferences is critical to guarantee a useful, safe, and pleasant user experience. Thus, human alignment has been extensively studied recently and several methods such as Reinforcement Learning from Human Feedback (RLHF), Direct Preference Optimisation (DPO) and Sequence Likelihood Calibration (SLiC) have emerged. In this paper, our contribution is two-fold. First, we show the equivalence between two recent alignment methods, namely Identity Preference Optimisation (IPO) and Nash Mirror Descent (Nash-MD). Second, we introduce a generalisation of IPO, named IPO-MD, that leverages the regularised sampling approach proposed by Nash-MD. This equivalence may seem surprising at first sight, since IPO is an offline method whereas Nash-MD is an online method using a preference model. However, this equivalence can be proven when we consider the online version of IPO, that is when both generations are sampled by the online policy and annotated by a trained preference model. Optimising the IPO loss with such a stream of data becomes then equivalent to finding the Nash equilibrium of the preference model through self-play. Building on this equivalence, we introduce the IPO-MD algorithm that generates data with a mixture policy (between the online and reference policy) similarly as the general Nash-MD algorithm. We compare online-IPO and IPO-MD to different online versions of existing losses on preference data such as DPO and SLiC on a summarisation task.

[*]Equal contribution [1]Google DeepMind. Correspondence to: Daniele Calandriello <dcalandriello@google.com>, Zhaohan Daniel Guo <danielguo@google.com>, Michal Valko <michal.valko@inria.fr>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Learning from feedback is a common approach to align the behaviour of artificial agents with human preferences (Knox & Stone, 2008; Griffith et al., 2013; Christiano et al., 2017; Warnell et al., 2018). In recent years, reinforcement learning from human feedback has become a common paradigm for fine-tuning large language models (Glaese et al., 2022; OpenAI, 2022).

The traditional approach to fine-tuning of large language models from human preferences is to learn a reward signal under the Bradley-Terry (BT) model (Bradley & Terry, 1952), and then perform reinforcement learning (RL) against this learnt reward signal (Christiano et al., 2017). Recently, Rafailov et al. (2023) proposed an alternative model-free approach, direct preference optimisation (DPO). DPO is mathematically equivalent to the method above, in the sense that the minimiser of the population loss is identical (Azar et al., 2023), yet DPO bypasses the learning of the reward signal. Both approaches, however, rely on the BT model.

Recently, two particular approaches to directly optimise against preference probabilities themselves, rather than a Bradley-Terry-derived reward function have been proposed. Identity preference optimisation (Azar et al., 2023, IPO) is an algorithm that aims to optimise preference probabilities against a fixed data distribution, and does so with an offline contrastive loss, as with DPO. By contrast, Nash-MD-PG (Munos et al., 2023) is an algorithm that aims to find a Nash equilibrium with respect to the preference probabilities, via online policy gradient updates against a regularised policy. Both algorithms have appealing properties, though on the face of it are unrelated: one is an offline contrastive algorithm optimising against a fixed policy, the other is an online algorithm aiming to find a Nash equilibrium.

In this work we bridge the gap between IPO and Nash-MD-PG, and use this theoretical bridge to propose a novel class of preference optimisation algorithms. Specifically, our principal contributions are: first, we identify several key factors of variation between IPO and Nash-MD-PG, including their use of offline/online data, contrastivity of their

losses, and the nature of their equilibria. We use this understanding to identify the strengths of these algorithms, and combine these strengths into new preference optimisation algorithms. This allows us to propose *Online IPO*, an online variant of IPO. In addition, we establish a theoretical connection between Online IPO and *self-play* in the regularised two-player preference game used in deriving Nash-MD-PG. Second, motivated by Online IPO, we propose a preference optimisation algorithm, aiming to capture the best aspects of both IPO and Nash-MD-PG: *IPO-MD*, a version of IPO that interpolates between offline and online variants by using the lagged data distribution of Nash-MD-PG. Finally, we provide an experimental suite contrasting these algorithms in several applications, which provides detailed comparisons between the proposed methods and several baselines, with notable take-aways for practitioners.

2. Background

We begin by introducing the central preference optimisation problem, and relevant prior work.

2.1. Preference optimisation in bandits

We consider a (non-contextual) bandit problem (rather than a sequential setting), with finite action space $\mathcal{Y}$. This simplifies the notation considerably, and the ideas presented are straightforwardly extensible to the contextual/sequential setting where the actions/generations y are conditioned on a state/prompt x; we explain this in further detail in the descriptions of implementation details in the experiments.

Preferences. A *preference function* $p : \mathcal{Y} \times \mathcal{Y} \to [0, 1]$ specifies pairwise preference probabilities between elements of $\mathcal{Y}$. Specifically, given $y, y' \in \mathcal{Y}$, $p(y \succ y')$ is the probability that y is preferred to y'. We will assume that preference functions satisfy the symmetry condition $p(y' \succ y) = 1 - p(y \succ y')$.

At a high level, the goal of learning in this context is to find a policy in $\Delta(\mathcal{Y})$ (the set of distributions over $\mathcal{Y}$) that tends to select actions $y \in \mathcal{Y}$ that are preferred over others. This is of course not a precise mathematical objective as stated, and there are several distinct ways in which this can be formalised, which we explore in greater detail below.

Data model. Actions are sampled from policies $\mu, \mu' \in \Delta(\mathcal{Y})$ as $Y \sim \mu, Y' \sim \mu'$ (we write $Y, Y' \sim \mu$ when $\mu = \mu'$). Given a preference function p, the *preference distribution* λ_p is defined for $y, y' \in \mathcal{Y}$ as the distribution corresponding to the following sampling procedure:

$$\lambda_p(y, y') \text{ yields } \begin{cases} (y, y') \text{ with probability } p(y \succ y') \\ (y', y) \text{ with probability } 1 - p(y \succ y') \,. \end{cases}$$

We will frequently make use of a collection of samples drawn from the data-generating policies and the preference distribution, which we denote by $(y_i^+, y_i^-)_{i=1}^N$. In this case we will say the data is sampled from (μ, μ', λ_p) (or (μ, λ_p) when $\mu = \mu'$). In *offline* settings, the data is typically generated from a fixed policy μ, whereas in *online* settings, new data can be generated from a currently estimated policy π. Another important aspect of the data available to the learner is an initial policy $\pi^{\text{ref}} \in \Delta(\mathcal{Y})$. Typically, this policy provides acceptable behaviour in some (though not all) aspects of interest in the problem, and may be used to define a trust region for the policy optimisation problem. Our central contributions in this paper draw together, unify, and contrast several different approaches to preference optimisation; we now recall these below.

2.2. RLHF with a Bradley-Terry reward model

The approach to preference optimisation proposed by Christiano et al. (2017) is split into two steps. First, a reward model $r : \mathcal{Y} \to \mathbb{R}$ is fitted via the BT model (Bradley & Terry, 1952). In more detail, the preference probability $p(y \succ y')$ is approximated by the logistic function

$$\sigma(r(y) - r(y')) = \frac{e^{r(y)-r(y')}}{1 + e^{r(y)-r(y')}} \,,$$

and a reward function $\hat{r}$ is learnt essentially by performing maximum likelihood in this model, given a collection of observed preferences $(y_i^+, y_i^-)_{i=1}^N$ sampled from (μ, λ_p), (approximately) maximising the objective

$$\frac{1}{N} \sum_{i=1}^{N} \log \left(\sigma(r(y_i^+) - r(y_i^-)) \right) \,. \tag{1}$$

Policy optimisation then proceeds by aiming to maximise the expected reward $\hat{r}$ under the π, subject to a KL constraint against the initial policy:

$$\arg\max_{\pi \in \Delta(\mathcal{Y})} \left[\mathop{\mathbb{E}}_{Y \sim \pi} [\hat{r}(Y)] - \tau \text{KL}(\pi \,||\, \pi^{\text{ref}}) \right] , \tag{2}$$

with $\tau > 0$ a temperature parameter controlling the degree of regularisation towards π^{ref}. Note that the solution π^* to Equation (2) is available in closed form, as

$$\pi^*(y) \propto \pi^{\text{ref}}(y) \exp(\tau^{-1} \hat{r}(y)) \,. \tag{3}$$

2.3. Direct preference optimisation (DPO)

Rafailov et al. (2023) propose *direct preference optimisation* (DPO) as an alternative to RLHF as described above, noting that with the closed form in Equation (3), the learning of a reward function can be completely bypassed, instead reparametrising the optimal reward in terms of the optimal

policy, substituting into Equation (1), and aiming to maximise the resulting objective with respect to π:

$$\frac{1}{N}\sum_{i=1}^{N}\log\left(\sigma\left(\tau\ \log\left(\frac{\pi(y_i^+)\pi^{\text{ref}}(y_i^-)}{\pi(y_i^-)\pi^{\text{ref}}(y_i^+)}\right)\right)\right). \quad (4)$$

The derivation of DPO implies that, mathematically, it yields the same optimal policy as the RLHF approach in Section 2.2, which is a regularised optimiser for the reward model $\hat{r}$.

2.4. Sequence Likelihood Calibration (SLiC)

Zhao et al. (2023) propose *sequence likelihood calibration* (SLiC) as an alternative to RLHF. Soon after, Liu et al. (2023) refine the SLiC loss by normalising the policy probabilities with the reference policy probabilities in order to get a regularised offline loss. In the remaining, we will refer to the following loss, with a dataset $(y_i^+, y_i^-)_{i=1}^N$ sampled from (μ, λ_p):

$$\frac{1}{N}\sum_{i=1}^{N}\max\left(0, 1-\tau\log\left(\frac{\pi(y_i^+)\pi^{\text{ref}}(y_i^-)}{\pi(y_i^-)\pi^{\text{ref}}(y_i^+)}\right)\right) \quad (5)$$

as the SLiC loss. This loss can be interpreted as an hinge-loss variation of DPO (Liu et al., 2023).

2.5. Identity preference optimisation (IPO)

Azar et al. (2023) note that in general, optimisation of the reward model described in Section 2.2, which is implicitly optimised by DPO, may not always yield an intuitively good policy for the preference probabilities p. They also note that in practice, removing the learnt reward function from the pipeline removes an important source of regularisation in the learning problem, and as such DPO may learn policies that are under-regularised and converge to deterministic actions.

In order to circumvent these two issues, Azar et al. (2023) propose *identity preference optimisation* (IPO). The derivation begins from the objective of aiming to directly optimise preference probabilities (rather than a proxy reward) against a fixed policy μ:

$$\mathop{\mathbb{E}}_{\substack{Y\sim\pi\\Y'\sim\mu}}[p(Y\succ Y')] - \tau\text{KL}(\pi\ ||\ \pi^{\text{ref}}). \quad (6)$$

Similar to Equation (3), the optimal policy for this objective is expressible directly as

$$\pi^*(y) \propto \pi^{\text{ref}}(y)\exp(\tau^{-1}\mathbb{E}_{Y'\sim\mu}[p(y\succ Y')]), \quad (7)$$

which Azar et al. (2023) use to derive the following equivalent offline IPO loss, with a dataset $(y_i^+, y_i^-)_{i=1}^N$ sampled from (μ, λ_p):

$$\frac{1}{N}\sum_{i=1}^{N}\left(\log\left(\frac{\pi(y_i^+)\pi^{\text{ref}}(y_i^-)}{\pi(y_i^-)\pi^{\text{ref}}(y_i^+)}\right)-\tau^{-1}/2\right)^2. \quad (8)$$

The quadratic aspect of the loss discourages log-probability ratios between pairs of actions under π from deviating too far from those under π^{ref}, which ensures regularisation of π against π^{ref}. By contrast, the minimiser of the DPO empirical loss does not have this property.

2.6. Nash-MD-PG

Rather than optimising preference probabilities against an offline dataset generated from some data-generating policy μ, Munos et al. (2023) propose instead interpreting the objective in Equation (6) as one player's objective in a two-player, constant-sum game. Specifically, two players select policies $\pi_1, \pi_2 \in \Delta(\mathcal{Y})$, with player i receiving payoff

$$\mathop{\mathbb{E}}_{\substack{Y\sim\pi_i\\Y'\sim\pi_{-i}}}[p(Y\succ Y')] - \tau\text{KL}(\pi_i\ ||\ \pi^{\text{ref}}) + \tau\text{KL}(\pi_{-i}\ ||\ \pi^{\text{ref}}), \quad (9)$$

where π_{-i} denotes the policy of the other player. Note that holding π_{-i} fixed at μ then yields an equivalent objective to Equation (6) for π_i. The proposal of Munos et al. (2023) is then to find a Nash equilibrium for this game, motivated by the idea that the policies in the resulting Nash equilibrium may be more robust, and are not overly specific to the data-generating distribution μ. On the flip-side there may be some benefit to regularising the sampling distribution toward the data distribution, and Nash-MD-PG has a parameter β that allows for this tradeoff, with self-play in one extreme ($\beta = 0$) and sampling from μ in the other ($\beta = 1$). We denote this algorithm by Nash-MD-PG(β).

The Nash-MD-PG(β) algorithm, motivated by mirror-descent approaches to saddle-point computation, aims to do so by updating the policy π in the direction of the following policy gradient:

$$\nabla\log\pi(y)\left(p(y\succ y') - \tfrac{1}{2} - \tau\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\right),$$

where importantly $y\sim\pi$, and y' is sampled from a *geometric mixture policy* $\pi^{1-\beta}(\pi^{\text{ref}})^\beta$, for some choice of $\beta\in[0,1]$.

3. Comparative discussion of preference optimisation algorithms

The algorithms DPO (Section 2.3), SLiC (Section 2.4), IPO (Section 2.5), and Nash-MD-PG (Section 2.6) are distinct along a number of axes.

Contrastivity. IPO is a *contrastive* algorithm, in that it labels a pair (Y, Y') according to the preference function (via λ_p) into a positive (preferred) and a negative (not preferred) example (Y^+, Y^-), and then updates the policy via gradients flowing through *both samples*, $\pi(Y^+)$ and $\pi(Y^-)$.

By contrast, Nash-MD-PG is not contrastive; only the sampled action's policy probability is directly updated based on the preference. Contrastive algorithms in general have the potential to be more data-efficient, making direct use of both samples in each policy update (see Appendix E for a condition under which a contrastive gradient estimate has lower variance than its non-contrastive counterpart).

Offline/online data. IPO is an *offline* algorithm, working with a fixed dataset, while Nash-MD-PG is an *online* algorithm that makes use of sampled actions from both the current estimated policy π, and a geometric mixture $\pi^{1-\beta}(\pi^{\text{ref}})^{\beta}$. In many settings it may be desirable to work with static, offline data sets, although if it is feasible to gather data online, this can be beneficial from the point of view of effective regularisation. In the offline setting, the data limits our ability to evaluate the quality of learned policies in terms of preferences, which can lead to learning bad policies that choose actions outside the data distribution, but have low empirical loss. Similar to offline RL (Fujimoto et al., 2019), this can be mitigated through regularisation to a reference policy π^{ref}, possibly related to the sampling distribution μ (Jaques et al., 2019; Wu et al., 2019). An alternative to regularisation is to use online data, and to train π on data that is close to what π generates. Online data may not be available in all settings, but, when it is, it can be an effective way to improve performance of learned policies.

Equilibria. The IPO loss is a supervised objective. In particular, its data distribution is fixed, and the optimiser is given by the closed-form policy in Equation (7), which in particular can be interpreted as a policy that is preferred over the data-generating distribution μ, regularised towards π^{ref}. By contrast, Nash-MD-PG is a game-theoretic algorithm, with a loss function whose data distribution and objective change as the estimated policy change themselves. The stationary points for Nash-MD-PG are not defined in a closed-form manner against reference and data policies π^{ref} and μ, but in a self-referential manner.

Regularised Sampling. As discussed in Section 2.6, Nash-MD-PG allows for sampling from a mixture distribution between π and the data-generating distribution μ, and this can also lead to improved performance versus sampling from either policy.

It is clear that there are several combinations of the various properties of previous methods for which no algorithm yet exists, including combinations that could have advantages over previous work, for example an online contrastive method. Table 1 gives an overview of existing methods in terms of what we consider are strengths of these methods, and how the methods introduced in this paper fit in this context, combining these strengths.

4. Online IPO

We first aim to bridge the online/offline divide between IPO and Nash-MD-PG, by proposing a new variant of IPO, *Online IPO*, which makes use of an online, shifting data distribution.

4.1. Algorithm

To derive an update, we first start with the *population* loss for IPO, which is obtained by taking the minibatch loss in Equation (8), and taking an expectation over the dataset (under i.i.d. sampling from (μ, λ_p)). This yields the (offline) IPO population loss

$$\mathop{\mathbb{E}}_{\substack{Y,Y'\sim\mu \\ Y^+,Y^-\sim\lambda_p(Y,Y')}} \left[\left(\log \left(\frac{\pi(Y^+)\pi^{\text{ref}}(Y^-)}{\pi(Y^-)\pi^{\text{ref}}(Y^+)} \right) - \tau^{-1}/2 \right)^2 \right]. \tag{10}$$

The *Online IPO* population loss is given by replacing the static data distribution, highlighted in red above, with the data distribution generated by the current policy, as displayed below

$$\mathop{\mathbb{E}}_{\substack{Y,Y'\sim\mathtt{SG}[\pi] \\ Y^+,Y^-\sim\lambda_p(Y,Y')}} \left[\left(\log \left(\frac{\pi(Y^+)\pi^{\text{ref}}(Y^-)}{\pi(Y^-)\pi^{\text{ref}}(Y^+)} \right) - \tau^{-1}/2 \right)^2 \right]. \tag{11}$$

Here, $\mathtt{SG}[\pi]$ denotes a stop-gradient around π in the data distribution, meaning that although we generate data from π to construct the loss, we do not differentiate through the data-generation process itself.

The population form of the Online IPO loss will be useful in the analysis that follows. We conclude our description of the approach by noting that the sample-based Online IPO loss coincides with the Offline IPO[1] loss in Equation (8), with the exception that the samples $(y_i^+, y_i^-)_{i=1}^N$ are drawn from the current policy π.

4.2. Analysis

Before studying the performance of the newly derived Online IPO loss empirically, we pause to consider it from a theoretical perspective. In particular, we aim to understand for which policies this loss is stationary.

By the analysis for (Offline) IPO (Azar et al., 2023) summarised in Section 2.5, the gradient for the Online IPO loss in Equation (11) is zero iff π satisfies

$$\pi(y) \propto \pi^{\text{ref}}(y) \exp(\tau^{-1} p(y \succ \pi)), \tag{12}$$

[1]For clarity, we will refer to the original formulation of IPO by Azar et al. (2023) as *Offline IPO* in what follows.

Method	Contrastive	Online	Regularised Sampling
RLHF (Christiano et al., 2017)		✓	
DPO (Rafailov et al., 2023)	✓		
SLiC (Zhao et al., 2023)	✓		
IPO (Azar et al., 2023)	✓		
Self-play (Nash-MD-PG($\beta = 0$)) (Munos et al., 2023)		✓	
Nash-MD-PG (Munos et al., 2023)		✓	✓
Online-IPO (Section 4)	✓	✓	
IPO-MD (Section 5)	✓	✓	✓

Table 1. Method comparison in terms of their properties.

where we use the shorthand $p(y \succ \pi) = \mathbb{E}_{Y' \sim \pi}[p(y \succ Y')]$. This is a fixed-point condition; note that π appears on both sides. This in fact says that π is a best-response against itself in the regularised game described in the background for Nash-MD-PG in Section 2.6. Hence, if π satisfies Equation (12), it must be the Nash equilibrium for this game; we have shown the following.

Proposition 4.1. *The minimiser of the online IPO objective is the Nash equilibrium of the regularised game described in Equation (9).*

This is perhaps a surprising conclusion. Offline IPO is not motivated by game-theoretic considerations, yet by moving to an online variant, we have obtained a loss whose stationary point is precisely the Nash equilibrium of the preference game optimised by Nash-MD-PG. In fact, we can go further, and deduce a direct equivalence of expected updates between Online IPO, and *self-play* in this game. The proof for the following result is given in Appendix D.

Proposition 4.2. *The expected gradient of the Online IPO loss in Equation (11) is identical to the self-play update direction in the game with payoff as in Equation 9.*

Here, *self-play* refers to the algorithm in which a policy is updated using gradient ascent on its expected payoff in the game described in Equation (9), against another player using the same policy:

$$\nabla_\pi \left(\mathop{\mathbb{E}}_{Y \sim \pi, Y' \sim \mathtt{SG}[\pi]} \left[p(Y \succ Y') - \tau \mathrm{KL}(\pi \,||\, \pi^{\mathrm{ref}}) \right] \right).$$

Note that in expectation, this corresponds to Nash-MD-PG with $\beta = 0$; however, an important difference is that in Online IPO, updates are *contrastive*, which may result in variance reduction of the gradient estimate. We have therefore established a close connection between Online IPO and Nash equilibria for the regularised game.

4.3. Online DPO

As the DPO and SLiC losses are similar to IPO, a natural question is whether online variants of DPO and SLiC are also related to the regularised game given Equation (9). We explore this question for online DPO in Appendix G. Lemma G.5 gives conditions for the stationary point of the regularised game and online IPO to be a stationary point of online DPO. However, the conditions seem difficult to satisfy: For example, Theorem G.6 says that there is no 2-action problem for which the condition is satisfied, except when preferences are uniform ($p(1 \succ 2) = \frac{1}{2}$). In this sense, apart from the trivial uniform-preference case, online DPO and online IPO are different objectives when $|\mathcal{Y}| = 2$.

As for stationary points of online DPO, we show that, under the Bradley-Terry model assumption, the RLHF solution (Equation (3)) is a stationary point of online DPO (Theorem G.7), coinciding with that of offline DPO.

5. IPO-MD

Having established a connection between Online IPO and self-play, it is natural to consider whether we can improve on self-play by using regularised policies to generate the data, similar to how Nash-MD optimises preferences against a regularised adversary.

5.1. Algorithm

We consider modulating the data distribution used in the IPO loss using the same geometric mixture $(\pi^{1-\beta}(\pi^{\mathrm{ref}})^\beta)$ between online and reference policies as in Nash-MD, we arrive at a new family of algorithms that we call IPO-MD that directly corresponds to the family of policies for Nash-MD-PG. This leads to the population loss

$$\mathop{\mathbb{E}}_{\substack{Y,Y' \sim \mathtt{SG}[\pi^{1-\beta}(\pi^{\mathrm{ref}})^\beta] \\ Y^+, Y^- \sim \lambda_p(Y,Y')}} \left[\left(\log \left(\frac{\pi(Y^+)\pi^{\mathrm{ref}}(Y^-)}{\pi(Y^-)\pi^{\mathrm{ref}}(Y^+)} \right) - \frac{1}{2\tau} \right)^2 \right].$$

where in the data distribution, we write $\pi^{1-\beta}(\pi^{\mathrm{ref}})^\beta$ as shorthand for the policy which is given by normalising the (unnormalised) geometric mixture $\pi^{1-\beta}(y)(\pi^{\mathrm{ref}})^\beta(y)$. For $\beta \in [0, 1]$, we obtain an algorithm that interpolates between these two in a certain sense, analogous to Nash-MD-PG

(Munos et al., 2023); we call this algorithm IPO-MD(β). When $\beta = 0$, we obtain Online IPO, i.e. self-play, and when $\beta = 1$, we get a variation of IPO that tries to improve against a fixed policy π^{ref}. An interesting observation is that if we use a slightly different mixture policy that mixes with μ, $\pi^{1-\beta}(\mu)^{\beta}$, then this actually interpolates between Online IPO and Offline IPO, where for $\beta = 1$ we get back the Offline IPO objective. In practice it is often difficult to get direct access to μ so we would not be able to form this geometric mixture. Some example dynamics in tabular settings are plotted in Appendix F.

5.2. Analysis

We now compare Nash-MD-PG with the new algorithm class IPO-MD described above. First, these two classes of algorithms are practically different. Nash-MD-PG is on-policy, in that the only gradient contributions appearing in its update are those corresponding to actions sampled under the current policy. By contrast, IPO-MD is an off-policy algorithm, since it updates its current policy based on actions sampled under $\pi^{1-\beta}(\pi^{\text{ref}})^{\beta}$.

By the analysis of Offline IPO (Azar et al., 2023), we have that any fixed point π^*_β of IPO-MD(β) must satisfy

$$\pi^*_\beta(y) \propto \pi^{\text{ref}}(y) \exp\left(\tau^{-1} p(y \succ (\pi^*_\beta)^{1-\beta}(\pi^{\text{ref}})^{\beta})\right). \quad (13)$$

In other words, π^*_β is a best-response against $(\pi^*_\beta)^{1-\beta}(\pi^{\text{ref}})^{\beta}$ in the regularised game described in Equation (9). But from the description of the Nash-MD-PG(β) algorithm by Munos et al. (2023), we have that a policy π^* is stationary under this algorithm's update iff

$$\pi^*(y) \propto \pi^{\text{ref}}(y) \exp\left(\tau^{-1} p(y \succ (\pi^*)^{1-\beta}(\pi^{\text{ref}})^{\beta})\right).$$

Hence, the fixed point of IPO-MD(β) coincides with the fixed point of Nash-MD-PG(β).

Having established the equivalence of the stationary points of IPO-MD(β) and Nash-MD-PG(β), we now study their gradients more generally in the following result; see Appendix D for the proof.

Proposition 5.1. *The gradients of the algorithms Nash-MD-PG(β) and IPO-MD(β) are, respectively,*

$$\begin{aligned} g_{\text{Nash-MD-PG}(\beta)} &= -\mathbb{E}_{y\sim\pi}\left[g(y)\right] \\ g_{\text{IPO-MD}(\beta)} &= -\frac{2}{\tau}\mathbb{E}_{y\sim(\pi)^{1-\beta}(\pi^{\text{ref}})^{\beta}}\left[g(y)\right] \end{aligned}$$

where $g(y)$ is given by

$$\nabla \log \pi(y) \left(p(y \succ (\pi)^{1-\beta}(\pi^{\text{ref}})^{\beta}) - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \right).$$

We recover the result mentioned earlier that when $\beta = 0$, IPO-MD($\beta = 0$) (i.e., Online IPO) has a gradient aligned with that of Nash-MD-PG($\beta = 0$) (i.e., Self-Play). Now as soon as $\beta > 0$, the gradients of these two algorithms are different. Interestingly however, as noted above, their fixed points remain the same. We can also relate the fixed points themselves back to the original regularised game given in Equation (9), as described below.

Proposition 5.2. *Let π^*_β be the fixed-point of IPO-MD(β), satisfying Equation (13). The policy $\pi'_\beta = (\pi^*_\beta)^{1-\beta}(\pi^{\text{ref}})^{\beta}$ is the Nash equilibrium for the version of the game in Equation (9) with regularisation parameter τ modified to $\tau(1-\beta)^{-1}$.*

Proof. Using the property in Equation (13), we have

$$\begin{aligned} &\pi'_\beta(y) \\ &\propto (\pi^{\text{ref}}(y))^{1-\beta} \exp(\tau^{-1}(1-\beta)p(y \succ \pi'_\beta))(\pi^{\text{ref}}(y))^{\beta} \\ &\propto \pi^{\text{ref}}(y) \exp(\tau^{-1}(1-\beta)p(y \succ \pi'_\beta)), \end{aligned}$$

which is the Nash equilibrium condition for the game described, as required. □

6. Experiments

We present our results on fine-tuning large language models where we compare our algorithms, online-IPO and IPO-MD, against recent baselines. In this section we only present the results for the online versions of IPO, DPO and SLIC to make the comparison against IPO-MD and Nash-MD-PG fair, and drop the corresponding "online-" prefix for simplicity. We refer the reader to the appendix for results concerning the offline versions of those algorithms. Note that to aid interpretability and reproducibility, we now consider the contextual bandit case where the actions y, also referred as generations, are conditioned on a prompt x.

Setup and algorithms. We perform RLHF-style experiments where we initialise from a supervised-fine-tuned checkpoint, and then further fine-tune using one of the following algorithms: RL (regularised policy gradient), IPO, DPO , SLiC, Nash-MD and IPO-MD. These algorithms use either a learned reward model r_ϕ (RLHF) or a learned preference model p_ϕ (IPO, DPO, SLiC, Nash-MD, and IPO-MD). For our RL baseline, similarly to Munos et al. (2023), we use a regularised policy gradient update:

$$\mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y\sim\pi_\theta(\cdot|x)}} \left[\nabla_\theta \log \pi_\theta(y|x)\left(r_\phi(y|x) - \tau \text{KL}(\pi_\theta(\cdot|x), \pi^{\text{ref}}(\cdot|x))\right)\right],$$

where $r_\phi(y|x)$ is the reward model's value for generation y and context x.

Implementation details. The contrastive offline algorithms such as IPO, DPO and SliC directly optimise the policy π_θ by minimising their respective losses over a pairwise dataset $(x_i, y_i^+, y_i^-)_{i=1}^N$. In practice, we sample batches

$(x_i, y_i^+, y_i^-)_{i=1}^B$ of size $B \ll N$ and we minimise the following loss:

$$\frac{1}{B}\sum_{i=1}^{B} \mathcal{L}_{\texttt{ALGO}}(\theta, x_i, y_i^+, y_i^-),$$

where:

$$\mathcal{L}_{\texttt{IPO}}(\theta, x, y, y') = \left(\log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right) - \tau^{-1}/2\right)^2,$$

$$\mathcal{L}_{\texttt{DPO}}(\theta, x, y, y') = \sigma\left(\tau \log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right)\right),$$

$$\mathcal{L}_{\texttt{SLiC}}(\theta, x, y, y') = \max\left(0, 1 - \tau \log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right)\right).$$

One important detail concerning IPO is that we use a simplified loss in our code. One can remark by expanding the square and removing terms that do not depend on θ that the IPO loss is equivalent to:

$$-\log\left(\frac{\pi_\theta(y|x)}{\pi_\theta(y'|x)}\right) + \tau\left(\log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right)\right)^2.$$

The contrastive online algorithms such as IPO, DPO and SLiC use a trained preference model p_ϕ. To train p_ϕ, we use a pairwise dataset $(x_i, y_i^+, y_i^-)_{i=1}^N$ and follow the same protocol as Munos et al. (2023). Then, to train the policy π_θ, for each context x_i of a batch, we sample two completely new generations $(y_i, y_i') \sim \pi_\theta$ according to π_θ and compute the preference $p_i = p_\phi(y_i \succ y_i'|x_i)$ via the preference model. Then, for each algorithm, we minimise the respective following loss:

$$\frac{1}{B}\sum_{i=1}^{B} \left(p_i \mathcal{L}_{\texttt{ALGO}}(\theta, x_i, y_i, y_i') + (1-p_i)\mathcal{L}_{\texttt{ALGO}}(\theta, x_i, y_i', y_i)\right).$$

Finally, regarding IPO-MD, the only difference with (online)-IPO is how the generations are sampled. In theory, we should sample from the mixture $\pi_\theta^{1-\beta}(\pi^{\text{ref}})^\beta$ which is not feasible (see (Munos et al., 2023)). In practice, we sample from the one-step-at-a-time mixture $\hat{\pi}_\beta$, which consists at step n to compute the mixture of logits between the online and reference logits:

$$\log(\hat{\pi}_\beta(.|y_{0:n-1}, x)) = (1-\beta)\log(\pi_\theta(.|y_{0:n-1}, x)) + \beta \log(\pi^{\text{ref}}(.|y_{0:n-1}, x)) + C(y_{0:n-1}, x),$$

where $C(y_{0:n-1}, x)$ is a path-dependent constant and sample according to the softmax of this mixture of logits. This sampling process is not equivalent to sampling from $\pi_\theta^{1-\beta}(\pi^{\text{ref}})^\beta$ as shown in (Munos et al., 2023). More implementational details such as pseudo-codes and diagrams for offline and online contrastive preference algorithms can be found in Appendix B.

Evaluation tasks and Models. In our experiments, we test all of the algorithms on an article summarisation task. We use the dataset described by Stiennon et al. (2020) that has been built from the TL;DR dataset (Völske et al., 2017). This is a dataset with pairwise preferences between alternate summaries. We train our preference and reward model on the train set $D_{\texttt{Train}}$, which contains 92820 examples. We evaluate reward and preference models on a test set of high confidence data $D_{\texttt{Test}}$ and use the checkpoints with the highest evaluation agreement score. To train the policies with online algorithms, we use prompts of the train set of the XSum dataset (Shashi et al., 2018).

We use T5X large language models (Roberts et al., 2022) to train our policies, rewards and preference models. The T5X models we use are auto-regressive transformers with an encoder-decoder architecture. All the details of the models architecture and the different sizes are provided in the documentation (Roberts et al., 2022). For the policy model, we use a *large* (L) encoder-decoder model ($770M$ parameters). For the preference and reward models, we use an *XL* encoder-decoder model ($3B$ parameters). To train reward and preference models we use the same losses and protocol as Munos et al. (2023). For summarisation, we initialise our policy with a T5X-L model and fine-tune it with supervised learning using the OpenAI dataset described by Stiennon et al. (2020). We call this supervised fine-tuned model the SFT. All our policies for summarisation are initialised with this SFT checkpoint.

Our evaluation pipeline is based upon the use of PaLM2 (Anil et al., 2023) as a judge for side-by-side comparisons. We sample responses for each of the policies trained by each algorithm from a test set of prompts, and ask PaLM2 to pick which one is better. We use validation and test prompts from the XSum dataset (Shashi et al., 2018) for evaluation for the summarisation task, which is the same procedure used by Munos et al. (2023). The evaluation prompt we use for the side-by-side comparison is:

```
You are an expert summary rater.  Given
a piece of text and two of its possible
summaries, output 1 or 2 to indicate which
summary is better.

Text - <text>, Summary 1 - <summary1>,
Summary 2 - <summary2>

Preferred Summary -
```

We use cloud Tensor Processing Units (TPUs; Jouppi et al., 2023) in their version $5e$ for our hardware compute, either in configurations of 2×4 devices for training offline experiments, or 4×4 devices for online experiments. This setup typically yields speed of around 0.25 training steps per second (24 hours per $20,000$ steps). We run our experiments with default parameters 10^{-4} for the learning rate,

$p(y \succ y')$	IPO	IPO-MD	DPO	Nash-MD-PG	SLiC	RL
IPO	0.500	**0.515** (0.024)	**0.608** (0.038)	**0.621** (0.030)	**0.608** (0.025)	**0.791** (0.012)
IPO-MD	**0.485** (0.024)	0.500	**0.600** (0.028)	**0.608** (0.026)	**0.594** (0.020)	**0.778** (0.004)
DPO	0.392 (0.038)	0.400 (0.028)	0.500	0.520 (0.041)	0.493 (0.040)	0.727 (0.020)
Nash-MD-PG	0.379 (0.030)	0.392 (0.026)	0.480 (0.041)	0.500	0.479 (0.029)	0.729 (0.020)
SLiC	0.392 (0.025)	0.406 (0.020)	0.507 (0.040)	0.521 (0.029)	0.500	0.728 (0.010)
RL	0.209 (0.012)	0.222 (0.004)	0.273 (0.020)	0.271 (0.020)	0.272 (0.010)	0.500

Table 2. Side-by-side evaluation for summarisation. Each entry is the average preference of responses generated by the row method (y) over the responses generated by the column method (y'). We also evaluated each method using 3 different seeds, computed a 3×3 comparisons matrix across seeds and report the standard deviation of this matrix's entries.

and a default total of $30,000$ training steps, using a batch size of 32. The τ factor is held constant throughout training, and we do not employ any warmup steps. The optimiser we use is AdaFactor (Shazeer & Stern, 2018) with a decay rate value of 0.8.

6.1. Main results

In this section, we present side-by-side evaluation scores between the following online algorithms: RL, IPO, DPO, SLiC, IPO-MD and Nash-MD-PG. Table 2 presents the side by side scores for the summarisation task. The checkpoints we evaluate are the best checkpoints except for RL that we use as a baseline of comparison to find the best checkpoints. The RL checkpoint is fixed and was chosen following the protocol of (Munos et al., 2023) after sweeping over 6 values of τ ($\{0.01, 0.02, 0.05, 0.1, 0.15, 0.2\}$) and comparing the performance against the SFT checkpoint after $10k$ learner steps. To find the best checkpoints for the other algorithms, we evaluate every checkpoint of each algorithm against the RL checkpoint (over 2000 prompts sampled from a validation split) at different learning steps values (we checkpoint every 2000 learner steps for a total of $30k$ learner steps), regularisation parameter τ (we sweep over 5 values $\{0.1, 0.5, 1.0, 5.0, 10.0\}$) and also β for IPO-MD and Nash-MD-PG (we sweep over 2 values 0.125 and 0.25) and we take the best checkpoint. After finding the best checkpoint for every algorithm (see App. C.3), we re-run each method for 3 different seeds using the best hyperparamters. We then perform 9 side-by-side evaluation (i.e., 3×3 1vs1 evaluations between each of the 3 seeds for each pair of methods) using 2000 prompts from a different validation split for each comparison. We report mean and standard deviation across these 9 comparisons.

On the summarisation task, looking only at the mean the best algorithm is IPO as it beats all the other algorithms on a side-by-side comparison. However, once we take into consideration the standard deviation IPO and IPO-MD's performance becomes statistically indistinguishable, with both algorithms consistently beating all the other algorithms. This shows that those algorithms are indeed robust and are closer to a Nash optimum than the other algorithms. Those results are limited to a summarisation task and more experiments should be conducted to validate these results on a general conversational agent. However, we do think that summarisation is a good test bed to showcase the quality of human alignment algorithms because it is a complex and high-in-demand task.

6.2. Ablations and Additional Results on Summarisation

Sweep on the regularisation parameter τ: Figure 1 sweeps the regularisation parameter for IPO and DPO. It's interesting to see for small values of regularisation IPO and DPO behave very similarly, but for larger values the score for IPO decays much faster. This matches the findings in Azar et al. (2023) that show that IPO has a much stronger regularisation effect than DPO as τ gets larger.

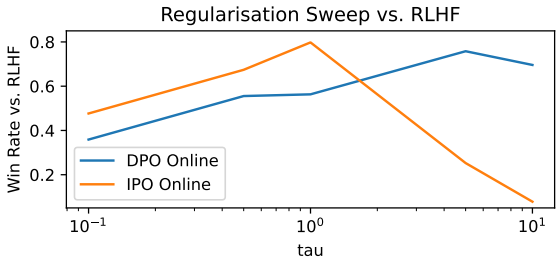

Figure 1. Sweep on regularisation parameter vs. RL on the *summarisation* task.

Learning steps curve Figure 2 shows the performance against RL for IPO Online as it trains. We can see that as the regularisation gets stronger, more training time is required to reach the best performance.

6.3. Offline vs Online Settings

To complement the results reported in Table 2, we include here also additional experiments that compare online and offline variants of IPO/DPO, as well as comparisons against the SFT checkpoint used to initialize all the fine-tuning runs.

$p(y \succ y')$	IPO	DPO	Offline-IPO	Offline-DPO	RLHF	SFT
IPO	0.500	0.608 (0.038)	0.972 (0.008)	0.962 (0.008)	0.791 (0.012)	0.996 (0.001)
DPO	0.392 (0.038)	0.500	0.958 (0.007)	0.944 (0.012)	0.727 (0.020)	0.995 (0.001)
Offline-IPO	0.028 (0.008)	0.042 (0.007)	0.500	0.459 (0.042)	0.096 (0.011)	0.840 (0.013)
Offline-DPO	0.038 (0.008)	0.056 (0.012)	0.541 (0.042)	0.500	0.120 (0.019)	0.869 (0.017)
RLHF	0.209 (0.012)	0.273 (0.020)	0.904 (0.011)	0.880 (0.019)	0.500	0.988 (0.000)
SFT	0.004 (0.001)	0.005 (0.001)	0.160 (0.013)	0.131 (0.017)	0.012 (0.000)	0.500

Table 3. Side-by-side evaluation of online vs offline variants for summarisation. Each entry is the average preference of responses generated by the row method (y) over the responses generated by the column method (y'). We also evaluated each method using 3 different seeds, computed a 3×3 comparisons matrix across seeds and report the standard deviation of this matrix's entries.

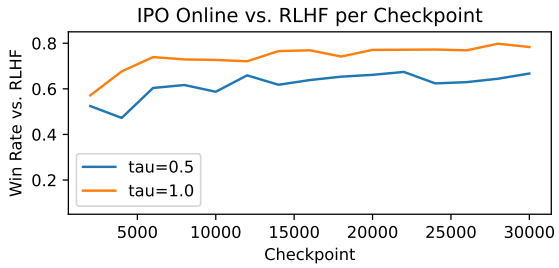

Figure 2. Training Progress of Online IPO (as a function of number of steps) on the *summarisation* task.

While we include them for completeness, note that online and offline methods present drastically different trade-offs, and any comparisons will be greatly influenced by the specific evaluation task. In general, online methods achieve much better exploration and final performance, but are also more computationally expensive than offline methods. Any comparison between online and offline methods will be impacted by the quantity and quality of the offline data, as well as the compute available for online exploration.

Table 3 reports our findings for the experimental pipeline we used in this paper (which evaluates mainly final performance). As we can see, while all methods improve the policy over the inital SFT checkpoint, online methods, including our RLHF baseline, greatly outperform offline methods in terms of final preference in the side-by-side comparison. This however comes at the expense of a larger computational budget. In particular, while the computational complexity of evaluating the IPO/DPO loss is the same for offline and online methods, online methods must generate their samples at runtime (i.e., run inference) which quickly becomes the bottleneck and incurs a roughly 3x slowdown (end-to-end) compared to offline methods that simply need to load pre-existing samples.

7. Conclusion

In this paper, we have identified several factors of variation, such as contrastivity, online/offline and regularised-sampling, between two recently proposed algorithms for preference optimisation, IPO and Nash-MD-PG. In doing so, we have introduced two new algorithms, online-IPO and IPO-MD, that combine different strengths of these existing algorithms, namely the loss function of IPO with the online sampling and regularised data distribution of Nash-MD. Theoretical analysis reveals a surprising equivalence at the level of expected update between online-IPO and self-play in a regularised two-player preference optimisation game. This important property is not possessed by online-DPO. Finally, our empirical investigation on a summarisation task also reveals that IPO-MD and online-IPO are promising approaches to preference optimisation at scale as they are the most robust algorithms. At the moment, our work is restricted to model of size $770M$ on a single task, future works will consist to scale our approach to a full conversational agent using a larger model ($100+$ billions parameter).

Acknowledgements

We are grateful for the collaborative environment at Google DeepMind. We would like to thank Shantanu Thakoor, Will Dabney, Doina Precup, Olivier Bachem, Sertan Girgin, Matt Hoffman, Nikola Momchev, Bobak Shahriari, Piotr Stanczyk, and in particular Mohammad Gheshlaghi Azar for motivating discussions. Finally we would like to thank the anonymous reviewers who helped us improve the quality of this final version.

Impact statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in AI safety. *arXiv*, 2016.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., Chu, E., Clark, J. H., Shafey, L. E., Huang, Y., Meier-Hellstern, K., Mishra, G., Moreira, E., Omernick, M., Robinson, K., Ruder, S., Tay, Y., Xiao, K., Xu, Y., Zhang, Y., Abrego, G. H., Ahn, J., Austin, J., Barham, P., Botha, J., Bradbury, J., Brahma, S., Brooks, K., Catasta, M., Cheng, Y., Cherry, C., Choquette-Choo, C. A., Chowdhery, A., Crepy, C., Dave, S., Dehghani, M., Dev, S., Devlin, J., Díaz, M., Du, N., Dyer, E., Feinberg, V., Feng, F., Fienber, V., Freitag, M., Garcia, X., Gehrmann, S., Gonzalez, L., Gur-Ari, G., Hand, S., Hashemi, H., Hou, L., Howland, J., Hu, A., Hui, J., Hurwitz, J., Isard, M., Ittycheriah, A., Jagielski, M., Jia, W., Kenealy, K., Krikun, M., Kudugunta, S., Lan, C., Lee, K., Lee, B., Li, E., Li, M., Li, W., Li, Y., Li, J., Lim, H., Lin, H., Liu, Z., Liu, F., Maggioni, M., Mahendru, A., Maynez, J., Misra, V., Moussalem, M., Nado, Z., Nham, J., Ni, E., Nystrom, A., Parrish, A., Pellat, M., Polacek, M., Polozov, A., Pope, R., Qiao, S., Reif, E., Richter, B., Riley, P., Ros, A. C., Roy, A., Saeta, B., Samuel, R., Shelby, R., Slone, A., Smilkov, D., So, D. R., Sohn, D., Tokumine, S., Valter, D., Vasudevan, V., Vodrahalli, K., Wang, X., Wang, P., Wang, Z., Wang, T., Wieting, J., Wu, Y., Xu, K., Xu, Y., Xue, L., Yin, P., Yu, J., Zhang, Q., Zheng, S., Zheng, C., Zhou, W., Zhou, D., Petrov, S., and Wu, Y. PaLM 2 technical report, 2023.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv*, 2023.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv*, 2022a.

Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., Chen, A., Goldie, A., Mirhoseini, A., McKinnon, C., Chen, C., Olsson, C., Olah, C., Hernandez, D., Drain, D., Ganguli, D., Li, D., Tran-Johnson, E., Perez, E., Kerr, J., Mueller, J., Ladish, J., Landau, J., Ndousse, K., Lukoiūtė, K., Lovitt, L., Sellitto, M., Elhage, N., Schiefer, N., Mercado, N., DasSarma, N., Lasenby, R., Larson, R., Ringer, S., Johnston, S., Kravec, S., Showk, S. E., Fort, S., Lanham, T., Telleen-Lawton, T., Conerly, T., Henighan, T. J., Hume, T., Bowman, S., Hatfield-Dodds, Z., Mann, B., Amodei, D., Joseph, N., McCandlish, S., Brown, T. B., and Kaplan, J. Constitutional AI: Harmlessness from AI feedback. *arXiv*, 2022b.

Boyd, S. P. and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, 2004.

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

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

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017.

Coste, T., Anwar, U., Kirk, R., and Krueger, D. S. Reward model ensembles help mitigate overoptimization. *arXiv*, 2023.

Dong, H., Xiong, W., Goyal, D., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. RAFT: Reward rAnked FineTuning for generative foundation model alignment. *arXiv*, 2023.

Eisenstein, J., Nagpal, C., Agarwal, A., Beirami, A., D'Amour, A., Dvijotham, D., Fisch, A., Heller, K., Pfohl, S. R., Ramachandran, D., Shaw, P., and Berant, J. Helping or herding? Rward model ensembles mitigate but do not eliminate reward hacking. *arXiv*, 2023.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *Proceedings of the International Conference on Machine Learning*, 2019.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *Proceedings of the International Conference on Machine Learning*, 2022.

Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., Campbell-Gillingham, L., Uesato, J., Huang, P.-S., Comanescu, R., Yang, F., See, A., Dathathri, S., Greig, R., Chen, C., Fritz, D., Elias, J. S., Green, R., Mokrá, S., Fernando, N., Wu, B., Foley, R., Young, S., Gabriel, I., Isaac, W., Mellor, J., Hassabis, D., Kavukcuoglu, K., Hendricks, L. A., and Irving, G. Improving alignment of dialogue agents via targeted human judgements. *arXiv*, 2022.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., and Thomaz, A. L. Policy shaping: Integrating human feedback with reinforcement learning. In *Advances in Neural Information Processing Systems*, 2013.

Hejna, J., Rafailov, R., Sikchi, H., Finn, C., Niekum, S., Knox, W. B., and Sadigh, D. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*, 2023.

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. *arXiv*, 2023.

Jaques, N., Ghandeharioun, A., Shen, J. H., Ferguson, C., Lapedriza, A., Jones, N., Gu, S., and Picard, R. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv*, 2019.

Jouppi, N. P., Kurian, G., Li, S., Ma, P. C., Nagarajan, R., Nai, L., Patil, N., Subramanian, S., Swing, A., Towles, B., Young, C., Zhou, X., Zhou, Z., and Patterson, D. A. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the Annual International Symposium on Computer Architecture*, 2023.

Kirk, R., Mediratta, I., Nalmpantis, C., Luketina, J., Hambro, E., Grefenstette, E., and Raileanu, R. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

Knox, W. B. and Stone, P. TAMER: Training an agent manually via evaluative reinforcement. In *Proceedings of the IEEE International Conference on Development and Learning*, 2008.

Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., Carbune, V., and Rastogi, A. RLAIF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv*, 2023.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. Statistical rejection sampling improves preference optimization. *arXiv*, 2023.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2016.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, D., Tang, Y., Geist, M., Mesnard, T., Michi, A., Selvi, M., Girgin, S., Momchev, N., Bachem, O., Mankowitz, D. J., Precup, D., and Piot, B. Nash learning from human feedback. *arXiv*, 2023.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. WebGPT: Browser-assisted question-answering with human feedback. *arXiv*, 2021.

OpenAI. Introducing ChatGPT, 2022. URL https://openai.com/blog/chatgpt.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. *arXiv*, 2022.

Pan, A., Bhatia, K., and Steinhardt, J. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv*, abs/2201.03544, 2022. URL https://api.semanticscholar.org/CorpusID:245837268.

Pang, R. Y., Padmakumar, V., Sellam, T., Parikh, A. P., and He, H. Reward gaming in conditional text generation. In *Annual Meeting of the Association for Computational Linguistics*, 2022. URL https://api.semanticscholar.org/CorpusID:253553557.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.

Ramé, A., Vieillard, N., Hussenot, L., Dadashi, R., Cideron, G., Bachem, O., and Ferret, J. WARM: On the benefits of weight averaged reward models. *arXiv*, 2024.

Roberts, A., Chung, H. W., Levskaya, A., Mishra, G., Bradbury, J., Andor, D., Narang, S., Lester, B., Gaffney, C., Mohiuddin, A., Hawthorne, C., Lewkowycz, A., Salcianu, A., van Zee, M., Austin, J., Goodman, S., Soares, L. B., Hu, H., Tsvyashchenko, S., Chowdhery, A., Bastings, J., Bulian, J., Garcia, X., Ni, J., Chen, A., Kenealy, K.,

Clark, J. H., Lee, S., Garrette, D., Lee-Thorp, J., Raffel, C., Shazeer, N., Ritter, M., Bosma, M., Passos, A., Maitin-Shepard, J., Fiedel, N., Omernick, M., Saeta, B., Sepassi, R., Spiridonov, A., Newlan, J., and Gesmundo, A. Scaling up models and data with `t5x` and `seqio`. *arXiv*, 2022.

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

Shashi, N., Cohen, S. B., and Mirella, L. Don't Give Me the Details, Just the Summary! Topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2018.

Shazeer, N. M. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv*, 2018.

Shin, D., Dragan, A. D., and Brown, D. S. Benchmarks and algorithms for offline preference-based reward learning. *arXiv*, 2023.

Singhal, P., Goyal, T., Xu, J., and Durrett, G. A long way to go: Investigating length correlations in rlhf. *ArXiv*, abs/2310.03716, 2023.

Skalse, J., Howe, N. H. R., Krasheninnikov, D., and Krueger, D. Defining and characterizing reward gaming. In *Neural Information Processing Systems*, 2022.

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. In *Advances in Neural Information Processing Systems*, 2020.

Swamy, G., Dann, C., Kidambi, R., Wu, Z. S., and Agarwal, A. A minimaximalist approach to reinforcement learning from human feedback. *arXiv*, 2024.

Touvron, H., Martin, L., Stone, K. R., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D. M., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A. S., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I. M., Korenev, A. V., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models. *arXiv*, 2023.

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

Völske, M., Potthast, M., Syed, S., and Stein, B. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*. Association for Computational Linguistics, 2017.

Wang, C., Jiang, Y., Yang, C., Liu, H., and Chen, Y. Beyond reverse KL: Generalizing direct preference optimization with diverse divergence constraints. *arXiv*, 2023.

Warnell, G., Waytowich, N., Lawhern, V., and Stone, P. Deep TAMER: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Wortsman, M., Ilharco, G., Gadre, S. Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A. S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., and Schmidt, L. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Proceedings of the International Conference on Machine Learning*, 2022.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv*, 2019.

Yuan, W., Pang, R. Y., Cho, K., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models, 2024.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv*, abs/2304.05302, 2023.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. SLiC-HF: Sequence likelihood calibration with human feedback. *arXiv*, 2023.

Zhuang, S. and Hadfield-Menell, D. Consequences of misaligned AI. In *Advances in Neural Information Processing Systems*, 2020.

APPENDICES

A. Related Work

RLHF. Reinforcement learning from human feedback, as introduced in Christiano et al. (2017) (see also Bai et al. (2022a); Ouyang et al. (2022)) and often based on proximal policy optimisation (Schulman et al., 2017), is a critical element of making large language models helpful and aligned with preferences of human operators. While in itself it typically does not result in improved benchmark performance (Touvron et al., 2023), RLHF is nonetheless key to satisfying human-mediated interactions such as dialogue (Nakano et al., 2021; Ouyang et al., 2022). The complexity of the RLHF procedure (Casper et al., 2023), which can also be accomplished by multiple reinforcement learning algorithms such as actor-critic (Mnih et al., 2016; Glaese et al., 2022), has led to searching for algorithmic alternatives (Dong et al., 2023; Yuan et al., 2023; Zhao et al., 2023).

Recent developments in preference optimisation. In the special case of an additional Bradley-Terry model (Bradley & Terry, 1952) assumption for the human reward model, reinforcement learning has been found redundant; this allows for casting the problem of RLHF as a supervised one (Rafailov et al., 2023). Recent developments have focused on scaling the performance of such direct preference optimisation (DPO) methods (Tunstall et al., 2023; Ivison et al., 2023), as well as generalising its mathematical formulation (Azar et al., 2023; Wang et al., 2023). One of the key issues with direct preference optimisation - and RLHF in general - resides in their propensity to game or *hack* rewards (Amodei et al., 2016; Skalse et al., 2022; Pan et al., 2022; Pang et al., 2022) and become overoptimised, or under-regularised (Gao et al., 2022; Singhal et al., 2023; Kirk et al., 2023), which can be mitigated e.g. by using ensembling techniques (Wortsman et al., 2022; Eisenstein et al., 2023; Coste et al., 2023; Ramé et al., 2024). Rather than a reinforcement versus supervised learning dichotomy, it is the distinction between *online* and *offline* (Jaques et al., 2019) methods that seems more relevant in practice, as an online policy's generations might start deviating substantially from the original dataset, leading to distribution shifts (Zhuang & Hadfield-Menell, 2020; Shin et al., 2023). Finally, alignment can also result from the self-play form of a two-player game, and not just single-policy optimisation. This perspective, taken in Munos et al. (2023); Swamy et al. (2024) has the added benefit of encompassing both online and offline settings, enabling smooth interpolation between them via a hyperparameter. In a similar vein, DPO has been shown to be able and improve thanks to iterated successive rounds (Yuan et al., 2024), expanding on known machine-critic alignment methods such as reinforcement learning from AI feedback (Bai et al., 2022b; Lee et al., 2023).

B. Additional Implementation Details

In this section, we add the pseudo-codes and diagrams of the offline and online contrastive preference algorithms described in the main paper. This should facilitate reproducibility of the results. We remind the reader of the following losses:

$$\mathcal{L}_{\texttt{IPO}}(\theta, x, y, y') = -\log\left(\frac{\pi_\theta(y|x)}{\pi_\theta(y'|x)}\right) + \tau\left(\log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right)\right)^2,$$

$$\mathcal{L}_{\texttt{DPO}}(\theta, x, y, y') = \sigma\left(\tau\log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right)\right),$$

$$\mathcal{L}_{\texttt{SLiC}}(\theta, x, y, y') = \max\left(0, 1 - \tau\log\left(\frac{\pi_\theta(y|x)\pi^{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi^{\text{ref}}(y|x)}\right)\right).$$

B.1. Pseudo-Codes for Offline and Online Contrastive Preference Algorithm

We remind the reader that offline methods learns from pairwise datasets: $(x_i, y_i^+, y_i^-)_{i=1}^N$ composed of prompts x_i, preferred generations y_i^+ and negative generations y_i^-. Whereas online methods can learn only from a dataset of prompts $(x_i)_{i=1}^N$ but need to use a trained preference model p_ϕ.

Algorithm 1 Offline Contrastive Preference Algorithms (Offline-IPO/DPO/SLiC)

Inputs: A pairwise dataset: $(x_i, y_i^+, y_i^-)_{i=1}^N$, a parameterised policy: π_θ, a reference policy: π^{ref}, a number of total steps K, a batch size B and an optimiser.
for $k = 1$ **to** K **do**
  Sample uniformly a batch: $(x_i, y_i^+, y_i^-)_{i=1}^B$
  Compute the sampled loss: $\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^B \mathcal{L}_{\texttt{ALGO}}(\theta, x_i, y_i^+, y_i^-)$
  Update the policy parameters: $\theta \leftarrow \texttt{UpdateOptimizer}(\theta, \mathcal{L}(\theta))$
end for
Outputs: π_θ

Algorithm 2 Online Contrastive Preference Algorithms (Online-IPO/DPO/SLiC)

Inputs: A dataset of prompts: $(x_i)_{i=1}^N$, a parameterised policy: π_θ, a reference policy: π^{ref}, a parameterised preference model p_ϕ, a number of total steps K, a batch size B and an optimiser.
for $k = 1$ **to** K **do**
  Sample uniformly a batch of prompts: $(x_i)_{i=1}^B$
  For each prompt x_i, generate two independent generations from the policy: $(y_i, y_i') \sim \pi_\theta$
  Score each triplet (x_i, y_i, y_i'): $p_i = p_\phi(y_i \succ y_i' | x_i)$
  Compute the sampled loss: $\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^B \left(p_i \mathcal{L}_{\texttt{ALGO}}(\theta, x_i, y_i, y_i') + (1-p_i)\mathcal{L}_{\texttt{ALGO}}(\theta, x_i, y_i', y_i)\right).$
  Update the policy parameters: $\theta \leftarrow \texttt{UpdateOptimizer}(\theta, \mathcal{L}(\theta))$
end for
Outputs: π_θ

Algorithm 3 Online IPO-MD

Inputs: A dataset of prompts: $(x_i)_{i=1}^N$, a parameterised policy: π_θ, a reference policy: π^{ref}, a parameterised preference model p_ϕ, a number of total steps K, a batch size B and an optimiser.
for $k = 1$ **to** K **do**
  Sample uniformly a batch of prompts: $(x_i)_{i=1}^B$
  For each prompt x_i, generate two independent generations from the mixture policy: $(y_i, y_i') \sim \hat{\pi}_\beta$
  Score each triplet (x_i, y_i, y_i'): $p_i = p_\phi(y_i \succ y_i' | x_i)$
  Compute the sampled loss: $\mathcal{L}(\theta) = \frac{1}{B}\sum_{i=1}^B \left(p_i \mathcal{L}_{\texttt{IPO}}(\theta, x_i, y_i, y_i') + (1-p_i)\mathcal{L}_{\texttt{IPO}}(\theta, x_i, y_i', y_i)\right).$
  Update the policy parameters: $\theta \leftarrow \texttt{UpdateOptimizer}(\theta, \mathcal{L}(\theta))$
end for
Outputs: π_θ

B.2. Diagrams for Offline and Online Contrastive Preference Algorithm

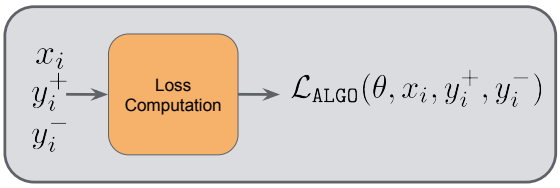

Figure 3. We show how the loss is computed for offline preference contrastive methods.

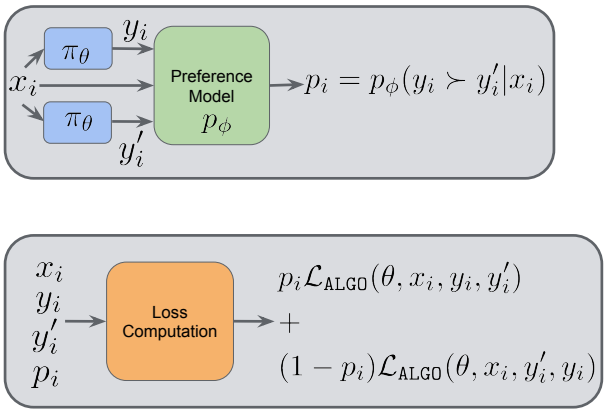

Figure 4. Diagram of Online Preference Contrastive Methods. We show how the generations are sampled and the preference is computed as well as how the loss is computed.

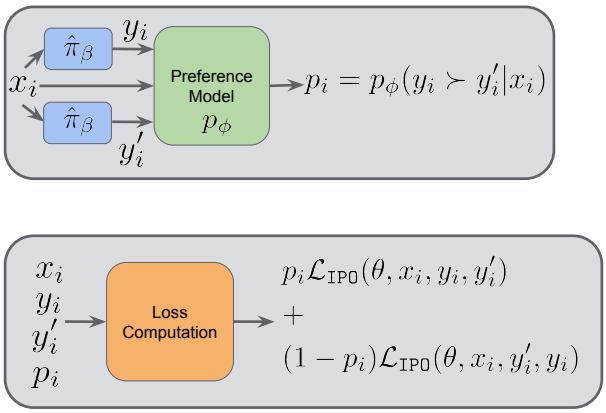

Figure 5. Diagram of IPO-MD.We show how the generations are sampled and the preference is computed as well as how the loss is computed.

C. Additional Experimental Results

C.1. Regularisation Sweep for Online and Offline

Figure 6 and Figure 7 show a sweep over the regularisation parameter for Online and Offline DPO and IPO vs. RLHF and SFT respectively. It is interesting to note that the online versions significantly outperform the offline versions. This is understandable as this setting favours tremendously online methods. Indeed, the starting point is an already fine-tuned policy on summarisation data. Therefore, for online methods the first checkpoint is already able to sample good summaries which make it very easy to obtain good rewards/preferences and from there optimise either the reward/preference model.

C.2. Mixing ratio curve

Here in Figure 8 we show a sweep over the mixing ratio β for IPO-MD and how it affects its win-rate over the RL baseline in summarisation. We draw the curves for a different learning rate and different learning steps than the optimal checkpoint of IPO-MD to show that most of the time the mixing ratio still help improve the performance.

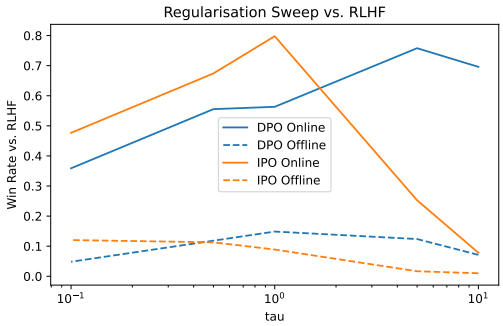

Figure 6. Sweep on Regularisation parameter vs. RL on the *summarisation* task.

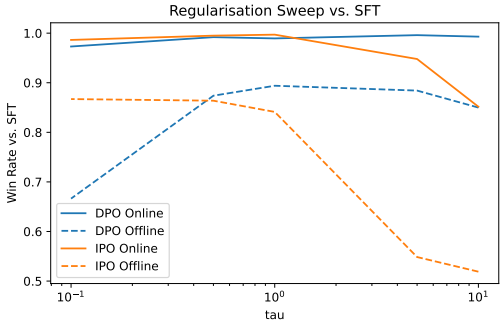

Figure 7. Sweep on Regularisation parameter vs. SFT on the *summarisation* task.

C.3. Best hyperparameters found and KL values for chosen checkpoints

We report here the optimal τ, learning rate `lr` and, where applicable, mixture ratio β used to obtain each algorithm's best performance in Table 2.

	τ	`lr`	β	steps
RL	0.05	10^{-4}	N/A	10000
IPO	1.0	10^{-4}	N/A	28000
DPO	5.0	10^{-4}	N/A	10000
SLiC	10.0	10^{-4}	N/A	30000
IPO-MD	1.0	10^{-4}	0.125	28000
Nash-MD-PG	0.008	$3 \cdot 10^{-5}$	0.125	20000

Table 4. Hyper-parameters for the chosen checkpoints.

We report the value of the KL-divergence between π^{ref} and π along the trajectory of a generation $y \sim \pi(.|x)$ where the prompt x is drawn from the Xsum validation set X of size $|X| = 11305$:

$$\texttt{KL}(\pi||\pi^{\text{ref}}) = \frac{1}{|X|}\sum_{x\in X}\texttt{KL}(\pi(.|x)||\pi^{\text{ref}}(.|x)) = \frac{1}{|X|}\sum_{x\in X}\sum_{t}\texttt{KL}(\pi(.|x,y_{<t})||\pi^{\text{ref}}(.|x,y_{<t}))$$

	IPO	IPO-MD	DPO	Nash-MD-PG	SLiC	RL
KL-value	79.04	80.7	68.26	57.2	79.32	25.28

Table 5.

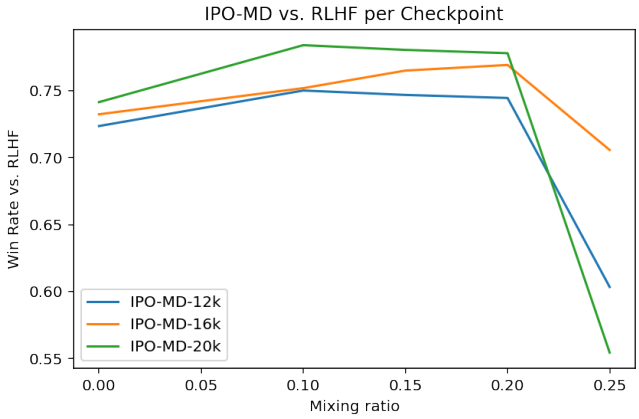

Figure 8. IPO-MD winning rate. The evolving parameter is the mixture ratio β while we fix the learning rate to $3 \cdot 10^{-5}$ and the regularisation parameter to $\tau = 1$. We plot this curves for 3 different learning steps $12k$, $16k$ and $20k$.

D. Proofs

Proposition 4.2. *The expected gradient of the Online IPO loss in Equation (11) is identical to the self-play update direction in the game with payoff as in Equation 9.*

Proof. We calculate expressions for the update directions directly, assuming that the policy π is parametrised via a vector ϕ.

Online IPO update. The update direction for online IPO is given by the negative of the derivative:

$$\begin{aligned}
&\nabla_\phi \mathop{\mathbb{E}}_{\substack{Y,Y'\sim \mathrm{SG}[\pi] \\ Y^+,Y^-\sim\lambda_p(Y,Y')}} \left[\left(\log\left(\frac{\pi(Y)\pi^{\mathrm{ref}}(Y')}{\pi(Y')\pi^{\mathrm{ref}}(Y)} \right) - \tau^{-1}/2 \right)^2 \right] \\
&\propto \sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y \succ y')\nabla_\phi \left(\left(\log\left(\frac{\pi(y)\pi^{\mathrm{ref}}(y')}{\pi(y')\pi^{\mathrm{ref}}(y)} \right) - \tau^{-1}/2 \right)^2 \right) \\
&\propto \sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y \succ y') \left(\log\left(\frac{\pi(y)\pi^{\mathrm{ref}}(y')}{\pi(y')\pi^{\mathrm{ref}}(y)} \right) - \tau^{-1}/2 \right) \nabla_\phi \log(\pi(y)/\pi(y')) \,.
\end{aligned}$$

We can simplify the above by first considering the terms with a factor of τ^{-1}:

$$\begin{aligned}
&-\tau^{-1}/2 \sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y \succ y')(\nabla_\phi \log\pi(y) - \nabla_\phi \log\pi(y')) \\
=&-\tau^{-1}/2 \left(\sum_{y\in\mathcal{Y}} \pi(y)p(y \succ \pi)\nabla_\phi \log\pi(y) - \sum_{y'\in\mathcal{Y}} p(\pi \succ y')\pi(y')\nabla_\phi \log\pi(y') \right) \\
=&-\tau^{-1}/2 \left(\sum_{y\in\mathcal{Y}} \pi(y)p(y \succ \pi)\nabla_\phi \log\pi(y) - \sum_{y'\in\mathcal{Y}} (1 - p(y' \succ \pi))\pi(y')\nabla_\phi \log\pi(y') \right) \\
=&-\tau^{-1} \sum_{y\in\mathcal{Y}} \pi(y)p(y \succ \pi)\nabla_\phi \log\pi(y) \,.
\end{aligned}$$

Now considering the terms not involving τ:

$$
\begin{aligned}
&\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\left(\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)-\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\right)(\nabla_\phi\log\pi(y)-\nabla_\phi\log\pi(y'))\\
&=\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y)\\
&\qquad+\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\nabla_\phi\log\pi(y')\\
&\qquad-\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y')\\
&\qquad-\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\nabla_\phi\log\pi(y)\\
&=\sum_{y,y'\in\mathcal{Y}} \pi(y)p(y\succ \pi)\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y)+\sum_{y,y'\in\mathcal{Y}} \pi(y')p(\pi\succ y')\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\nabla_\phi\log\pi(y')\\
&\qquad-\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y')\\
&\qquad-\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')(1-p(y\succ y'))\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y')\\
&=\sum_{y,y'\in\mathcal{Y}} \pi(y)p(y\succ \pi)\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y)+\sum_{y,y'\in\mathcal{Y}} \pi(y')(1-p(y\succ \pi))\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\nabla_\phi\log\pi(y')\\
&\qquad-\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y')\\
&=\sum_{y,y'\in\mathcal{Y}} \pi(y')\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\nabla_\phi\log\pi(y')\\
&\qquad-\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')\log\left(\frac{\pi(y)}{\pi^{\text{ref}}(y)}\right)\nabla_\phi\log\pi(y')\\
&=\sum_{y,y'\in\mathcal{Y}} \pi(y')\log\left(\frac{\pi(y')}{\pi^{\text{ref}}(y')}\right)\nabla_\phi\log\pi(y')\,.
\end{aligned}
$$

Self-play update. Self-play leads to the update direction given by

$$
\begin{aligned}
&\nabla_\phi\left[\sum_{y,y'\in\mathcal{Y}} \pi(y)\mathtt{SG}[\pi(y')]p(y\succ y')-\tau\sum_{y\in\mathcal{Y}}\pi(y)\log(\pi(y)/\pi^{\text{ref}}(y))\right]\\
&=\sum_{y,y'\in\mathcal{Y}} \pi(y)\pi(y')p(y\succ y')\nabla_\phi\log\pi(y)-\tau\sum_{y\in\mathcal{Y}}\pi(y)\log(\pi(y)/\pi^{\text{ref}}(y))\nabla_\phi\log\pi(y)\\
&=\sum_{y} \pi(y)p(y\succ \pi)\nabla_\phi\log\pi(y)-\tau\sum_{y\in\mathcal{Y}}\pi(y)\log(\pi(y)/\pi^{\text{ref}}(y))\nabla_\phi\log\pi(y)\,.
\end{aligned}
$$

Therefore the expected online IPO update direction is exactly the same as that of self-play. □

Proposition 5.1. *The gradients of the algorithms Nash-MD-PG(β) and IPO-MD(β) are, respectively,*

$$
\begin{aligned}
g_{\textit{Nash-MD-PG}(\beta)} &= -\mathbb{E}_{y\sim\pi}\left[g(y)\right]\\
g_{\textit{IPO-MD}(\beta)} &= -\frac{2}{\tau}\mathbb{E}_{y\sim(\pi)^{1-\beta}(\pi^{\text{ref}})^{\beta}}\left[g(y)\right]
\end{aligned}
$$

where $g(y)$ is given by

$$\nabla \log \pi(y) \left(p(y \succ (\pi)^{1-\beta}(\pi^{\text{ref}})^{\beta}) - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \right) .$$

Proof. The gradient of Nash-MD-PG(β) is:

$$\begin{aligned} g_{\text{Nash-MD-PG}(\beta)} &= -\mathbb{E}_{y\sim\pi, y'\sim\pi'} \left[\nabla \log \pi(y) \left(p(y \succ y') - \frac{1}{2} \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \right) \right] \\ &= -\mathbb{E}_{y\sim\pi} \Big[\underbrace{\nabla \log \pi(y) \left(p(y \succ \pi') - \frac{1}{2} - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \right)}_{g(y)} \Big] \end{aligned}$$

where we write $\pi' = (\pi)^{1-\beta}(\pi^{\text{ref}})^{\beta}$.

Now the gradient of IPO-MD(β) is:

$$\begin{aligned} g_{\text{IPO-MD}(\beta)} &= \mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \nabla \left(\rho_\pi(y, y') - \frac{1}{2\tau} \right)^2 \right] \\ &= 2\mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \left(\rho_\pi(y, y') - \frac{1}{2\tau} \right) \nabla \rho_\pi(y, y') \right], \end{aligned}$$

where $\rho_\pi(y, y') = \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} - \log \frac{\pi(y')}{\pi^{\text{ref}}(y')}$, thus $\nabla \rho_\pi(y, y') = \nabla \log \pi(y) - \nabla \log \pi(y')$.

Using the anti-symmetry of the preference model, i.e., $p(y \succ y') = 1 - p(y' \succ y)$, and combining terms, we have that

$$\begin{aligned} \mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \nabla \rho_\pi(y, y') \right] &= \mathbb{E}_{y\sim\pi'} \left[p(y \succ \pi') \nabla \log \pi(y) \right] - \mathbb{E}_{y'\sim\pi'} \left[p(\pi' \succ y') \nabla \log \pi(y') \right] \\ &= 2\mathbb{E}_{y\sim\pi'} \left[\nabla \log \pi(y) \left(p(y \succ \pi') - \frac{1}{2} \right) \right]. \end{aligned}$$

Similarly:

$$\begin{aligned} & \mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \rho_\pi(y, y') \nabla \rho_\pi(y, y') \right] \\ = \; & \mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \left(\log \frac{\pi(y)}{\pi^{\text{ref}}(y)} - \log \frac{\pi(y')}{\pi^{\text{ref}}(y')} \right) \left[\nabla \log \pi(y) - \nabla \log \pi(y') \right] \right], \\ = \; & \mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \nabla \log \pi(y) + (1 - p(y' \succ y)) \log \frac{\pi(y')}{\pi^{\text{ref}}(y')} \nabla \log \pi(y') \right] \\ & -\mathbb{E}_{y,y'\sim\pi'} \left[p(y \succ y') \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \nabla \log \pi(y') + (1 - p(y' \succ y)) \log \frac{\pi(y')}{\pi^{\text{ref}}(y')} \nabla \log \pi(y) \right] \\ = \; & \mathbb{E}_{y,y'\sim\pi'} \left[\log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \left(\nabla \log \pi(y) - \nabla \log \pi(y') \right) \right] \end{aligned}$$

We deduce that:

$$\begin{aligned} g_{\text{IPO-MD}(\beta)} &= -\frac{2}{\tau} \mathbb{E}_{y\sim\pi'} \left[\nabla \log \pi(y) \left(p(y \succ \pi') - \frac{1}{2} \right) \right] + 2\mathbb{E}_{y,y'\sim\pi'} \left[\log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \left(\nabla \log \pi(y) - \nabla \log \pi(y') \right) \right] \\ &= -\frac{2}{\tau} \mathbb{E}_{y,y'\sim\pi'} \left[\nabla \log \pi(y) \left(p(y \succ \pi') - \frac{1}{2} \right) - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \left(\nabla \log \pi(y) - \nabla \log \pi(y') \right) \right] \\ &= -\frac{2}{\tau} \mathbb{E}_{y\sim\pi'} \Big[\underbrace{\nabla \log \pi(y) \left(p(y \succ \pi') - \frac{1}{2} - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \right)}_{=g(y)} \Big] - 2\mathbb{E}_{y\sim\pi'} \left[\log \frac{\pi(y)}{\pi^{\text{ref}}(y)} \right] \underbrace{\mathbb{E}_{y\sim\pi'} \left[\nabla \log \pi(y) \right]}_{=0} \end{aligned}$$

where we used that $\mathbb{E}_{y\sim\pi'}\left[\nabla \log \pi(y)\right] = \frac{1}{1-\beta}\mathbb{E}_{y\sim\pi'}\left[\nabla \log \pi'(y)\right] = 0$, from the definition of π'.

We deduce that

$$\begin{aligned} g_{\text{Nash-MD-PG}(\beta)} &= -\mathbb{E}_{y\sim\pi}\left[g(y)\right] \\ g_{\text{IPO-MD}(\beta)} &= -\frac{2}{\tau}\mathbb{E}_{y\sim\pi'}\left[g(y)\right]. \end{aligned}$$

□

E. Comparison of the variance of contrastive versus non-contrastive gradient estimates

Define the following gradient estimates based on non-contrastive vs contrastive loss functions:

$$\begin{aligned} \hat{g}_{\text{non-contrastive}} &= -\nabla \log \pi(y)\left(p(y \succ y') - \frac{1}{2} - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} + \tau \log \frac{\pi(y')}{\pi^{\text{ref}}(y')}\right) \\ \hat{g}_{\text{contrastive}} &= -\frac{1}{2}\left(\nabla \log \pi(y) - \nabla \log \pi(y')\right)\left(p(y \succ y') - \frac{1}{2} - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} + \tau \log \frac{\pi(y')}{\pi^{\text{ref}}(y')}\right). \end{aligned}$$

These two estimate resemble those of the algorithms Self-Play (as implemented by Nash-MD-PG($\beta = 0$)), which uses a non-contrastive loss, and online-IPO (equivalent to IPO-MD($\beta = 0$)), which uses a contrastive loss, and we have

$$\begin{aligned} g_{\text{Self-Play}} &= \mathbb{E}_{y,y'\sim\pi}\left[\hat{g}_{\text{non-contrastive}}\right] \\ g_{\text{online-IPO}} &= \frac{2}{\tau}\mathbb{E}_{y,y'\sim\pi}\left[\hat{g}_{\text{contrastive}}\right]. \end{aligned}$$

We know from the previous result that these two estimates have the same expectation. However their variance may differ. Their respective variance depends on a non-trivial combinaison of the policy representation and the specifics of the preference model. We now state a sufficient condition under which the contrastive gradient estimate has lower variance than its non-contrastive counterpart.

Proposition E.1. *If the policy representation and the preference model are such that we have*

$$\mathbb{E}_{y,y'\sim\pi}\left[\nabla \log \pi(y) \nabla \log \pi(y') f(y,y')^2\right] \geq 0, \tag{14}$$

where $f(y,y') := p(y \succ y') - \frac{1}{2} - \tau \log \frac{\pi(y)}{\pi^{\text{ref}}(y)} + \tau \log \frac{\pi(y')}{\pi^{\text{ref}}(y')}$*, then the variance of the contrastive gradient estimate is at least as small as that of the non-contrastive one:* $\mathrm{Var}(\hat{g}_{\text{contrastive}}) \leq \mathrm{Var}(\hat{g}_{\text{non-contrastive}})$.

Proof. Defining the random variables $X_1(y,y') := -\nabla \log \pi(y) f(y,y')$ and $X_2(y,y') := \nabla \log \pi(y') f(y,y')$, we have

$$\begin{aligned} \hat{g}_{\text{non-contrastive}} &= X_1(y,y') \\ \hat{g}_{\text{contrastive}} &= \frac{X_1(y,y') + X_2(y,y')}{2}. \end{aligned}$$

Now, let us compare the variance of these estimates when y and y' are independently drawn from the same distribution π. The variance of the contrastive estimate is

$$\mathrm{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{\mathrm{Var}(X_1) + \mathrm{Var}(X_2) + 2\mathrm{Cov}(X_1, X_2)}{4} = \frac{1}{2}\left(\mathrm{Var}(X_1) + \mathrm{Cov}(X_1, X_2)\right).$$

This variance is lower than $\mathrm{Var}(X_1)$ as soon as $\mathrm{Cov}(X_1, X_2)$ is negative (this is the principle of antithetic variates for variance reduction).

Using the property that $f(y,y') = -f(y',y)$, and writing $c = \mathbb{E}_{y,y'\sim\pi}\left[\nabla \log \pi(y) f(y,y')\right] = -\mathbb{E}_{y,y'\sim\pi}\left[\nabla \log \pi(y') f(y,y')\right]$, we have:

$$\begin{aligned} \mathrm{Cov}(X_1, X_2) &= \mathbb{E}_{y,y'\sim\pi}\left[\left(-\nabla \log \pi(y) f(y,y') + c\right)\left(\nabla \log \pi(y') f(y,y') + c\right)\right] \\ &= \mathbb{E}_{y,y'\sim\pi}\left[-\nabla \log \pi(y) \nabla \log \pi(y') f(y,y')^2\right] - c^2. \end{aligned}$$

We deduce that a sufficient condition for the contrastive estimate to have a lower variance than that of the non-contrastive estimate is:

$$\mathbb{E}_{y,y'\sim\pi}\left[\nabla \log \pi(y) \nabla \log \pi(y') f(y,y')^2\right] + \left(\mathbb{E}_{y,y'\sim\pi}\left[\nabla \log \pi(y) f(y,y')\right]\right)^2 \geq 0.$$

This condition is true as soon as Equation (14) is satisfied. □

E.1. A Toy Example

In this section, we present a toy example that shows a lower variance for a contrastive gradient estimate versus a non-contrastive gradient estimate. For simplicity, let's consider the gradients when there is no regularization involved:

$$\begin{aligned}\hat{g}_{\text{non-contrastive}} &= -\nabla \log \pi(y) \left(p(y \succ y') - \frac{1}{2} \right) \\ \hat{g}_{\text{contrastive}} &= -\frac{1}{2} \left(\nabla \log \pi(y) - \nabla \log \pi(y') \right) \left(p(y \succ y') - \frac{1}{2} \right).\end{aligned}$$

Consider for π_θ a scalar Gaussian distribution centered in θ (which is the parameter) and unit variance, and define the preference

$$p(y \succ y') = \mathbb{I}\{y \geq y'\}.$$

Then we have that $\nabla_\theta \log \pi_\theta(y) = y - \theta$ and both estimate have the same expectation:

$$\mathbb{E}[\hat{g}_{\text{non-contrastive}}] = \mathbb{E}[\hat{g}_{\text{contrastive}}] = -\mathbb{E}\left[(y - \theta) \left(\mathbb{I}\{y \geq y'\} - \frac{1}{2} \right) \right].$$

Now the expected squared of the non-contrastive loss is

$$\begin{aligned}\mathbb{E}[\hat{g}^2_{\text{non-contrastive}}] &= \mathbb{E}\left[(y - \theta)^2 \left(\mathbb{I}\{y \geq y'\} - \frac{1}{2} \right)^2 \right] \\ &= \frac{1}{4} \mathbb{E}\left[(y - \theta)^2 \right] = \frac{1}{4},\end{aligned}$$

whereas the expected squared of the contrastive loss is

$$\begin{aligned}\mathbb{E}[\hat{g}^2_{\text{contrastive}}] &= \frac{1}{4} \mathbb{E}\left[(y - \theta - y' + \theta)^2 \left(\mathbb{I}\{y \geq y'\} - \frac{1}{2} \right)^2 \right] \\ &= \frac{1}{16} \mathbb{E}\left[(y - y')^2 \right] = \frac{1}{8} \\ &< \mathbb{E}[\hat{g}^2_{\text{non-contrastive}}].\end{aligned}$$

Since they have the same expectation, the contrastive estimate has lower variance than non-contrastive one.

This shows a particular example where the contrastive estimate has lower variance than its non-contrastive counterpart. However, we would like to highlight that this property is not true in general. Proposition E.1 above gives a sufficient condition for when this happens.

F. Tabular example

To build some intuition for IPO-MD, in Figure 9 we provide a plot of the trajectories obtained with a variety of values of β for the game with preference probabilities as displayed below, with $\tau = 0.1$, and π^{ref} set to the uniform policy.

$$\begin{pmatrix} 0.5 & 0.8 & 0.1 \\ 0.1 & 0.5 & 0.8 \\ 0.9 & 0.1 & 0.5 \end{pmatrix}$$

These payoffs are chosen as a variant of the classic rock-paper-scissors game. For one extreme, with $\beta = 1$, we have Offline IPO, and as the analysis of Azar et al. (2023) suggests, we observe convergence of the dynamics to the regularised best-response against the reference uniform policy, which is also close to uniform. As β is decreased, we observe two phenomena. The first is that the limiting point of the dynamics shifts from the regularised best response described above towards the regularised Nash equilibrium for the game. Second, as the IPO-MD algorithm becomes closer to Online IPO, which was shown to have equivalent dynamics to self-play earlier in the paper, we observe that the dynamics, while still convergent, exhibit a greater degree of cyclic-like behaviour as the equilibrium point is approached, a common characteristic of algorithms related to self-play.

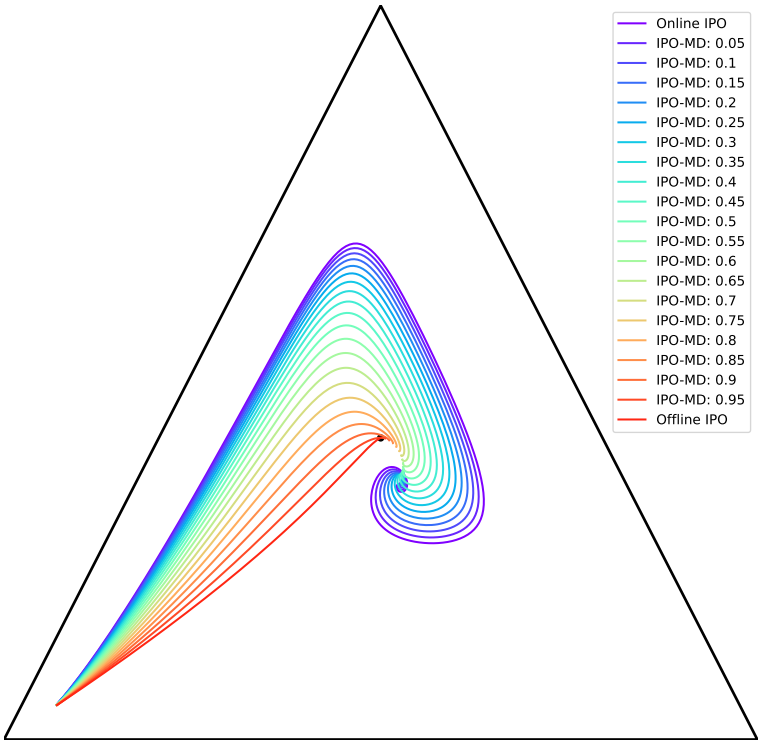

Figure 9. Online, offline, and MD variants of IPO.

G. Supplementary Theoretical Study of Online DPO

In this section, we explore whether online DPO is related to the regularised game given Equation (9), given that Proposition 4.1 shows a similar relationship for online IPO and the regularised game.

Our strategy is to derive the gradient of the offline DPO objective (Lemma G.3), and then inspect whether different sampling distributions μ and candidate solutions π satisfy the KKT conditions (Boyd & Vandenberghe, 2004) for minimising the objective. We can also explore online DPO, by inspecting the KKT conditions when the sampling distribution matches the candidate solution ($\mu = \pi$). For example, in Lemma G.5 we present conditions for the solution of the regularised game (see Equation (12)) to also be a stationary point of online DPO. Given that the minimiser of the online online IPO objective is the Nash equilibrium of the regularised game given in Equation (9) (Proposition 4.1), Lemma G.5 gives us a condition for the online DPO problem to be equivalent (solution-wise) to the online IPO problem. However, the condition seems difficult to satisfy: For example, there is no 2-action problem for which the condition is satisfied with the exception when preferences are uniform ($p(1 \succ 2) = \frac{1}{2}$). In this sense, apart from the trivial uniform-preference case, online DPO and online IPO are different objectives when $|\mathcal{Y}| = 2$.

Regaring stationary points of online DPO, we show that, under the Bradley-Terry model assumption, the RLHF solution (Equation (3)) is a stationary point of online DPO (Theorem G.7). We note in passing (Remark G.9) that the RLHF solution is not the only solution of offline DPO when $\mu(y) = 0$ for some $y \in \mathcal{Y}$. In fact, there are infinitely many, with arbitrarily small probabilities for y such that $\mu(y) > 0$. So while we can say that stationary points of online DPO are also stationary points of offline DPO, a solution for the offline DPO problem with a given μ may not be a solution for corresponding the online DPO problem, because of what happens for y such that $\mu(y) = 0$.

G.1. Results and Proofs

We let $\Delta^\circ(\mathcal{Y}) \doteq \{p \in \Delta(\mathcal{Y}) : p(y) > 0, \forall y \in \mathcal{Y}\}$ be the interior of the simplex.

Proposition G.1. *For $t \in \mathbb{R}$, we have*

$$\frac{d}{dt} \log \sigma(t) = \sigma(-t).$$

Proof. We have:

$$\begin{aligned}\frac{d}{dt} \log \sigma(t) &= \frac{d}{dt} - \log(1 + e^{-t}) = -\frac{1}{1 + e^{-t}} \cdot \frac{d}{dt}(1 + e^{-t}) \\ &= \frac{e^{-t}}{1 + e^{-t}} = \frac{1}{1 + e^{t}} = \sigma(-t).\end{aligned}$$

□

Proposition G.2. *For $t \in \mathbb{R}$, we have*

$$\sigma(t) + \sigma(-t) = 1$$

Proof. We have:

$$\sigma(t) + \sigma(-t) = \frac{1}{1 + e^{-t}} + \frac{1}{1 + e^{t}} = \frac{1}{1 + e^{t}} + \frac{e^{-t}}{1 + e^{-t}} = 1.$$

□

Lemma G.3. *Assume that $|\mathcal{Y}| < \infty$ and that $p(y \succ y') = 1 - p(y' \succ y)$ for all $y, y' \in \mathcal{Y}$. The offline DPO problem with sampling distribution μ can be written as*

$$\sup_{\pi \in \Delta(\mathcal{Y})} J_{\mathrm{DPO}}(\pi),$$

where

$$\begin{aligned}J_{\mathrm{DPO}}(\pi) \doteq \sum_{y,y'} \mu(y)\mu(y') \Bigg(& p(y \succ y') \log \sigma \left(\tau \log \frac{\pi(y)\pi^{\mathrm{ref}}(y')}{\pi(y')\pi^{\mathrm{ref}}(y)} \right) \\ & + (1 - p(y \succ y')) \log \sigma \left(-\tau \log \frac{\pi(y)\pi^{\mathrm{ref}}(y')}{\pi(y')\pi^{\mathrm{ref}}(y)} \right) \Bigg).\end{aligned}$$

Moreover, for $\pi \in \Delta^\circ(\mathcal{Y})$,

$$\nabla J_{\mathrm{DPO}}(\pi)_y = 2\tau \frac{\mu(y)}{\pi(y)} \sum_{y'} \mu(y') \left(p(y \succ y') - \sigma\left(\tau \log \frac{\pi(y)\pi^{\mathrm{ref}}(y')}{\pi(y')\pi^{\mathrm{ref}}(y)} \right) \right). \tag{15}$$

Proof. We will use the shorthands $\mu_{y,y'} \equiv \mu(y)\mu(y')$, $\pi_y \equiv \pi(y)$ (and so forth) and $s_{y,y'} \doteq \tau \log \frac{\pi(y)\pi^{\mathrm{ref}}(y')}{\pi(y')\pi^{\mathrm{ref}}(y)}$.

The offline DPO objective is obtained by taking the expectation of the DPO loss (Rafailov et al., 2023) with $Y^+, Y^- \sim (\mu, \lambda_p)$:

$$\sum_{y,y'} \mu_{y,y'} (p(y \succ y') \log \sigma(s_{y,y'}) + (1 - p(y \succ y')) \log \sigma(-s_{y,y'})) = J_{\mathrm{DPO}}(\pi).$$

Before looking at $\nabla J_{\mathrm{DPO}}(\pi)$, we will make some simplifying observations. For $y'' \neq y, y'$,

$$\frac{d}{d\pi_{y''}} s_{y,y'} = 0. \tag{16}$$

Since $\mu_{y,y'} = \mu_{y',y}$, $s_{y,y'} = -s_{y',y}$ and $p(y \succ y') = 1 - p(y' \succ y)$, we have

$$\begin{aligned} &\mu_{y,y'} \left(p(y \succ y') \log \sigma(s_{y,y'}) + (1 - p(y \succ y')) \log \sigma(-s_{y,y'}) \right) \\ &= \mu_{y',y} \left((1 - p(y' \succ y)) \log \sigma(-s_{y',y}) + p(y' \succ y) \log \sigma(s_{y',y}) \right). \end{aligned} \tag{17}$$

Also note that

$$\frac{d}{d\pi_y} s_{y,y'} = \frac{d}{d\pi_y} \left(\tau \log \frac{\pi_y \pi^{\mathrm{ref}}(y')}{\pi_{y'} \pi^{\mathrm{ref}}(y)} \right) = \tau \frac{d}{d\pi_y} \log \pi_y = \frac{\tau}{\pi_y}. \tag{18}$$

Therefore, for all $y \in \mathcal{Y}$

$$\begin{aligned} &\frac{d}{d\pi_y} J_{\mathrm{DPO}}(\pi) \\ &= 2 \sum_{y'} \mu_{y,y'} \left(p(y \succ y') \frac{d}{d\pi_y} \log \sigma(s_{y,y'}) + (1 - p(y \succ y')) \frac{d}{d\pi_y} \log \sigma(-s_{y,y'}) \right), && \text{(Equations (16) and (17))} \\ &= 2 \sum_{y'} \mu_{y,y'} \left(p(y \succ y') \sigma(-s_{y,y'}) \frac{d}{d\pi_y} s_{y,y'} - (1 - p(y \succ y')) \sigma(s_{y,y'}) \frac{d}{d\pi_y} s_{y,y'} \right) && \text{(Proposition G.1)} \\ &= 2 \sum_{y'} \mu_{y,y'} \left(p(y \succ y') \sigma(-s_{y,y'}) - (1 - p(y \succ y')) \sigma(s_{y,y'}) \right) \frac{d}{d\pi_y} s_{y,y'} \\ &= 2 \sum_{y'} \mu_{y,y'} \left(p(y \succ y') - \sigma(s_{y,y'}) \right) \frac{d}{d\pi_y} s_{y,y'} && \text{(Proposition G.2)} \\ &= 2 \sum_{y'} \mu_{y,y'} \left(p(y \succ y') - \sigma\left(\tau \log \frac{\pi_y \pi^{\mathrm{ref}}_{y'}}{\pi_{y'} \pi^{\mathrm{ref}}_y} \right) \right) \frac{\tau}{\pi_y}, && \text{(Equation (18))} \\ &= 2\tau \frac{\mu_y}{\pi_y} \sum_{y'} \mu_{y,y'} \left(p(y \succ y') - \sigma\left(\tau \log \frac{\pi_y \pi^{\mathrm{ref}}_{y'}}{\pi_{y'} \pi^{\mathrm{ref}}_y} \right) \right). \end{aligned}$$

□

Remark G.4. We have restricted the statement of Lemma G.3 to the interior of the simplex to stay consistent with the DPO formulation, however Equation (15) holds for any function $\exp \circ f$ with $f : \mathcal{Y} \to \mathbb{R}$. To see this, it suffices to notice that J_{DPO} is a function of the ratios $\frac{\pi_i}{\pi_j}$, so for any $\alpha > 0$ $J_{\mathrm{DPO}}(\pi) = J_{\mathrm{DPO}}(\alpha \cdot \pi)$ (where $\alpha \cdot \pi \doteq y \mapsto \alpha \cdot \pi(y)$).

Lemma G.5. *The Nash equilibrium π^* of the regularised game given in Equation (9) is a stationary point of online DPO iff:*

$$p(y \succ \pi^*) = \sum_{y'} \pi^*(y')\sigma(p(y \succ \pi^*) - p(y' \succ \pi^*)). \tag{19}$$

Proof. Let π^* be the Nash equilibrium of the regularised game given in Equation (9), the fixed point in Equation (12):

$$\pi^*(y) \propto \pi^{\text{ref}}(y) \exp\left(\frac{1}{\tau} p(y \succ \pi^*)\right). \tag{20}$$

π^* will be a stationary point of online DPO iff $\pi = \pi^*$ is a solution of the offline DPO problem with $\mu = \pi^*$. Since $\pi^* \in \Delta^\circ(\mathcal{Y})$, we can use Lemma G.3, π^* is a solution of the offline DPO problem iff

$$\nabla J_{\text{DPO}}(\pi^*) = 0.$$

We have for all $y \in \mathcal{Y}$:

$$\begin{aligned}
\frac{d}{d\pi_y} J_{\text{DPO}}(\pi^*) &= 2\tau \frac{\mu(y)}{\pi^*(y)} \sum_{y'} \mu(y') \left(p(y \succ y') - \sigma\left(\tau \log \frac{\pi^*(y)\pi^{\text{ref}}(y')}{\pi^*(y')\pi^{\text{ref}}(y)}\right)\right) && \text{(Equation (15))} \\
&= 2\tau \sum_{y'} \pi^*(y') \left(p(y \succ y') - \sigma\left(\tau \log \frac{\pi^*(y)\pi^{\text{ref}}(y')}{\pi^*(y')\pi^{\text{ref}}(y)}\right)\right) && (\mu = \pi^*) \\
&= 2\tau \sum_{y'} \pi^*(y') \left(p(y \succ y') - \sigma\left(p(y \succ \pi^*) - p(y' \succ \pi^*)\right)\right) && \text{(Equation (20))} \\
&= 2\tau \left(p(y \succ \pi^*) - \sum_{y'} \pi^*(y')\sigma\left(p(y \succ \pi^*) - p(y' \succ \pi^*)\right)\right),
\end{aligned}$$

and the result follows by using the fact that $\tau > 0$. □

Theorem G.6. *No 2-action regularised game given in Equation (9) has a Nash equilibrium that satisfies Equation (19), except for the regularised games with $p(y_1 \succ y_2) = \frac{1}{2}$.*

Proof. Let $\mathcal{Y} = \{1, 2\}$. For this two-action problem, we can write the preference matrix as

$$P \doteq \begin{pmatrix} \frac{1}{2} & 1-p \\ p & \frac{1}{2} \end{pmatrix},$$

where $P_{yy'} = p(y \succ y')$.

Let α be such that $\pi^* = (\alpha, 1-\alpha)^\top$. Then

$$p(y \succ \pi^*) = P\pi^* = \left(1 - \frac{\alpha}{2} - p + \alpha p, \frac{1}{2} - \frac{\alpha}{2} + \alpha p\right)^\top,$$

and

$$p(1 \succ \pi^*) - p(2 \succ \pi^*) = \frac{1}{2} - p.$$

Then the difference between both sides of Equation (19) for $y = 1$ is:

$$\begin{aligned}
&p(1 \succ \pi^*) - \sum_{y'} \pi^*(y')\sigma(p(1 \succ \pi^*) - p(y' \succ \pi^*)) \\
&= 1 - \frac{\alpha}{2} - p + \alpha p - \alpha\sigma(0) - (1-\alpha)\sigma\left(\frac{1}{2} - p\right) \\
&= 1 - \alpha - p + \alpha p - (1-\alpha)\sigma\left(\frac{1}{2} - p\right) \\
&= (1-\alpha)\left(1 - p - \sigma\left(\frac{1}{2} - p\right)\right).
\end{aligned}$$

For $y = 2$ we get

$$\begin{aligned}
&p(2 \succ \pi^*) - \sum_{y'} \pi^*(y')\sigma(p(2 \succ \pi^*) - p(y' \succ \pi^*)) \\
&= \alpha p + \frac{(1-\alpha)}{2} - \alpha\sigma\left(p - \frac{1}{2}\right) - (1-\alpha)\sigma(0) \\
&= \alpha p - \alpha\sigma\left(p - \frac{1}{2}\right) \\
&= \alpha p - \alpha + \alpha\sigma\left(\frac{1}{2} - p\right) \\
&= -\alpha\left(1 - p - \sigma\left(\frac{1}{2} - p\right)\right).
\end{aligned}$$

Now, if we let $\varepsilon \doteq \frac{1}{2} - p$, we can see that for $p < \frac{1}{2}$

$$\sigma\left(\frac{1}{2} - p\right) = \sigma(\varepsilon) < \sigma(0) = \frac{1}{2} < \frac{1}{2} + \varepsilon = 1 - p,$$

so (considering the analogous case for $p < \frac{1}{2}$)

$$1 - p - \sigma\left(\frac{1}{2} - p\right) \begin{cases} > 0, & p < \frac{1}{2}, \\ = 0, & p = \frac{1}{2}, \\ < 0, & p > \frac{1}{2}. \end{cases}$$

Therefore, if $p \neq \frac{1}{2}$, we cannot satisfy Equation (19) for $y = 1$ and $y = 2$ (note that $\alpha = 0$ or $\alpha = 1$ satisfy the equation for only one y). □

Theorem G.7. *Assume that the preferences admit a Bradley-Terry model (Rafailov et al., 2023), that is, there exists $r : \mathcal{Y} \to \mathbb{R}$ such that for all $y, y' \in \mathcal{Y}$*

$$p(y \succ y') = \sigma(r(y) - r(y')),$$

Then

$$\pi^r(y) \propto \pi^{\text{ref}}(y) \exp\left(\frac{1}{\tau} r(y)\right) \tag{21}$$

is a critical point for offline IPO for any μ, and a stationary point of online DPO.

Proof. First, assume that $\pi^{\text{ref}} \in \Delta^\circ(\mathcal{Y})$ (we will deal with the general case at the end).

Let us consider the offline DPO problem with sampling distribution μ. Assuming that the preferences admit a Bradley-Terry model, we know from Rafailov et al. (2023) that the solution of the offline DPO problem is given by

$$\pi^r(y) \propto \pi^{\text{ref}}(y) \exp\left(\frac{1}{\tau} r(y)\right). \tag{22}$$

Indeed, we can verify that for any μ

$$\begin{aligned}
\nabla J_{\text{DPO}}(\pi^r) &= 2\tau \frac{\mu(y)}{\pi(y)} \sum_{y'} \mu(y') \left(p(y \succ y') - \sigma\left(\tau \log \frac{\pi(y)\pi^{\text{ref}}(y')}{\pi(y')\pi^{\text{ref}}(y)}\right)\right) && (\pi^r \in \Delta^\circ(\mathcal{Y}), \text{Equation (15)}) \\
&= 2\tau \frac{\mu(y)}{\pi^r(y)} \sum_{y'} \mu(y') \left(p(y \succ y') - \sigma\left(r(y) - r(y')\right)\right) && (\text{Equation (21)}) \\
&= 2\tau \frac{\mu(y)}{\pi(y)} \sum_{y'} \mu(y') \left(p(y \succ y') - p(y \succ y')\right) && (\text{Bradley-Terry model assumption}) \\
&= 0
\end{aligned}$$

It follows that π^r is an offline DPO solution under any sampling distribution μ, and if $\mu \in \Delta^\circ(\mathcal{Y})$ then π^r is the only solution. In particular this means that π^r is also a stationary point for online DPO (by taking $\mu = \pi^r$).

In the case where $\pi^{\text{ref}} \in \Delta(\mathcal{Y}) - \Delta^\circ(\mathcal{Y})$, we can prove the result as follows. Let $\mathcal{Y}' \doteq \{y : \pi^{\text{ref}}(y) = 0\}$. For $y \in \mathcal{Y}'$, we have $\pi^r(y) = 0$ by Equation (21). Since π^r is the sampling distribution, the gradient of the offline DPO objective in $\mathcal{Y}'$ is zero (and it does not matter that $\log \pi^r(y)$ and $\log \pi^{\text{ref}}(y)$ are undefined since they do not appear in the objective). Now it suffices to note that π^r restricted to $\mathcal{Y} - \mathcal{Y}'$ is a distribution in $\Delta^\circ(\mathcal{Y} - \mathcal{Y}')$, so we can use the previous case ($\pi^r \in \Delta^\circ(\mathcal{Y})$) to show the result, and it follows that π^r satisfies the KKT conditions for all $y \in \mathcal{Y}$. □

Remark G.8. In the special case of rock-paper-scissors preferences (which do not admit a Bradley-Terry model) with π^{ref} uniform and $\tau > 0$, we can still satisfy Equation (19), since π^* is also uniform and $p(y \succ \pi^*) = \frac{1}{2}$.

Remark G.9. In offline DPO, the presence of sets of measure zero guarantees the solution is not unique. Let π^{DPO} be a solution of offline DPO under sampling distribution μ (assume it exists). π^{DPO} may not necessarily be π^r from (21) (even if we assume that the preferences admit a Bradley-Terry model).

To see this, assume that there exists $\mathcal{Y}' \subset \mathcal{Y}$ such that $\mathcal{Y}' \neq \emptyset$ and $\mu(y) = 0$ for all $y \in \mathcal{Y}'$. For any $\alpha \in (0, 1]$, define $\pi^\alpha \in \Delta^\circ(\mathcal{Y})$ by

$$\pi^\alpha(y) \doteq \begin{cases} \alpha \cdot \pi^{\text{DPO}}(y), & y \notin \mathcal{Y}', \\ \text{arbitrary}, & y \in \mathcal{Y}'. \end{cases}$$

By Lemma G.3, $\nabla J_{\text{DPO}}(\pi^\alpha) = 0$, so π^α is a minimum of the offline DPO objective. In fact, this holds for any $\alpha > 0$, no matter how small, and this also means that if $\mu \notin \Delta^\circ(\mathcal{Y})$ we can find offline DPO solutions with arbitrarily small probabilities for sampled y (y such that $\mu(y) > 0$).