Nash Learning from Human Feedback

Rémi Munos [*1] **Michal Valko** [*1] **Daniele Calandriello** [*1] **Mohammad Gheshlaghi Azar** [*1†] **Mark Rowland** [*1] **Daniel Guo** [*1] **Yunhao Tang** [*1] **Matthieu Geist** [*1†] **Thomas Mesnard** [1] **Côme Fiegel** [2] **Andrea Michi** [1] **Marco Selvi** [1] **Sertan Girgin** [1] **Nikola Momchev** [1] **Olivier Bachem** [1] **Daniel J. Mankowitz** [1] **Doina Precup** [1] **Bilal Piot** [*1]

Abstract

Reinforcement learning from human feedback (RLHF) has emerged as the main paradigm for aligning large language models (LLMs) with human preferences. Traditionally, RLHF involves the initial step of learning a reward model from pairwise human feedback, i.e., expressed as preferences between pairs of text generations. Subsequently, the LLM's policy is fine-tuned to maximize the reward through a reinforcement learning algorithm.

In this study, we introduce an alternative pipeline for the fine-tuning of LLMs using pairwise human feedback. Our approach entails the initial learning of a *pairwise* preference model, which is conditioned on two inputs (instead of a single input in the case of a reward model) given a prompt, followed by the pursuit of a policy that consistently generates responses preferred over those generated by any competing policy, thus defining the *Nash equilibrium* of this preference model. We term this approach *Nash learning from human feedback* (NLHF).

In the context of a tabular policy representation, we present a novel algorithmic solution, Nash-MD, founded on the principles of mirror descent. This algorithm produces a sequence of policies, with the last iteration converging to the regularized Nash equilibrium. Additionally, we explore parametric representations of policies and introduce gradient descent algorithms for deep-learning architectures. We illustrate the effectiveness of our approach by presenting experimental results on a text summarization task. We believe NLHF offers a compelling avenue for fine-tuning LLMs and enhancing the alignment of LLMs with human preferences.

[*]Equal contribution [1]Google DeepMind [2]ENSAE Paris [†]Now at Cohere. Correspondence to: Remi Munos <remi.munos@inria.fr>, Michal Valko <michal.valko@inria.fr>, Daniele Calandriello <dcalandriello@google.com>, Bilal Piot <piot@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. Introduction

Large language models (LLMs) (Glaese et al., 2022; Anil et al., 2023; OpenAI, 2023; Ouyang et al., 2022) have made remarkable strides in enhancing natural language understanding and generation. Their success in conversational applications often relies on aligning these models with human preferences, a process primarily guided by the paradigm of reinforcement learning from human feedback (RLHF). A prevailing approach within RLHF involves the initial step of constructing a reward model based on pairwise human preferences, frequently employing the Bradley-Terry model (BT; Bradley & Terry, 1952). This reward model assigns an individual score to each generation of the language model conditioned on a given prompt, akin to how the Elo ranking system (Elo, 1978) assigns scores to chess players to estimate their relative strengths. Subsequently, model refinement takes place by optimizing the LLM's performance with respect to this reward model through reinforcement learning (RL) over sampled text generations.

However, this BT model has its limitations, primarily coming from its inability to accommodate the full spectrum of possible preferences. For example, Bertrand et al. (2023) show the limitations of the Elo model by illustrating where Elo score alone cannot predict the right preferences, even in transitive situations. There are also situations where maximizing the Elo score is not aligned with maximizing the probability of winning against the corresponding population of players, even when the preference model can be perfectly expressed using a BT model (see Appendix A for an example). These observations highlight the necessity for a more profound understanding of the implications of BT-based reward maximization in RLHF for achieving genuine alignment with human preferences.

The NLHF approach: In this paper, we introduce an alternative pipeline for fine-tuning LLMs from human preference data, which we term **Nash learning from human feedback** (NLHF). In this framework, we depart from the conventional approach of learning a reward model and instead focus on learning a **preference model** and define our objective to compute the **Nash equilibrium** of this preference model.

The preference model takes two responses, denoted as y and y' (possibly conditioned on a prompt x), as input and produces a preference score $\mathcal{P}(y \succ y'|x)$, indicating the preference of response y over response y' given the context x. We may think of $\mathcal{P}(y \succ y'|x)$ as the probability that a randomly chosen human prefers response y to response y' given x.

In order to learn a preference model, we can initialize it using AI-feedback by leveraging a LLM prompted in a manner akin to how humans have been asked for their preference, such as by instructing the LLM to generate a 1-vs-2 comparison in response to a prompt like: "Given x, which answer do you prefer, answer 1: y or answer 2: y'?". This initial preference model can be further refined, through supervised learning, by aligning it with human preference data.

Notably, such a learnt preference model does not make any Bradley-Terry assumption thus has the potential to capture the diversity and richness of human preferences contained in the training data.

Moreover, in contrast to the traditional RLHF setting where the reward model depends on the distribution of data that has been used to train the model, a preference model remains essentially invariant to this data distribution. The main reason why preference models are less sensitive to the data distribution than reward models is that preference models takes as input the two responses to be compared whereas the reward model makes an implicit comparison between the (single) input to the distribution it has been trained on (see Section 3.3).

Once the preference model is established, our primary objective is to calculate the corresponding Nash equilibrium. This equilibrium represents a policy that consistently produces responses preferred, as determined by the preference model, over responses generated by any alternative policy.

The three key properties of our approach, namely, the ability of the preference model to capture a wide spectrum of human preferences (contained in the data), its lower sensitivity on the data distribution, and the potential for the Nash equilibrium to provide a better alignment with the diversity of human preferences, mark a substantial departure from the conventional RLHF framework. We discuss these properties in greater detail in Section 3.

Practical algorithms: To approximate the Nash equilibrium of the two-player game in which actions are responses, and payoffs are specified by the preference model, we employ a deep reinforcement learning algorithm. Given a prompt x, we generate two responses, denoted as y and y'. The first response, y, is generated under the current policy π_θ that we are in the process of optimizing. In contrast, the second response, y', is produced by an alternative policy π', which we implement in two different versions: **Nash-MD** and **Nash-EMA** (further elaboration on these versions will be provided below). Nash-MD defines the alternative policy π' as a geometric mixture between the initial and the current policy (motivated by mirror descent), whereas Nash-EMA implements a first-order approximation of an exponential moving average (EMA) mixture of past policies. Then, the preference model computes $\mathcal{P}(y \succ y'|x)$, and this preference signal serves as a reward for optimizing our policy π_θ using a (regularized) policy gradient algorithm, as outlined in (Geist et al., 2019).

Our contributions: Our contributions in this work can be summarized as follows. First, we introduce the concept of Nash learning from human feedback (NLHF), framing it as the task of computing the Nash equilibrium for a general preference model. We proceed by introducing and defining a regularized variant of the preference model. We also establish the existence and uniqueness of the corresponding Nash equilibrium in this context. Then, we consider the case of tabular policy representations and introduce a novel algorithm named Nash-MD. This algorithm, founded on the principles of mirror descent (MD) possesses two important properties. First, it converges to the Nash equilibrium, with the final iteration reaching this equilibrium. This differs from conventional regret-minimization-based algorithms, where it is typically the mixture of past policies that converges, necessitating the storage of past policies. Secondly, Nash-MD learns by competing against alternative policies π' that represent a (geometric) mixture between the current policy π_θ and the initial policy. Importantly, this can be accomplished without the need to retain intermediate policies, a feature of particular significance in the context of LLMs with their substantial memory requirements. Additionally, we introduce Nash-EMA, a variation inspired by fictitious play, which uses an exponential moving average of past policy parameters. We introduce policy-gradient algorithms for deep learning architectures, Nash-MD-PG and Nash-EMA-PG, inspired by the tabular algorithms. We present the results of numerical experiments conducted on a text summarizing task utilizing the TL;DR dataset (Völske et al., 2017). In these experiments, we employ the NLHF approach to train several models. To assess their performance, we conduct a pairwise evaluation (using the PaLM 2 Large LLM) of the performance of the models and include a comparison to an RLHF baseline. We conclude that NLHF

opens up new promising directions for aligning LLMs with human preferences.

2. Prior work

Preference-based RL. Our contribution falls into a broader area of preference-based RL, where we directly learn from pairwise human preferences instead of a hand-designed or learned scalar reward (see, e.g., the survey by Wirth et al., 2017). The canonical form of RLHF was proposed in (Christiano et al., 2017) and popularized by (OpenAI, 2022), in which one learns a scalar reward model from the preference feedback, followed by policy optimization against the reward model. However, an advantage of directly optimizing for preferences rather than a learnt scalar reward function is the potential to avoid *reward hacking* (Amodei et al., 2016), when agents find a way to maximize a reward without performing what was intended. Furthermore, in domains such as medical applications, it may not only be challenging but also undesirable to provide a single scalar reward.

In general, the preference feedback can be provided in different ways, e.g., on the level of states, actions, or a full trajectory. In this work, we focus on the *trajectory feedback* where the experts provide feedback by selecting the preferred one of the two proposed trajectories. Such a simple form of pairwise feedback is the easiest to implement, and has seen applications in summarization (Stiennon et al., 2020), question-answering (Nakano et al., 2021; Menick et al., 2022) and general language-based assistants (Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022). Ranking based algorithms in the RLHF literature include the RAFT (Dong et al., 2023) and ReST (Gulcehre et al., 2023) approaches. More complicated forms of feedback has been studied in theoretical literature such the work of Efroni et al. (2021).

Theoretical guarantees for learning from preferences. Learning policies from preference feedback of histories was studied by Akrour et al. (2011) who learned the preference model for histories and by Cheng et al. (2011) who trained a model ranking actions for a state. Busa-Fekete et al. (2014; 2013) approached this setting by comparing and ranking policies and Wilson et al. (2012) by learning a distribution over policy space. Preference-based RL is also explored in dueling RL (Novoseller et al., 2020; Pacchiano et al., 2023), which generalizes the well-studied *dueling bandits* problem. In particular, Pacchiano et al. (2023) assumes a Bradley-Terry model, which they estimate using maximum likelihood in the tabular setting.

Our work is also related to results of Wang et al. (2023) who consider learning Nash equilibria of the human preference model, and reduce the problem to finding Nash equilibria for a special class of factored two-player Markov games under a restricted set of policies. The interaction of Nash equilibria and LLMs have been also explored by Jacob et al. (2023a;b). Moreover, Chen et al. (2022) gave first results for function approximation in preference-based RL, however with a computationally inefficient algorithm.

Optimization without reward function. A number of recent works has attempted to optimize for preference feedback without learning a reward function. For example, Direct Preference Optimization (DPO; Rafailov et al., 2023) optimizes the policy through a loss function defined via the Bradley-Terry reward model. SLiC (Zhao et al., 2023) modifies the classical RLHF training loss by calibrating a ranking loss which contrasts a positive and a negative sequence. This resembles directly optimizing for the pairwise preference, albeit without convergence guarantees. Identity Preference Optimization (IPO; Azar et al., 2023) and the Generalized Preference Optimization (GPO; Tang et al., 2024) proposed to directly optimize the pairwise human preference with offline preference data by optimizing against a fixed opponent. And recently, it has been observed that the online version of IPO (Calandriello et al., 2024) approximates the Nash equilibrium of a preference model using a particular case of Nash-MD (called Self-Play).

3. The preference model and its Nash equilibrium

We now introduce the core conceptual ideas behind our approach to learning from preference feedback. We consider a preference model in a contextual bandit setting. Given a context (or prompt) x in the context space $\mathcal{X}$ and two actions (or responses/choices) y and y' in the action space $\mathcal{Y}$, the preference of y over y' given x is a number between 0 and 1 which is written $\mathcal{P}(y \succ y'|x)$. We will assume that the preference model is antisymmetric: $\mathcal{P}(y \succ y'|x) = 1 - \mathcal{P}(y' \succ y|x)$.

In the context of LLMs we can think of the preference $\mathcal{P}(y \succ y'|x)$ as the probability that a randomly chosen human prefers a response y over the other response y' given the context x.

We define the preference between two distributions conditioned on a state x:

$$\mathcal{P}(\pi \succ \pi'|x) \stackrel{\text{def}}{=} \mathbb{E}_{y\sim\pi(\cdot|x),y'\sim\pi'(\cdot|x)}\left[\mathcal{P}(y \succ y'|x)\right]$$

and the preference of an action over a distribution $\mathcal{P}(y \succ \pi'|x) \stackrel{\text{def}}{=} \mathbb{E}_{y'\sim\pi'(\cdot|x)}\left[\mathcal{P}(y \succ y'|x)\right]$. Finally, given a distribution ρ over contexts, we define the preference between two policies:

$$\mathcal{P}(\pi \succ \pi') \stackrel{\text{def}}{=} \mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x),y'\sim\pi'(\cdot|x)}\left[\mathcal{P}(y \succ y'|x)\right].$$

We say that a policy π is preferred over (or simply wins against) another policy π' if $\mathcal{P}(\pi \succ \pi') \geq 1/2$. In the remainder of the paper, we assume (without loss of generality) that ρ assigns every context positive probability.

In this paper we will consider the objective of finding a policy π^* which is preferred over any other alternative policy:

$$\pi^* \stackrel{\text{def}}{=} \arg\max_{\pi} \min_{\pi'} \mathcal{P}(\pi \succ \pi')\,. \tag{1}$$

This objective implicitly defines a two-player game, in which the players select policies π and π', the first player receiving a payoff of $\mathcal{P}(\pi \succ \pi')$, and the second player receiving $\mathcal{P}(\pi' \succ \pi) = 1 - \mathcal{P}(\pi \succ \pi')$. This is therefore a two-player, antisymmetric, constant-sum game, and it follows that when both players use a policy π^* solving Equation (1), this is a *Nash equilibrium* for this game, by the minimax theorem (von Neumann, 1928). This is the fundamental solution concept we study in this paper.

The objective introduced in Equation (1) has two central differences relative to the majority of existing work on RLHF. First, the objective is expressed directly in terms of preferences themselves, not in terms of a reward function learnt from preferences, and also not in terms of a non-linear transformation of the preferences. Second, our solution concept relies on the notion of Nash equilibrium, rather than on optimization against a fixed behavior. We discuss the impact of both of these choices through several examples below.

3.1. Limited expressivity of reward models

A learnt preference model possesses the capacity to encompass any property of human preferences as much as they are contained in the dataset used to train the model. For example they can model non-transitive preferences (see the examples in Appendix C), a characteristic not attainable by reward models since they inherently assign a single score to each policy. Whether humans exhibit non-transitive preferences or not has been a subject of longstanding research (see, for instance, (Tversky, 1969; Klimenko, 2015)). But even if single individuals are transitive, it is nevertheless possible that the resulting expected preference model is not (see Appendix C.2 for an example).

Additionally, non-transitivity is not the only limitation of Bradley-Terry-based reward models; see, e.g., Example 3 in (Bertrand et al., 2023) where Elo score fails to capture the correct preference ordering between policies, even in transitive situations. In fact, we show in Appendix A that even when the preference model is perfectly captured by the Bradley-Terry model, optimization of the reward/Elo score may *still* disagree with any reasonable notion of preference optimization.

Therefore, we can safely argue that preference models offer a more flexible and nuanced framework for modeling preferences than BT-based preference models models.

3.2. Alignment with diversity of human preferences

Here, we illustrate that in some situations, the solution offered by the Nash equilibrium of the preference model (which we refer to as the NLHF solution) is more aligned with the diversity of human preferences than the optimum of the reward model (which we refer to as the RLHF solution).

Consider the following situation where there are 3 different actions (y_1, y_2, y_3) and we have a population composed of 3 types of humans with respective preferences $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$, defined in the following way: $\mathcal{P}_i(y_1 \succ y_2) = \mathcal{P}_i(y_1 \succ y_3) = \mathcal{P}_i(y_2 \succ y_3) = 1/2$, for $1 \leq i \leq 3$, except for the following cases: $\mathcal{P}_1(y_2 \succ y_1) = 1$ (thus $\mathcal{P}_1(y_1 \succ y_2) = 0$), $\mathcal{P}_2(y_1 \succ y_3) = 1$ (thus $\mathcal{P}_2(y_3 \succ y_1) = 0$), and $\mathcal{P}_3(y_3 \succ y_2) = 1$ (thus $\mathcal{P}_3(y_2 \succ y_3) = 0$).

Now, let us assume these 3 types form a near-uniform distribution over humans, for example $\mathbb{P}(\text{Type } 1) = 1/3 - \epsilon$, $\mathbb{P}(\text{Type } 2) = \mathbb{P}(\text{Type } 3) = 1/3 + \epsilon/2$. The corresponding population preference is thus $\mathcal{P}_\epsilon = (1/3 - \epsilon)\mathcal{P}_1 + (1/3 + \epsilon/2)(\mathcal{P}_2 + \mathcal{P}_3)$. In the case $\epsilon > 0$ (so Type 1 is slightly less frequent than the other types) then a reward model will assign a slightly better reward (assuming a Bradley-Terry model) to action y_1, thus optimizing the expected reward (the RLHF solution) will produce a deterministic policy choosing exclusively y_1.

However, here we are in a situation where the preferences are not uniformly aligned across humans (Moskovitz et al., 2023). In the case of uniform sampling of humans (i.e., $\epsilon = 0$), the Nash equilibrium of $\mathcal{P}_{\epsilon=0}$ is a uniform mixture between the 3 policies. Actually, the preference model $\mathcal{P}_\epsilon$ corresponding to any ϵ is defined as: $\mathcal{P}_\epsilon(y_2 \succ y_1) = 2/3 - \epsilon/2$, $\mathcal{P}_\epsilon(y_3 \succ y_1) = 1/3 - \epsilon/4$, $\mathcal{P}_\epsilon(y_3 \succ y_2) = 2/3 + \epsilon/4$, $\mathcal{P}_\epsilon(y_i \succ y_i) = 1/2$, and $\mathcal{P}_\epsilon(y_i \succ y_j) = 1 - \mathcal{P}_\epsilon(y_j \succ y_i)$, for $1 \leq i < j \leq 3$. By a simple calculation, we deduce that for any $|\epsilon| \leq 1/3$, the Nash equilibrium of this preference model consists in selecting y_1 and y_2 with probability $1/3 + \epsilon/2$ each, and y_3 with probability $1/3 - \epsilon$.

We believe that in this situation, the Nash solution of the preference model (i.e., the NLHF solution), assigning close to uniform probability to these 3 actions (one being preferred by each category of humans) is more aligned with the diversity of human preferences than the optimum of the reward model (i.e., the RLHF solution), which would deterministically select a single action. Also the Nash equilibrium is less sensitive to the preference distribution, since the corresponding equilibrium is smooth w.r.t. change in the distribution over types of humans (i.e., when ϵ varies near 0), whereas the RLHF solution will switch from selecting exclusively y_1 when $\epsilon > 0$ to selecting exclusively y_2 when

$\epsilon < 0$.

3.3. Sensitivity to the data distribution

Another difference between reward and preference models is that a reward model depends on the distribution it has been trained on, whereas a preference model essentially does not. Indeed, when we learn a reward model we are solving the following optimization problem (in the limit of an infinite amount of data):

$$r^\pi \stackrel{\text{def}}{=} \arg\max_{r(\cdot,\cdot)} \mathbb{E}_{\substack{x\sim\rho \\ y,\, y'\sim\pi(\cdot|x) \\ Z\sim\nu}} \left[\log\left(\sigma(r(x,y_w^Z) - r(x,y_l^Z))\right)\right],$$

where y_w^Z and y_l^Z are respectively the preferred (and less preferred) response (among y and y') according to a randomly sampled human $Z \sim \nu$, given x. The (optimal) solution to this problem r^π depends on the policy π that has generated the data. Indeed, as mentioned in the introduction (see Section 1), the reward model assigns an Elo score to each individual response, which is defined in terms of a comparison against other responses; thus, it depends on the overall distribution over responses it has been trained on. See Theorem 2 (Appendix B) for a precise statement.

On the contrary, since the preference model takes two responses as input, the output does not depend directly on the distribution these responses have been sampled from. The preference model is simply learnt by supervised learning, where for each x, y, y', the preference model $\mathcal{P}(y \succ y'|x)$ is regressed to the human preference $\mathbb{I}\{y \text{ is preferred to } y' \text{ given } x\}$ using a cross entropy loss:

$$\mathcal{P}^* \stackrel{\text{def}}{=} \arg\max_{\mathcal{P}(\cdot\succ\cdot|\cdot)} \mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi(\cdot|x) \\ y'\sim\pi'(\cdot|x) \\ Z\sim\nu}} \left[\log \mathcal{P}(y_w^Z \succ y_l^Z|x)\right],$$

where y_w^Z (resp. y_l^Z) is the preferred (resp. less preferred) response (among the responses y and y') given x according to a random human $Z \sim \nu$. Notice that the optimal solution to this optimization problem is just the probability that a random human prefers y to y' given x: for any $x \in \text{supp}(\rho)$, $y \in \text{supp}(\pi(\cdot|x))$, $y' \in \text{supp}(\pi'(\cdot|x))$,

$$\mathcal{P}^*(y \succ y'|x) = \mathbb{P}_{Z\sim\nu}\left(\text{Human } Z \text{ prefers } y \text{ to } y' \text{ given } x\right).$$

This quantity is just a function of x, y, y' and does not depends on how x, y and y' have been chosen, thus is independent of ρ, π or π'. Thus in this ideal case of perfect representations, a preference model is insensitive to the data generation distribution (ρ, π, π') whereas a reward model depends on it. Now, of course, when using approximate models or a finite amount of data, the learned preference model may still depend on the data distribution as the quality of the approximation depends on the local quantity of collected data.

Thus it is our general expectation that the preference model is significantly less reliant on the specific policy that generated the data when compared to the reward model.

This observation becomes even more important in scenarios where multiple iterations of RLHF/NLHF occur, comprising data collection, constructing a reward/preference model, policy optimization based on the model, and collecting new data following the updated policy. In the case of RLHF, the reward model from a prior iteration diverges from the next iteration due to shifts in data distributions, necessitating complete relearning. On the contrary, in the NLHF approach, the preference model can be preserved and further enriched through the introduction of novel data, thereby offering a more seamless and efficient adaptation process.

4. Regularized preference model

We now consider a regularized version of the preference model. This is motivated by situations where the preference model is more accurately estimated along responses obtained by following a given *reference* policy. This could be the policy responsible for generating the data used to train the preference model or include situations where it is imperative to ensure that our solution remains close to a known safe policy. In such cases, we incorporate a penalty mechanism into our preference model, employing KL-regularization to quantify the divergence between the policy under consideration and a designated reference policy denoted as μ; see (Jaques et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022) for further details on the role of KL-regularization in RLHF.

The regularized preference between actions $y \sim \pi(\cdot|x), y' \sim \pi'(\cdot|x)$ is defined as

$$\mathcal{P}_\tau^{\pi,\pi'}(y \succ y'|x) \stackrel{\text{def}}{=} \mathcal{P}(y \succ y'|x) - \tau \log \frac{\pi(y|x)}{\mu(y|x)} + \tau \log \frac{\pi'(y'|x)}{\mu(y'|x)},$$

and we define accordingly the KL-regularized preference between policies:

$$\begin{aligned}\mathcal{P}_\tau(\pi \succ \pi') &\stackrel{\text{def}}{=} \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x), y'\sim\pi'(\cdot|x)}\left[\mathcal{P}_\tau^{\pi,\pi'}(y \succ y'|x)\right] \\ &= \mathcal{P}(\pi \succ \pi') - \tau \text{KL}_\rho(\pi,\mu) + \tau \text{KL}_\rho(\pi',\mu) \quad (2)\end{aligned}$$

where $\text{KL}_\rho(\pi,\mu) \stackrel{\text{def}}{=} \mathbb{E}_{x\sim\rho}[\text{KL}(\pi(\cdot|x), \mu(\cdot|x))]$. We now state the existence and uniqueness of the Nash equilibrium of this regularized preference model (see the proof in the Appendix E):

Proposition 1 (Nash equilibrium)**.** *There exists a unique Nash equilibrium of the regularized preference model $\mathcal{P}_\tau$.*

5. Algorithms for approximating the Nash eq.

The regularized preference model $\mathcal{P}_\tau(\pi \succ \pi')$ defines a constant-sum two-player game where Player 1 selects π and

Player 2 selects π'. There are well-known techniques for approximating the Nash equilibrium. Some of them offer a convergence on average (in the sense that it is a mixture of the sequence of policies that converges to the Nash equilibrium), whereas other methods offer convergence of the last iterate.

Convergence on average. *Fictitious play* (FP; Brown, 1951; Robinson, 1951; Heinrich et al., 2015; Fudenberg & Levine, 1998) consists in playing, at every iteration, each player's best response against the uniform mixture of the opponent's past strategies. Here we would define $\pi_{t+1} \stackrel{\text{def}}{=} \arg\max_\pi \mathcal{P}(\pi \succ \bar{\pi}_t)$, where $\bar{\pi}_t$ is the mixture policy $\frac{1}{t}\sum_{s=1}^{t}\pi_s$. It is known that the mixture policy $\bar{\pi}_t$ converges to the Nash equilibrium in constant-sum games (see (Hofbauer & Sorin, 2006) for a reference in the general concave-convex case considered here). Also, FP has been considered with function approximation (Heinrich & Silver, 2016). *Online convex optimization:* In the context of solving convex-concave constant-sum games, we rely on online convex optimization where each player minimizes its own convex loss. See for example (Cesa-Biachi & Lugosi, 2006; Nesterov, 2005; Hoda et al., 2010). *Regret minimization* has been extensively considered in games since the average strategy of self-playing no-regret algorithms converges to a Nash equilibrium (Rakhlin & Sridharan, 2013; Kangarshahi et al., 2018). Counterfactual regret minimization (CFR) has been considered in the setting of imperfect information games in (Zinkevich et al., 2007) showing a $O(1/\sqrt{t})$ convergence rate in terms of exploitability. Other techniques provide a faster rate of convergence $O(1/t)$ (Daskalakis et al., 2011; Syrgkanis et al., 2015; Abernethy et al., 2018; Farina et al., 2019). These techniques have been usually studied in the discrete time setting but has also been looked at in continuous time (Mertikopoulos et al., 2018).

Convergence of the last iterate. *Extragradient or optimistic mirror descent* methods have been proven to converge to a Nash equilibrium (Korpelevich, 1976; Mertikopoulos et al., 2019) with possibly an exponential rate in unconstrained spaces (Mokhtari et al., 2020). The most closely related extragradient method in this domain is optimistic multiplicative-weights-update (OMWU; Daskalakis & Panageas, 2019) which provides convergence guarantees to the Nash equilibrium of the last iterate. Another approach uses the Frank-Wolfe method to compute Nash equilibria in normal-form games (Gidel et al., 2016), although convergence is attained at the same rate as for fictitious play. A related algorithm introduced by Munos et al. (2020) for imperfect information games consists in each player doing a step of *mirror ascent against an improved opponent* (MAIO) for which exponential convergence of the last-iterate was proven (with a instance-dependent exponent). Other works include the *regularized Nash dynamics* (Perolat et al., 2021; 2022) under continuous-time dynamics, the Magnet Mirror Descent (Sokota et al., 2023) which are related to Online Mirror Descent (OMD) performed on the regularized game, and the MTPO algorithm of Shani et al. (2024) in the context of multi-turn LLMs. A thorough comparison between Nash-MD and OMD is given in the next section.

6. Analysis of a tabular algorithm: Nash-MD

For simplicity of notation we remove the dependence on the context x, thus policies $\pi \in \Delta(\mathcal{Y})$ are probability distributions over $\mathcal{Y}$. We now introduce an algorithm, called **Nash-MD**, which is a novel variant of mirror descent (Nemirovski & Yudin, 1983; Bubeck, 2015; Lattimore & Szepesvári, 2020) that makes use of a specific regularized policy π_t^μ which is a geometric mixture between the current policy π_t and the reference policy μ. We prove the convergence (in terms of KL distance) of the last iterate to the Nash equilibrium of $\mathcal{P}_\tau$.

The Nash-MD algorithm: Define the regularized policy π_t^μ as a geometric mixture between the current policy π_t and the reference policy μ:

$$\pi_t^\mu(y) \stackrel{\text{def}}{=} \frac{\pi_t(y)^{1-\eta_t\tau}\mu(y)^{\eta_t\tau}}{\sum_{y'}\pi_t(y')^{1-\eta_t\tau}\mu(y')^{\eta_t\tau}}, \tag{3}$$

where η_t is a learning rate. We define the **Nash-MD algorithm** as a step of mirror descent relative to the regularized policy π_t^μ:

$$\pi_{t+1} \stackrel{\text{def}}{=} \arg\max_\pi \left[\eta_t \mathcal{P}(\pi \succ \pi_t^\mu) - \mathrm{KL}(\pi, \pi_t^\mu)\right]. \tag{4}$$

The optimization above can also be made explicit in the following form:

$$\pi_{t+1}(y) \propto \pi_t^\mu(y)\exp\left(\eta_t \mathcal{P}(y \succ \pi_t^\mu)\right),$$

or equivalently

$$\begin{aligned}\log \pi_{t+1}(y) \quad &= \quad (1-\eta_t\tau)\log\pi_t(y) + \eta_t\tau\log\mu(y) \\ &\quad +\eta_t\mathcal{P}(y \succ \pi_t^\mu) + c,\end{aligned} \tag{5}$$

where c is a normalization constant that is independent of y.

The intuition for this algorithm is to improve the current policy π_t in a direction that increases the preference $\pi \mapsto \mathcal{P}(\pi, \pi_t^\mu)$ against the regularized policy π_t^μ, while not deviating too much from it. We now state our main theoretical result; see Appendix D for the proof.

Theorem 1. *Let π_τ^* be the Nash equilibrium of the regularized preference model: $\mathcal{P}_\tau(\pi \succ \pi') = \mathcal{P}(\pi \succ \pi') - \tau KL(\pi,\mu) + \tau KL(\pi',\mu)$. At every iteration t we have that*

$$KL(\pi_\tau^*, \pi_{t+1}) \le (1-\eta_t\tau)KL(\pi_\tau^*, \pi_t) + 2\eta_t^2. \tag{6}$$

We deduce that for the choice $\eta_t = 2/(\tau(t+2))$ we have

$$KL(\pi^*_\tau, \pi_T) \leq \frac{8}{\tau^2(T+1)}.$$

Thus this algorithm produces a sequence of policies $(\pi_t)_{1\leq t\leq T}$ with last-iterate convergence (in KL-divergence) to the regularized Nash equilibrium π^*_τ at a speed $O(1/T)$. We now mention several important features of this algorithm, specially in the context of LLMs.

Nash-MD does not require playing against the full mixture $\bar{\pi}_t$. In order to compute π_{t+1} we do not need to play against the mixture $\bar{\pi}_t = \frac{1}{t}\sum_{s=1}^t \pi_s$ of past policies (where by 'playing against a policy $\tilde{\pi}$' we mean computing (or estimating) the preference $\mathcal{P}(y \succ \tilde{\pi})$), unlike in fictitious play. We play against a single (geometric) mixture π_t^μ between the current policy π_t and the reference policy μ. This is important in situations, such as in LLMs, where storing and generating sample from several policies is costly.

Nash-MD has a last-iterate convergence property. The second important property of Nash-MD is that we have convergence of the last-iterate (i.e., the current policy π_t converges to π^*_τ) and not only convergence on average (as is typically the case of fictitious play and usual regret minimization algorithms like CFR and OMD). This feature is particularly important in the context of LLMs as well due to the substantial memory resources that would be otherwise needed to store a mixture policy like $\bar{\pi}_t$.

Comparison with online mirror descent (OMD). In general the analysis of constant-sum concave-convex games can be performed in the framework of online convex optimization where the goal is to find a sequence of solutions π_t that minimizes the sum of a sequence of convex loss functions $\pi \mapsto l_t(\pi)$. The OMD algorithm (using the KL as Bregman divergence) defines the sequence:

$$\pi_{t+1} \stackrel{\text{def}}{=} \arg\min_\pi \left[\eta_t \nabla l_t(\pi_t)\cdot(\pi - \pi_t) + \mathrm{KL}(\pi, \pi_t)\right], \tag{7}$$

for which it can be shown (see Cesa-Biachi & Lugosi (2006)) that the average cumulative regret, under optimal choice of learning rate, can be bounded as

$$\frac{1}{T}\sum_{t=1}^T l_t(\pi_t) - \min_\pi \frac{1}{T}\sum_{t=1}^T l_t(\pi) = O\left(1/\sqrt{T}\right).$$

This type of upper bound on the regret can be further used to obtain convergence of constant-sum games where each player follows an OMD strategy to minimize their own convex loss. In our context, we could apply this OMD strategy to minimize the regularized preference model $\mathcal{P}_\tau$, and since $\mathcal{P}_\tau$ is antisymmetric, we only need to consider the dynamics of a single player. So the loss function at time t is the negative preference against the current policy of the opponent: $l_t(\pi) = -\mathcal{P}_\tau(\pi \succ \pi_t)$. We deduce that

$\nabla l_t(\pi_t) = -\left[\partial_\pi \mathcal{P}_\tau(\pi \succ \pi_t)\right]_{\pi=\pi_t}$, thus

$\nabla l_t(\pi_t)\cdot\pi = -\sum_y \pi(y)\left[\mathcal{P}(y \succ \pi_t) - \tau\left(\log\frac{\pi_t(y)}{\mu(y)} + 1\right)\right]$.

The OMD update rule in Equation (7) can be rewritten as

$$\begin{aligned}\pi_{t+1} \quad = \quad & \arg\max_\pi \Big[\eta_t \sum_y \pi(y)\Big(\mathcal{P}(y \succ \pi_t) - \tau \log\frac{\pi_t(y)}{\mu(y)}\Big) \\ & -\mathrm{KL}(\pi, \pi_t)\Big].\end{aligned}$$

Now, using the regularized policy π_t^μ introduced in Equation (3), we can rewrite this update rule as

$$\pi_{t+1} = \arg\max_\pi \left[\eta_t \mathcal{P}(\pi \succ \pi_t) - \mathrm{KL}(\pi, \pi_t^\mu)\right]. \tag{8}$$

Comparing Equation (4) and Equation (8) we notice that both OMD and Nash-MD make use of the same KL penalty term $\mathrm{KL}(\pi, \pi_t^\mu)$. However they differ in the fact that OMD optimizes the preference $\pi \mapsto \mathcal{P}(\pi \succ \pi_t)$ against the current policy π_t whereas Nash-MD optimizes the preference $\pi \mapsto \mathcal{P}(\pi \succ \pi_t^\mu)$ against the regularized policy π_t^μ.

In the context of convex-concave games, the regret bound on the average cumulative regret translates into an upper bound on the exploitability of the game when players play their average policies, thus entailing their on-average convergence to the Nash equilibrium. However it is known that usual regret-minimization algorithms may not possess a last-iterate convergence property because the sequence of policies π_t may oscillate around the Nash equilibrium (see, for example, Mertikopoulos et al., 2018). Nevertheless, last-iterate convergence have been obtained for variants of OMD, such as extra-gradient and optimistic versions, see e.g., (Rakhlin & Sridharan, 2013; Daskalakis & Panageas, 2019; Mertikopoulos et al., 2019; Munos et al., 2020; Mokhtari et al., 2020). In the context of LLMs, the MTPO algorithm of Shani et al. (2024), defined by the same update rule (8), is shown to converge, in the last-iterate, to the regularized Nash equilibrium, but with a speed that depends on the inverse minimum non-zero probability of the reference policy $\mu_{\min} = \min_{y:\mu(y)>0}\mu(y)$. Also the on-line version of IPO (Calandriello et al., 2024) offers a deep-learning version of OMD similar to the update rule defined by Equation (8).

To the best of our knowledge, it appears that Nash-MD has not been introduced before, despite its simplicity. Nash-MD enjoys a last-iterate convergence property with a KL-divergence to the Nash equilibrium decaying as $O(1/T)$ (with a constant in the big O notation independent of $\mu_{\min}$). We believe the reason this simple modification of OMD possesses these nice properties is because of the special structure of the regularized preference function that we con-

sider here which is the sum of a bilinear function (in policy spaces) and a KL-penalty term.

Comparison with Ruppert-Polyak averaging. The weighted averaging of log-probabilities in Equation (5), along with the updates in OMD and other algorithms for regret minimisation and learning in games (Cesa-Biachi & Lugosi, 2006), also bear relation with Ruppert-Polyak averaging (Ruppert, 1988; Polyak, 1990; Polyak & Juditsky, 1992), a general technique in stochastic approximation in which iterates are averaged to accelerate convergence. However, the specific form of averaging and its uses within Nash-MD (over log-probabilities, and for use in opponent policies) are essential for shaping the dynamics of the algorithm and establishing convergence, not just as a method for acceleration.

The contextual bandit setting. All the results mentioned in this section are for the state-independent case, where policies and preferences do not depend on the context x. In the case of LLMs the context is the prompt x, and responses y and y' are generated conditioned on x. However the theoretical results do not change. Indeed, we would define the Nash-MD algorithm in the contextual bandit case as follows: for every $x \in \mathtt{supp}(\rho)$,

$$\pi_{t+1}(\cdot|x) \stackrel{\text{def}}{=} \arg\max_{\pi(\cdot)} \left[\eta_t \mathcal{P}(\pi(\cdot|x) \succ \pi_t^{\mu}(\cdot|x)|x) - \mathrm{KL}(\pi(\cdot), \pi_t^{\mu}(\cdot|x))\right],$$

where

$$\pi_t^{\mu}(y|x) \propto \pi_t(y|x)^{1-\eta_t\tau}\mu(y|x)^{\eta_t\tau}.$$

We prove the convergence of this algorithm, in exactly the same way as in Theorem 1, by showing that at every iteration t we have

$$\mathrm{KL}(\pi_\tau^*, \pi_{t+1}) \leq (1-\eta_t\tau)\mathrm{KL}(\pi_\tau^*, \pi_t) + 2\eta_t^2,$$

where $\mathrm{KL}(\pi, \pi') = \mathbb{E}_{x\sim\rho}[\mathrm{KL}(\pi(\cdot|x), \pi'(\cdot|x))]$.

7. Deep learning implementation of NLHF

We build upon the insights from Nash-MD and describe gradient-based algorithms **Nash-MD-PG** and **Nash-EMA-PG** for deep-learning architectures designed for the computation of the Nash equilibrium of a preference model, with a specific focus on their applicability in the context of LLMs. The general form of the policy gradient is

$$\nabla_\theta \log \pi_\theta(y|x) \left(\mathcal{P}(y \succ y'|x) - 1/2 - \tau \log \frac{\pi_\theta(y|x)}{\mu(y|x)}\right),$$

where the prompt $x \sim \rho$, the first responses $y \sim \pi_\theta(\cdot|x)$ and the second response $y' \sim \pi'(\cdot|x)$ where the *alternative policy* π' is either

- the (geometric) mixture policy $\pi'(y|x) \propto (\pi_\theta(y|x))^{1-\beta}(\mu(y|x))^{\beta}$ between π_θ and μ (for some mixture parameter $\beta \in [0,1]$), in the case of the **Nash-MD-PG** algorithm,
- the policy with a parameter being an exponentially moving average of past parameters, in the case of the **Nash-EMA-PG** algorithm.

Notice we have subtracted the baseline $1/2 = \mathcal{P}(y \succ y|x)$ from the preference $\mathcal{P}(y \succ y'|x)$ (which does not change the expectation of the gradient) as a variance reduction technique that does not require learning a value function. All the details of these algorithms are given in Appendix F.

8. Experiments on a text summarization task

In Appendix G we report experiments on a text summarization task and compare several algorithms for NLHF (Self-Play, Best-Response against μ, Nash-MD-PG and Nash-EMA-PG) as well as a RLHF baseline.

We made a pairwise evaluation of all the models by querying a very large LLM (PaLM 2 Large) (Anil et al., 2023) to obtain a preference signal, which is reported in Table 1. The models that are compared are respectively: SFT (Supervised Fined-Tuned, from which all other models are initialized and regularized to, i.e., this is also μ), RLHF, SP (Self-Play, equivalent to Nash-MD-PG with $\beta = 0$), MD1-MD6 (Nash-MD-PG with $\beta \in \{0.125, 0.25, 0.375, 0.5, 0.625, 0.75\}$), BR (Best Response against SFT, equivalent to Nash-MD-PG with $\beta = 1$), EMA1-2 (last-iterate of Nash-EMA-PG with $\beta \in \{0.999, 0.9995\}$), and EMA$1^* - 2^*$ (average policy of Nash-EMA-PG). See the full description in Appendix G.

The Nash-MD-PG models (specially for $\beta \in [0.125, 0.375]$) emerge as the best-performing method, surpassing the other models in this pairwise comparison.

The choice of the mixture parameter β in Nash-MD-PG entails an interesting trade-off (see the numbers highlighted in blue). A parameter value $\beta = 0$ corresponds to Self-Play, while a value $\beta = 1$ represents Best-Response against the initial policy (SFT). Notably, intermediate values within the range of 0.125 to 0.375 consistently outperform both Self-Play and Best-Response, highlighting the advantages of self-improvement when playing against a mixture policy (between self and a past version of self) as opposed to against a pure policy (either self or fixed).

Notice that it is difficult to establish a fair comparison between the NLHF and RLHF approaches since they rely on different models: a preference model for NLHF versus a reward model for RLHF, and the quality of the learnt models should be part of the picture for a full comparison between the NLHF and RLHF approaches. Thus the goal of these experiments is not to show the superiority of a method over

Table 1. PaLM 2 preference $\mathcal{P}^*(\pi_c \succ \pi_r)$ model between column policy π_c against row policy π_r.

$\mathcal{P}^*$	SFT	RLHF	SP	**MD1**	MD2	MD3	MD4	MD5	MD6	BR	EMA1	EMA2	EMA1*	EMA2*
SFT	0.500	0.990	0.983	**0.982**	0.989	0.987	0.985	0.982	0.965	0.943	0.970	0.961	0.977	0.980
RLHF	0.010	0.500	**0.489**	**0.598**	**0.519**	**0.561**	**0.501**	**0.436**	**0.284**	**0.148**	0.468	0.320	0.477	0.510
SP	0.017	0.511	0.500	**0.592**	0.504	0.545	0.499	0.451	0.310	0.211	0.445	0.362	0.464	0.488
MD1	0.018	0.402	0.408	**0.500**	0.425	0.470	0.369	0.362	0.238	0.163	0.391	0.270	0.400	0.447
MD2	0.011	0.481	0.496	**0.575**	0.500	0.513	0.491	0.434	0.298	0.196	0.460	0.351	0.430	0.496
MD3	0.013	0.439	0.455	**0.530**	0.487	0.500	0.484	0.408	0.273	0.187	0.429	0.323	0.413	0.472
MD4	0.015	0.499	0.501	**0.631**	0.509	0.516	0.500	0.428	0.265	0.161	0.468	0.358	0.437	0.503
MD5	0.018	0.564	0.549	**0.638**	0.566	0.592	0.572	0.500	0.329	0.210	0.532	0.389	0.518	0.539
MD6	0.035	0.716	0.690	**0.762**	0.702	0.727	0.735	0.671	0.500	0.342	0.652	0.548	0.651	0.691
BR	0.057	0.852	0.789	**0.837**	0.804	0.813	0.839	0.790	0.658	0.500	0.743	0.640	0.752	0.774
EMA1	0.030	0.532	0.555	**0.609**	0.540	0.571	0.532	0.468	0.348	0.257	0.500	0.381	0.480	0.556
EMA2	0.039	0.680	0.638	**0.730**	0.649	0.677	0.642	0.611	0.452	0.360	0.619	0.500	0.585	0.659
EMA1*	0.023	0.523	0.536	**0.600**	0.570	0.587	0.563	0.482	0.349	0.248	0.520	0.415	0.500	0.555
EMA2*	0.020	0.490	0.512	**0.553**	0.504	0.528	0.497	0.461	0.309	0.226	0.444	0.341	0.445	0.500

another one (this would also require a more intensive and larger scale empirical evaluation) but rather to illustrate how the proposed NLHF approach, and in particular the Nash-MD algorithm, can be implemented in a practical LLM setting.

This is also the reason we have not tried to over-optimize the hyper-parameters (such as the learning rate, the number of learning steps, etc) of the different methods (and we used the same parameters for all NLHF algorithms) and this could explains some differences observed in the pairwise preference Table compared with the results reported in (Calandriello et al., 2024), in which they optimize each algorithm with a different set of hyper-parameters.

9. Conclusion and future work

NLHF emerges as an interesting and promising alternative to RLHF, offering a fresh perspective on aligning models with human preferences. Given human pairwise preference data, learning a preference model is a more intuitive and natural approach compared to learning a reward model. It involves simpler techniques, such as supervised learning, and doesn't requires making specific assumptions, such as Bradley-Terry. Once a preference model is established, the concept of the Nash equilibrium naturally arises as a compelling solution concept.

For this new NLHF framework we have introduced Nash-MD, an algorithm that optimizes policies by following a self-improvement technique where the current model improves itself by playing (i.e., by generating and comparing responses) against a (geometric) mixture of the current model and a past model. The parameter β of the mixture ranges from $\beta = 0$ (which corresponds to playing against itself) to $\beta = 1$ (playing against a fixed policy) and may take any value in between.

In the case of tabular policy representations we have establish its last-iterate convergence to the Nash equilibrium of the regularized preference model. We have also introduced and implemented deep learning versions Nash-MD-PG and Nash-EMA-PG inspired by Nash-MD, and described how these ideas can be applied to LLMs by reported experimental results on a text-summarizing task.

Future research directions would consider the exploration of various mixtures between the current policy and past checkpoints, extending the concept initially introduced by Nash-MD. Additionally, another immediate direction would consider incorporating a decaying mixing coefficient $\beta \to 0$ to the deep Nash-MD variants to align more closely with theoretical considerations.

In conclusion, NLHF offers a compelling avenue for preference learning and policy optimization for aligning models with human preferences. An an example of a possible NLHF implementation we have introduced Nash-MD as a possible algorithmic solution and described some theoretical properties as well as deep learning adaptations. Further research in this direction, including the use of different mixture strategies, holds significant promise for advancing the field of aligning LLMs with human preferences.

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.

Acknowledgements

We would like to thank the individuals who designed and built the RL training infrastructure used in this paper: Eugene Tarassov, Léonard Hussenot, Johan Ferret, Robert Dadashi, Geoffrey Cideron, Alexis Jacq, Sabela Ramos, Piotr Stanczyk, Danila Sinopalnikov, Amélie Héliou, Ruba Haroun, Matt Hoffman, Bobak Shahriari, and in particular Olivier Pietquin for motivating discussions. We would like to express our gratitude to Ivo Danihelka, David Silver, Guillaume Desjardins, Tor Lattimore, and Csaba Szepesvári for their feedback on this work. Finally we would like to thank the anonymous reviewers who helped us improve the quality of this final version.

References

Abernethy, J., Lai, K. A., Levy, K. Y., and Wang, J.-K. Faster rates for convex-concave games. In *Proceedings of the Annual Conference on Learning Theory*, 2018.

Akrour, R., Schoenauer, M., and Sebag, M. Preference-based policy learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011.

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. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 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*, 2022.

Bertrand, Q., Czarnecki, W. M., and Gidel, G. On the limitations of the Elo: Real-world games are transitive, not additive. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2023.

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.

Brown, G. W. Iterative solution of games by fictitious play. *Act. Anal. Prod Allocation*, 13(1):374, 1951.

Bubeck, S. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4): 231–357, 2015.

Busa-Fekete, R., Szörenyi, B., Weng, P., Cheng, W., and Hüllermeier, E. Preference-based evolutionary direct policy search. In *Autonomous Learning Workshop at the IEEE International Conference on Robotics and Automation*, 2013.

Busa-Fekete, R., Szörényi, B., Weng, P., Cheng, W., and Hüllermeier, E. Preference-based reinforcement learning: Evolutionary direct policy search using a preference-based racing algorithm. *Machine Learning*, 97(3):327–351, 2014.

Busbridge, D., Ramapuram, J., Ablin, P., Likhomanenko, T., Dhekane, E. G., Suau, X., and Webb, R. How to scale your EMA. In *Advances in Neural Information Processing Systems*, 2023.

Calandriello, D., Guo, D., Munos, R., Rowland, M., Tang, Y., Pires, B. A., Richemond, P. H., Lan, C. L., Valko, M., Liu, T., Joshi, R., Zheng, Z., and Piot, B. Human alignment of large language models through online preference optimisation. *arXiv*, 2024.

Cesa-Biachi, N. and Lugosi, G. *Predition, Learning, and Games*. Cambridge University Press, 2006.

Chen, X., Zhong, H., Yang, Z., Wang, Z., and Wang, L. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *Proceedings of the International Conference on Machine Learning*, 2022.

Cheng, W., Fürnkranz, J., Hüllermeier, E., and Park, S.-H. Preference-based policy iteration: Leveraging preference learning for reinforcement learning. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2011.

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

Clopper, C. J. and Pearson, E. S. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika*, 26(4):404–413, 1934.

Csiszar, I. and Korner, J. *Information Theory: Coding Theorems for Discrete Memoryless Systems*. Academic Press, Inc., 1982.

Daskalakis, C. and Panageas, I. Last-iterate convergence: Zero-sum games and constrained min-max optimization. In *Proceedings of the Conference on Innovations in Theoretical Computer Science*, 2019.

Daskalakis, C., Deckelbaum, A., and Kim, A. Near-optimal no-regret algorithms for zero-sum games. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2011.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., SHUM, K., and Zhang, T. RAFT: Reward rAnked FineTuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023.

Efroni, Y., Merlis, N., and Mannor, S. Reinforcement learning with trajectory feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Elo, A. E. *The Rating of Chessplayers, Past and Present*. Arco Pub., 1978.

Farina, G., Kroer, C., and Sandholm, T. Optimistic regret minimization for extensive-form games via dilated distance-generating functions. In *Advances in Neural Information Processing Systems*, 2019.

Fudenberg, D. and Levine, D. K. *The Theory of Learning in Games*. MIT Press, 1998.

Gardner, M. The paradox of the nontransitive dice. *Scientific American*, (223):110–111, 1970.

Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized Markov decision processes. In *Proceedings of the International Conference on Machine Learning*, 2019.

Gidel, G., Jebara, T., and Lacoste-Julien, S. Frank-Wolfe algorithms for saddle point problems. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2016.

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.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, 2020.

Gulcehre, C., Paine, T. L., Srinivasan, S., Konyushkova, K., Weerts, L., Sharma, A., Siddhant, A., Ahern, A., Wang, M., Gu, C., Macherey, W., Doucet, A., Firat, O., and de Freitas, N. Reinforced self-training (ReST) for language modeling, 2023.

Heinrich, J. and Silver, D. Deep reinforcement learning from self-play in imperfect-information games. *arXiv*, 2016.

Heinrich, J., Lanctot, M., and Silver, D. Fictitious self-play in extensive-form games. In *Proceedings of the International Conference on Machine Learning*, 2015.

Hennes, D., Morrill, D., Omidshafiei, S., Munos, R., Perolat, J., Lanctot, M., Gruslys, A., Lespiau, J. B., Parmas, P., Duéñez-Guzmán, E., and Tuyls, K. Neural replicator dynamics: Multiagent learning via hedging policy gradients. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems*, 2020.

Hoda, S., Gilpin, A., Peña, J., and Sandholm, T. Smoothing techniques for computing Nash equilibria of sequential games. *Mathematics of Operations Research*, 35(2):494–512, 2010.

Hofbauer, J. and Sorin, S. Best response dynamics for continuous zero-sum games. *Discrete and Continuous Dynamical Systems Series B*, 6(1):215, 2006.

Jacob, A. P., Farina, G., and Andreas, J. Regularized conventions: Equilibrium computation as a model of pragmatic reasoning. *arXiv*, 2023a.

Jacob, A. P., Shen, Y., Farina, G., and Andreas, J. The consensus game: Language model generation via equilibrium search, 2023b.

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.

Kangarshahi, E. A., Hsieh, Y.-P., Sahin, M. F., and Cevher, V. Let's be honest: An optimal no-regret framework for zero-sum games. In *Proceedings of the International Conference on Machine Learning*, 2018.

Klimenko, A. Y. Intransitivity in theory and in the real world. *Entropy*, 17(6):4364–4412, 2015.

Korpelevich, G. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976.

Lattimore, T. and Szepesvári, C. *Bandit Algorithms*. Cambridge University Press, 2020.

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.

Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., and McAleese, N. Teaching language models to support answers with verified quotes. *arXiv*, 2022.

Mertikopoulos, P., Papadimitriou, C., and Piliouras, G. Cycles in adversarial regularized learning. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2018.

Mertikopoulos, P., Lecouat, B., Zenati, H., Foo, C., Chandrasekhar, V., and Piliouras, G. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *Proceedings of the International Conference on Learning Representations*, 2019.

Mokhtari, A., Ozdaglar, A., and Pattathil, S. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2020.

Moskovitz, T., Singh, A. K., Strouse, D., Sandholm, T., Salakhutdinov, R., Dragan, A. D., and McAleer, S. Confronting reward model overoptimization with constrained RLHF. *arXiv*, 2023.

Munos, R., Perolat, J., Lespiau, J.-B., Rowland, M., De Vylder, B., Lanctot, M., Timbers, F., Hennes, D., Omidshafiei, S., Gruslys, A., Azar, M. G., Lockhart, E., and Tuyls, K. Fast computation of Nash equilibria in imperfect information games. In *Proceedings of the International Conference on Machine Learning*, 2020.

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.

Nemirovski, A. and Yudin, D. *Problem complexity and method efficiency in optimization*. Wiley-Interscience Series in Discrete Mathematics, 1983.

Nesterov, Y. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1): 235–249, 2005.

Novoseller, E., Wei, Y., Sui, Y., Yue, Y., and Burdick, J. Dueling posterior sampling for preference-based reinforcement learning. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2020.

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

OpenAI. GPT-4 technical report. *arXiv*, 2023.

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.

Pacchiano, A., Saha, A., and Lee, J. Dueling RL: Reinforcement learning with trajectory preferences. *arXiv*, 2023.

Perolat, J., Munos, R., Lespiau, J.-B., Omidshafiei, S., Rowland, M., Ortega, P., Burch, N., Anthony, T., Balduzzi, D., De Vylder, B., Piliouras, G., Lanctot, M., and Tuyls, K. From Poincaré recurrence to convergence in imperfect

information games: Finding equilibrium via regularization. In *Proceedings of the International Conference on Machine Learning*, 2021.

Perolat, J., Vylder, B. D., Hennes, D., Tarassov, E., Strub, F., de Boer, V., Muller, P., Connor, J. T., Burch, N., Anthony, T., McAleer, S., Elie, R., Cen, S. H., Wang, Z., Gruslys, A., Malysheva, A., Khan, M., Ozair, S., Timbers, F., Pohlen, T., Eccles, T., Rowland, M., Lanctot, M., Lespiau, J.-B., Piot, B., Omidshafiei, S., Lockhart, E., Sifre, L., Beauguerlange, N., Munos, R., Silver, D., Singh, S., Hassabis, D., and Tuyls, K. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378(6623):990–996, 2022.

Polyak, B. T. New stochastic approximation type procedures. *Automat. i Telemekh*, 7(98-107):2, 1990.

Polyak, B. T. and Juditsky, A. B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30(4):838–855, 1992.

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.

Rakhlin, S. and Sridharan, K. Optimization, learning, and games with predictable sequences. In *Advances in Neural Information Processing Systems*, 2013.

Rame, A., Couairon, G., Shukor, M., Dancette, C., Gaya, J.-B., Soulier, L., and Cord, M. Rewarded soups: Towards Pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. In *Advances in Neural Information Processing Systems*, 2023.

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`. *Journal of Machine Learning Research*, 24(377):1–8, 2023.

Robinson, J. An iterative method of solving a game. *Annals of Mathematics*, 54(2):296–301, 1951.

Roit, P., Ferret, J., Shani, L., Aharoni, R., Cideron, G., Dadashi, R., Geist, M., Girgin, S., Hussenot, L., Keller, O., Momchev, N., Ramos, S., Stanczyk, P., Vieillard, N., Bachem, O., Elidan, G., Hassidim, A., Pietquin, O., and Szpektor, I. Factually consistent summarization via reinforcement learning with textual entailment feedback. In *Proceedings of the Annual Meeting of the Associating for Computational Linguistics*, 2023.

Rosen, J. B. Existence and uniqueness of equilibrium points for concave n-person games. *Econometrica: Journal of the Econometric Society*, pp. 520–534, 1965.

Ruppert, D. Efficient estimations from a slowly convergent Robbins–Monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

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

Shani, L., Rosenberg, A., Cassel, A., Lang, O., Calandriello, D., Zipori, A., Noga, H., Keller, O., Piot, B., Szpektor, I., Hassidim, A., Matias, Y., and Munos, R. Multi-turn reinforcement learning from preference human feedback. *arXiv*, 2024.

Sion, M. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958.

Sokota, S., D'Orazio, R., Kolter, J. Z., Loizou, N., Lanctot, M., Mitliagkas, I., Brown, N., and Kroer, C. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In *Proceedings of the International Conference on Learning Representations*, 2023.

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.

Syrgkanis, V., Agarwal, A., Luo, H., and Schapire, R. E. Fast convergence of regularized learning in games. In *Advances in Neural Information Processing Systems*, 2015.

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

Tversky, A. Intransitivity of preferences. *Psychological Review*, 76(1):31–48, 1969.

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.

von Neumann, J. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928.

Wang, Y., Liu, Q., and Jin, C. Is RLHF more difficult than standard RL? In *Advances in Neural Information Processing Systems*, 2023.

Wilson, A., Fern, A., and Tadepalli, P. A Bayesian approach for policy learning from trajectory preference queries. In *Advances in Neural Information Processing Systems*, 2012.

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

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.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. SLiC-HF: Sequence likelihood calibration with human feedback. In *Proceedings of the International Conference on Learning Representations*, 2023.

Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, 2007.

A. Maximizing expected Elo vs maximizing probability of winning

Consider the following preference model, where the set of actions is $\mathcal{Y} = \{y_1, y_2, y_3\}$ and the preference table between these actions is

$\mathcal{P}(y \succ y')$	$y = y_1$	$y = y_2$	$y = y_3$
$y' = y_1$	1/2	9/10	2/3
$y' = y_2$	1/10	1/2	2/11
$y' = y_3$	1/3	9/11	1/2

This preference model can be perfectly captured by a Bradley-Terry reward model in which the Elo score of the actions would be (up to an additive constant): $R(y_1) = 0$, $R(y_2) = \log 9$, and $R(y_3) = \log 2$.

If we optimize over the simplex $\Delta(\mathcal{Y})$, then the policy selecting deterministically y_2 is optimal both in terms of rewards and in terms of preference against any policy. However, if we consider a constrained optimization problem where we search for a policy in a subset $\mathcal{S} \subset \Delta(\mathcal{Y})$, then the optimum of the expected reward and preference may be different. To illustrate, let $\mathcal{S}$ be the set of probability distributions $\pi \in \Delta(\mathcal{Y})$ such that $\pi(y_1) = 2\pi(y_2)$.

In that case, the policy $\pi^*_R \stackrel{\text{def}}{=} (2/3, 1/3, 0)$ is optimal in terms of maximizing expected rewards whereas the policy $\pi^*_{\mathcal{P}} \stackrel{\text{def}}{=} (0, 0, 1)$ is optimal in terms of maximizing preference against any alternative policy in $\mathcal{S}$. In particular we have

$$\mathbb{E}_{y\sim\pi^*_R}[R(y)] = 0 \times 2/3 + \log(9) \times 1/3 > \log(2) = \mathbb{E}_{y\sim\pi^*_{\mathcal{P}}}[R(y)],$$

whereas policy $\pi^*_{\mathcal{P}}$ is preferred over π^*_R, since

$$\mathcal{P}(\pi^*_{\mathcal{P}} \succ \pi^*_R) = \mathcal{P}(y_3 \succ y_1) \times 2/3 + \mathcal{P}(y_3 \succ y_2) \times 1/3 = 50/99 > 1/2.$$

Thus if one searches for a policy in $\mathcal{S}$, then the optimum in terms of maximizing expected (Elo) reward and maximizing preference (probability of winning) are different.

Note that the constraint $\pi \in \mathcal{S}$ may be imposed in a soft way using regularization. Here for example we could implement a 2-step decisions process where in a first step one would choose the probability mass assigned to y_3, and in the second step, one would choose the remaining mass to allocate between y_1 and y_2. The second step may be constrained in a soft way by penalizing distributions (over y_1 and y_2) that are different from a reference distribution $\mu = (2/3, 1/3)$ by using a KL-regularization with a large τ coefficient. In this way the set of effective policies that would be considered would be close to $\mathcal{S}$.

This example illustrates the fact that in constrained (or regularized) optimization settings, maximizing Elo versus preference are different objectives, even in a setting where preferences can be perfectly expressed in a Bradley-Terry model.

B. Sensitivity of reward models w.r.t. the sampling distribution

Proposition 2. *For a given preference model $\mathcal{P}(y \succ y')$ and a distribution π, let us define the best Bradley-Terry reward model:*

$$r^\pi \stackrel{\text{def}}{=} \arg\max_r \mathbb{E}_{\substack{y,\, y' \sim \pi(\cdot) \\ Z \sim \nu}} \left[\log\left(\sigma(r(y^Z_w) - r(y^Z_l))\right)\right] \tag{9}$$

where y^Z_w and y^Z_l are respectively the preferred (and less preferred) response (among y and y' sampled from $\pi(\cdot)$) according to a randomly sampled human $Z \sim \nu$.

Define the corresponding Bradley-Terry preference model $\mathcal{P}^\pi_{BT}(y \succ y') \stackrel{\text{def}}{=} \sigma(r^\pi(y) - r^\pi(y'))$. Then we have that for any y in the support of π,

$$\mathcal{P}(y \succ \pi) = \mathcal{P}^\pi_{BT}(y \succ \pi).$$

Proof. First notice that, from the definition of y^Z_w and y^Z_l, we have

$$r^\pi = \arg\max_r \mathbb{E}_{y,y'\sim\pi}\left[\mathcal{P}(y \succ y') \log \sigma(r(y) - r(y'))\right].$$

Write $\mathcal{L}(r)$ the loss that is minimized. Thus $r^\pi = \arg\max_r \mathcal{L}(r)$, and

$$\begin{aligned}\nabla\mathcal{L}(r) &= \mathbb{E}_{y,y'\sim\pi}\left[\mathcal{P}(y\succ y')\frac{\nabla\sigma(r(y)-r(y'))}{\sigma(r(y)-r(y'))}\right]\\ &= \mathbb{E}_{y,y'\sim\pi}\left[\mathcal{P}(y\succ y')\sigma(r(y')-r(y))(\nabla r(y)-\nabla r(y'))\right]\end{aligned}$$

Thus looking a the z-th element of $\nabla\mathcal{L}(r^\pi)$:

$$\begin{aligned}\partial_{r(z)}\mathcal{L}(r^\pi) &= \pi(z)\mathbb{E}_{y\sim\pi}\left[\mathcal{P}(z\succ y)\sigma(r^\pi(y)-r^\pi(z))-(1-\mathcal{P}(z\succ y)(1-\sigma(r^\pi(y)-r^\pi(z)))\right]\\ &= \pi(z)\mathbb{E}_{y\sim\pi}\left[\mathcal{P}(z\succ y)+\mathcal{P}^\pi_{BT}(y\succ z)-1\right]\\ &= \pi(z)\left[\mathcal{P}(z\succ\pi)-\mathcal{P}^\pi_{BT}(z\succ\pi)\right].\end{aligned}$$

Setting the derivative $\partial_{r(z)}\mathcal{L}(r^\pi)=0$ we deduce that for z in the support of π we have $\mathcal{P}(z\succ\pi)=\mathcal{P}^\pi_{BT}(z\succ\pi)$. □

Theorem 2 (The optimal BT-reward model depends on the sampling distribution)**.** *If a preference model $\mathcal{P}$ cannot be perfectly captured by a Bradley-Terry reward model, in the sense that the preference model $\mathcal{P}^\pi_{BT}(y\succ y')\stackrel{\text{def}}{=}\sigma\left(r^\pi(y)-r^\pi(y')\right)$ corresponding to the best Bradley-Terry reward model r^π, solution to Equation (9) for some policy π, is not identical to $\mathcal{P}$, then the reward model r^π depends explicitly on the sampling distribution π.*

More precisely, if there exists y, y' and π such that $\mathcal{P}^\pi_{BT}(y\succ y')\neq\mathcal{P}(y\succ y')$, then there exists another policy $\pi'\neq\pi$ (with same support as π) such that $r^\pi(y)-r^\pi(y')\neq r^{\pi'}(y)-r^{\pi'}(y')$. Thus we also have that $\mathcal{P}^\pi_{BT}(y\succ y')\neq\mathcal{P}^{\pi'}_{BT}(y\succ y')$.

This result shows that the reward model r^π (thus the corresponding Bradley-Terry preference model $\mathcal{P}^\pi_{BT}$) depends on the sampling distribution π (which explains our use of π as superscript).

Proof. Assume there exists y, y' and π such that $\mathcal{P}^\pi_{BT}(y\succ y')\neq\mathcal{P}(y\succ y')$. Let us define π' (with same support as π) as follows: $\pi'(z)=\frac{1}{2}\pi(z)$ for $z\neq y'$, and $\pi'(y')=c\pi(y')$ for some constant $c>1$ (defined such that π' is a proper probability distribution). We deduce that

$$\mathcal{Q}(y\succ\pi')=\frac{1}{2}\mathcal{Q}(y\succ\pi)+(c-1/2)\pi(y')\mathcal{Q}(y\succ y'),$$

for any preference model $\mathcal{Q}$. Applying this equality both with $\mathcal{Q}=\mathcal{P}$ and $\mathcal{Q}=\mathcal{P}^\pi_{BT}$, and since $\mathcal{P}^\pi_{BT}(y\succ y')\neq\mathcal{P}(y\succ y')$ and $\mathcal{P}^\pi_{BT}(y\succ\pi)=\mathcal{P}(y\succ\pi)$ (from Proposition 2), we deduce that $\mathcal{P}^\pi_{BT}(y\succ\pi')\neq\mathcal{P}(y\succ\pi')$. Applying Proposition 2 again we have that $\mathcal{P}(y\succ\pi')=\mathcal{P}^{\pi'}_{BT}(y\succ\pi')$, thus

$$\mathcal{P}^\pi_{BT}(y\succ\pi')\neq\mathcal{P}^{\pi'}_{BT}(y\succ\pi').$$

We deduce that

$$\sum_z\pi'(z)\left[\sigma\big(r^\pi(y)-r^\pi(z)\big)-\sigma\big(r^{\pi'}(y)-r^{\pi'}(z)\big)\right]\neq 0,$$

thus there exists (at lease one) z such that $r^\pi(y)-r^\pi(z)\neq r^{\pi'}(y)-r^{\pi'}(z)$. This concludes the proof that the two BT-reward models r^π and $r^{\pi'}$ are different as well as the corresponding BT-preference models $\mathcal{P}^\pi_{BT}$ and $\mathcal{P}^{\pi'}_{BT}$. □

C. Preference models may be non-transitive

C.1. Example of a non-transitive preference model

Notice that in general a preference model may not be transitive. Here is a simple illustration of a non-transitive preference model where we have 3 policies π_1, π_2 and π_3 such that $\mathcal{P}(\pi_1\succ\pi_2)>1/2$, $\mathcal{P}(\pi_2\succ\pi_3)>1/2$ and $\mathcal{P}(\pi_3\succ\pi_1)>1/2$.

We consider the set of outcomes being the subset of integers $\mathcal{Y}=\{1,2,\ldots,9\}$ and the 3 policies being defined by $\pi_1=\mathcal{U}(\{2,4,9\})$, $\pi_2=\mathcal{U}(\{1,6,8\})$, and $\pi_3=\mathcal{U}(\{3,5,7\})$, where $\mathcal{U}(S)$ refers to a uniform distribution over the set S. The preference is defined as $\mathcal{P}(\pi\succ\pi')=\mathbb{E}_{y\sim\pi,y'\sim\pi'}[\mathbb{I}\{y\geq y'\}]$. Then we have

$$\mathcal{P}(\pi_1\succ\pi_2)=\mathcal{P}(\pi_2\succ\pi_3)=\mathcal{P}(\pi_3\succ\pi_1)=5/9.$$

This mirrors the classical example of non-transitive dice (Gardner, 1970).

C.2. Non-transitive aggregation of individual transitive preferences

Here we show that even if each human has transitive individual preferences, the resulting average preference model may not be transitive. Let us consider a specific case of a preference model defined as the probability (under some random outcome Z) that $f(x, y, Z) \geq f(x, y', Z)$, where f is a (deterministic) absolute scoring function:

$$\mathcal{P}(y \succ y'|x) = \mathbb{E}_{Z\sim\nu}\left[\mathbb{I}\{f(x, y, Z) \succ f(x, y', Z)\}\right],$$

where we define the function $\mathbb{I}\{u \succ v\} \stackrel{\text{def}}{=} (\text{sign}(u - v) + 1)/2$, which behaves as an indicator for the event $u > v$, and assigning a value of $1/2$ in the case where $u = v$. For example, this could represent the probability that a randomly selected human $Z \sim \nu$ prefers choice y over choice y' in a context x.

Consider the following example, where there are 3 possible responses y_1, y_2, y_3 and 3 possible humans z_1, z_2, z_3 chosen uniformly at random: $\nu = \mathcal{U}(\{z_1, z_2, z_3\})$. Define the scoring function f as follows:

$$\begin{aligned} &f(y_1, z_1) = 2, \quad f(y_1, z_2) = 4, \quad f(y_1, z_3) = 9,\\ &f(y_2, z_1) = 1, \quad f(y_2, z_2) = 6, \quad f(y_2, z_3) = 8,\\ &f(y_3, z_1) = 3, \quad f(y_3, z_2) = 5, \quad f(y_3, z_3) = 7. \end{aligned}$$

Notice that this defines a transitive preference model for each individual human $z \in \{z_1, z_2, z_3\}$. However, when aggregated the preference model satisfies $\mathcal{P}(y_1 \succ y_2) = \mathcal{P}(y_2 \succ y_3) = \mathcal{P}(y_3 \succ y_1) = 2/3$. This example thus illustrates that even if for each individual, preferences are totally ordered, when averaged over humans, the resulting preferences model may be non-transitive.

D. Proof of Theorem 1

We start with a first lemma.

Lemma 1. *For any π, and $0 \leq \eta_t\tau \leq 1$, we have*

$$KL(\pi, \pi_t^\mu) \leq \eta_t\tau KL(\pi, \mu) + (1 - \eta_t\tau)KL(\pi, \pi_t) - \eta_t\tau KL(\pi_t^\mu, \mu).$$

Proof. From the definition of π_t^μ, we have

$$\log \pi_t^\mu(y) = (1 - \eta_t\tau)\log \pi_t(y) + \eta_t\tau \log \mu(y) - \log Z,$$

where we define $Z = \sum_{y'}(\pi_t(y'))^{1-\eta_t\tau}(\mu(y'))^{\eta_t\tau}$. Thus, for any π, we have

$$\text{KL}(\pi, \pi_t^\mu) = \eta_t\tau\text{KL}(\pi, \mu) + (1 - \eta_t\tau)\text{KL}(\pi, \pi_t) + \log Z.$$

We have that

$$\begin{aligned} \eta_t\tau\text{KL}(\pi_t^\mu, \mu) &= \eta_t\tau \sum_y \pi_t^\mu(y) \log \frac{\big(\pi_t(y)\big)^{1-\eta_t\tau}\big(\mu(y)\big)^{\eta_t\tau}}{Z\mu(y)}\\ &= (1 - \eta_t\tau)\sum_y \pi_t^\mu(y) \log \frac{\big(\pi_t(y)\big)^{\eta_t\tau}}{\big(\mu(y)\big)^{\eta_t\tau}} - \eta_t\tau \log Z\\ &\leq (1 - \eta_t\tau)\log \sum_y \pi_t^\mu(y) \frac{\big(\pi_t(y)\big)^{\eta_t\tau}}{\big(\mu(y)\big)^{\eta_t\tau}} - \eta_t\tau \log Z\\ &= (1 - \eta_t\tau)\log \sum_y \frac{\big(\pi_t(y)\big)^{1-\eta_t\tau}\big(\mu(y)\big)^{\eta_t\tau}}{Z} \frac{\big(\pi_t(y)\big)^{\eta_t\tau}}{\big(\mu(y)\big)^{\eta_t\tau}} - \eta_t\tau \log Z\\ &= -\log Z, \end{aligned}$$

where we used Jensen's inequality applied with the concave logarithmic function. We deduce

$$\text{KL}(\pi, \pi_t^\mu) \leq \eta_t\tau\text{KL}(\pi, \mu) + (1 - \eta_t\tau)\text{KL}(\pi, \pi_t) - \eta_t\tau\text{KL}(\pi_t^\mu, \mu).$$

□

Now we use Lemma 7 of (Munos et al., 2020), restated below with notation.

Lemma 2. *Let $p \geq 1$ and $q \geq 1$ such that $1/p + 1/q = 1$. Let φ be a strongly convex function with respect to the ℓ_p-norm $\|\cdot\|_p$ with some modulus σ, i.e., for any π, π',*

$$\varphi(\pi) \geq \varphi(\pi') + \nabla\varphi(\pi') \cdot (\pi - \pi') + \frac{\sigma}{2}\|\pi - \pi'\|^2.$$

Write D_φ the associated Bregman divergence: for π, π',

$$D_\varphi(\pi, \pi') \stackrel{\text{def}}{=} \varphi(\pi) - \varphi(\pi') - \nabla\varphi(\pi') \cdot (\pi - \pi').$$

Let δ be a vector of dimension $|\mathcal{Y}|$. For any $\pi^- \in \Delta(\mathcal{Y})$, define π^+ as

$$\pi^+ = \arg\max_{\pi \in \Delta(\mathcal{Y})} \left[\sum_y \pi(y)\delta(y) - D_\varphi(\pi, \pi^-) \right], \tag{10}$$

Then for any $\pi \in \Delta(\mathcal{Y})$, we have,

$$D_\varphi(\pi, \pi^+) \leq D_\varphi(\pi, \pi^-) + \sum_y (\pi^-(y) - \pi(y))\delta(y) + (2/\sigma)\|\delta\|_q^2.$$

We apply this lemma with $\pi^+ = \pi_{t+1}$ and $\pi^- = \pi_t^\mu$, with the vector $\delta(y) = \eta_t \mathcal{P}(y \succ \pi_t^\mu)$, and as Bregman divergence D_φ we choose the KL (which corresponds to the choice of the entropy regularizer $\varphi(\pi) = \sum_y \pi(y) \log \pi(y)$). For $p = 1$, $q = \infty$, the regularizer φ is a strongly convex function with respect to the ℓ_1-norm with a modulus $\sigma = 1$; this is a consequence of Pinsker's inequality, see (Csiszar & Korner, 1982).

We deduce that for any π,

$$\text{KL}(\pi, \pi_{t+1}) \leq \text{KL}(\pi, \pi_t^\mu) + \eta_t \sum_y (\pi_t^\mu(y) - \pi(y))\mathcal{P}(y \succ \pi_t^\mu) + 2\eta_t^2. \tag{11}$$

For the choice $\pi = \pi_\tau^*$ and using the previous lemma, we have

$$\begin{aligned}
\text{KL}(\pi_\tau^*, \pi_{t+1}) &\leq \text{KL}(\pi_\tau^*, \pi_t^\mu) + \eta_t \sum_y (\pi_t^\mu(y) - \pi_\tau^*(y))\mathcal{P}(y \succ \pi_t^\mu) + 2\eta_t^2 \\
&\leq (1 - \eta_t\tau)\text{KL}(\pi_\tau^*, \pi_t) + \eta_t\tau\left(\text{KL}(\pi_\tau^*, \mu) - \text{KL}(\pi_t^\mu, \mu)\right) \\
&\quad + \eta_t\left(\mathcal{P}(\pi_t^\mu \succ \pi_t^\mu) - \mathcal{P}(\pi_\tau^* \succ \pi_t^\mu)\right) + 2\eta_t^2 \\
&= (1 - \eta_t\tau)\text{KL}(\pi_\tau^*, \pi_t) + \eta_t\left[1/2 - \mathcal{P}(\pi_\tau^* \succ \pi_t^\mu) + \tau\text{KL}(\pi_\tau^*, \mu) - \tau\text{KL}(\pi_t^\mu, \mu)\right] + 2\eta_t^2 \\
&= (1 - \eta_t\tau)\text{KL}(\pi_\tau^*, \pi_t) + \eta_t\left[1/2 - \mathcal{P}_\tau(\pi_\tau^* \succ \pi_t^\mu)\right] + 2\eta_t^2 \\
&\leq (1 - \eta_t\tau)\text{KL}(\pi_\tau^*, \pi_t) + 2\eta_t^2,
\end{aligned}$$

where the last inequality comes from the fact that π_τ^* is the Nash of the regularized game $\mathcal{P}_\tau$: $\mathcal{P}_\tau(\pi_\tau^* \succ \pi_t^\mu) \geq \mathcal{P}_\tau(\pi_\tau^* \succ \pi_\tau^*) = 1/2$ and the last equality comes from the definition of the regularized preference.

This inequality with $\eta_t = 2/(\tau(t+2))$ applied to $t = 0$ gives

$$\text{KL}(\pi_\tau^*, \pi_1) \leq \frac{2}{\tau^2}.$$

Then, by induction, assuming $\text{KL}(\pi_\tau^*, \pi_t) \leq \frac{8}{\tau^2(t+1)}$,

$$\begin{aligned}
\text{KL}(\pi_\tau^*, \pi_{t+1}) &\leq \left(1 - \frac{2}{t+2}\right)\frac{8}{\tau^2(t+1)} + \frac{8}{\tau^2(t+2)^2} \\
&\leq \left(1 - \frac{2}{t+2} + \frac{1}{t+2}\right)\frac{8}{\tau^2(t+1)} \\
&= \frac{8}{\tau^2(t+2)}.
\end{aligned}$$

E. Proof of Proposition 1

The mappings $\pi \mapsto \mathcal{P}(\pi \succ \pi')$ and $\pi' \mapsto \mathcal{P}(\pi \succ \pi')$ are linear in π (respectively in π') thus $\pi \mapsto \mathcal{P}_\tau(\pi \succ \pi')$ is concave and $\pi' \mapsto \mathcal{P}_\tau(\pi \succ \pi')$ is convex. Existence of a Nash equilibrium is derived from the minimax theorem for convex-concave functions (Sion, 1958).

The uniqueness of the Nash equilibrium essentially relies on the strict convexity/concavity of these mappings. We now give a proof of existence and uniqueness using variational inequalities.

We first note that since $\mathcal{P}_\tau(\pi' \succ \pi) = 1 - \mathcal{P}_\tau(\pi \succ \pi')$ we can re-express the minimax game of Eq. 2 as an antisymmetric two-player game with payoffs of policy π and π' are defined as

$$R(\pi; \pi') = \mathcal{P}(\pi \succ \pi') - \tau \mathrm{KL}_\rho(\pi, \mu)$$

and

$$R(\pi'; \pi) = \mathcal{P}(\pi' \succ \pi) - \tau \mathrm{KL}_\rho(\pi', \mu)$$

respectively. First we notice that since the payoff of this game is concave in π and π', it possesses a Nash equilibrium (Rosen, 1965, Theorem 1).

To show that this game has unique Nash equilibrium we need to show that its corresponding variational inequality is strictly monotone (Rosen, 1965, Theorem 2). Let $\bar{\pi} = [\pi, \pi']$ and $v(\bar{\pi}) = [\nabla_\pi R(\pi; \pi'), \nabla_{\pi'} R(\pi'; \pi)]$. Then every Nash equilibrium of the game should satisfy the following variational inequality for all $\bar{\pi}$:

$$v^T(\bar{\pi}^*)(\bar{\pi}^* - \bar{\pi}) \leq 0$$

Furthermore the variational inequality is strictly monotone if and only if for every $\bar{\pi}_1$ and $\bar{\pi}_2$ we have that

$$(v(\bar{\pi}_1) - v(\bar{\pi}_2))^T(\bar{\pi}_1 - \bar{\pi}_2) \leq 0 \tag{12}$$

with equality only holds at $\bar{\pi}_1 = \bar{\pi}_2$ (Rosen, 1965, Theorem 2). We can show this inequality holds by expanding the terms on LHS. For every context x let denote $v(\bar{\pi})(x)$ as the partial derivative $v(\bar{\pi})$ for x. We have:

$$v(\bar{\pi})(x) = \rho(x)[\mathbf{P}(y \succ \pi'|x) - \tau \log(\pi/\mu|x) - 1, \mathbf{P}(y \succ \pi|x) - \tau \log(\pi'/\mu|x) - 1],$$

where $\mathbf{P}(y \succ \pi'|x) = [p(y_i \succ \pi|x)]_{i=1:N}$ and $\log(\pi/\mu|x) = [\log(\pi(y_i|x)/\mu(y_i|x))]_{i=1:N}$, in which N is the size of the generation set. Plugging this in the LHS of Eq.12 and then exploiting the non-negativity of KL-divergence implies:

$$\begin{aligned}
(v(\bar{\pi}_1) - v(\bar{\pi}_2))^T(\bar{\pi}_1 - \bar{\pi}_2) = & \underbrace{\mathcal{P}(\pi_1 \succ \pi_1') + \mathcal{P}(\pi_1' \succ \pi_1) + \mathcal{P}(\pi_2 \succ \pi_2') + \mathcal{P}(\pi_2' \succ \pi_2)}_{=2} \\
& - \underbrace{(\mathcal{P}(\pi_2 \succ \pi_1') + \mathcal{P}(\pi_1' \succ \pi_2) + \mathcal{P}(\pi_1 \succ \pi_2') + \mathcal{P}(\pi_2' \succ \pi_1))}_{=2} \\
& - \tau(\mathrm{KL}_\rho(\pi_1||\pi_2) + \mathrm{KL}_\rho(\pi_2||\pi_1) + \mathrm{KL}_\rho(\pi_1'||\pi_2') + \mathrm{KL}_\rho(\pi_2'||\pi_1')) \\
= & -\tau(\mathrm{KL}_\rho(\pi_1||\pi_2) + \mathrm{KL}_\rho(\pi_2||\pi_1) + \mathrm{KL}_\rho(\pi_1'||\pi_2') + \mathrm{KL}_\rho(\pi_2'||\pi_1')) \leq 0
\end{aligned}$$

with equality only at $\bar{\pi}_1 = \bar{\pi}_2$.

F. Deep Learning Implementation of NLHF

Now, building upon the insights from Nash-MD, we explore potential gradient-based algorithms for deep-learning architectures designed for the computation of the Nash equilibrium of a preference model, with a specific focus on their applicability in the context of LLMs.

F.1. Generating one token at the time, instead of a full sequence

In LLMs it is usually the case that tokens are generated one at a time in an autoregressive manner. Thus the response $y \sim \pi(\cdot|x)$ can be written as $y = y_{0:N}$ (where $y_{0:N} \stackrel{\text{def}}{=} (y_0, \ldots, y_N)$), where each token y_n is generated from a distribution $\pi(\cdot|x, y_{0:n-1})$ conditioned on previous tokens, such that $\pi(y_{0:N}|x) = \prod_{n=0}^{N} \pi(y_n|x, y_{0:n-1})$. In practice (see the experiments section for results on LLMs) we will implement this token-per-token autoregressive generation of responses $y \sim \pi(\cdot|x)$ using next token distributions (implemented as a softmax over logits).

Now consider a parametric policy π_θ. Nash-MD requires the generation of alternative responses $y' \sim \pi_\theta^\beta(y|x)$ sampled from the regularized policy $\pi_\theta^\beta(y|x) \propto (\pi_\theta(y|x))^{1-\beta}(\mu(y|x))^\beta$ which is defined like in Equation (3) as a geometric mixture between the current policy π_θ and the reference policy μ. However it is not easy to generate a sequence y from this distribution by sampling one token y_n at a time. In particular, since π_θ^β is not a simple (arithmetic) mixture, we cannot select one policy π_θ or μ according to some prior probability (that would depend on the mixing parameter β) and then generate a sequence of tokens following that policy. Additionally, defining the normalization constant c as in Equation (5) for the full mixture π_θ^β is computationally prohibitive given the large number of possible sequences; instead, we would like to proceed by generating a token at a time. The approach we follow in our experiments consists in generating a token y_n from the marginal (geometric) mixture $\tilde{\pi}_\theta^\beta(\cdot|x, y_{0:n-1})$ defined by

$$\log \tilde{\pi}_\theta^\beta(y_n|x, y_{0:n-1}) = (1-\beta)\log \pi_\theta(y_n|x, y_{0:n-1}) + \beta \log \mu(y_n|x, y_{0:n-1}) + C(x, y_{0:n-1}),$$

where the normalization constant C depends on $x, y_{0:n-1}$. In order to sample from this marginal geometric mixture over the nth token, we evaluate the corresponding logits of both the current policy π_θ and the reference policy μ (conditioned on $(x, y_{0:n-1})$), we compute their (β-arithmetic) mixture, and sample a next token y_n from the corresponding softmax distribution. We call this corresponding product of marginal (geometric) mixtures over individual tokens the **one-step-at-a-time regularized policy**

$$\tilde{\pi}_\theta^\beta(y|x) \stackrel{\text{def}}{=} \prod_{n=0}^{N} \tilde{\pi}_\theta^\beta(y_n|x, y_{0:n-1}).$$

Notice that the one-step-at-a-time regularized policy $\tilde{\pi}_\theta^\beta(y|x)$ is different from the original regularized policy $\pi_\theta^\beta(y|x)$ because the sequence of normalization constants $C(x, y_{0:n-1})$ depend on the specific sample path $y_{0:n-1}$ and does not necessarily correspond to the full normalization constant c defined in Equation (5). We leave the analysis of the difference between these two policies for future work.

F.2. Computing the Nash equilibrium using regularized policy gradient

Our general algorithm for computing the Nash equilibrium of the preference model consists in repeating these steps:

- We randomly select a prompt $x \sim \rho$.
- We generate two responses y and y' (in an autoregressive fashion in the case of LLMs):
  - the first one $y \sim \pi_\theta(\cdot|x)$ by following the *current policy* π_θ that is being optimized;
  - the second one $y' \sim \pi'(\cdot|x)$ by following an *alternative policy* π'.

  The choice of the alternative policy π' that we use for the second generated sample y' depends on the specific algorithm we consider (the description of which is given in the next subsection).
- We update the parameter θ of the policy π_θ in the direction of the gradient $\nabla_\theta \mathcal{P}_\tau(\pi_\theta \succ \pi')$ of the regularized preference model $\mathcal{P}_\tau$.

We consider two cases, depending on whether a preference model is learnt or not.

$\mathcal{P}$-model-based approach. If we have learnt a preference model $\mathcal{P}$ (see Section G.1 for example for how one can learn a preference model) we query it to get the preference reward $\mathcal{P}(y \succ y'|x)$ and update θ by moving it in the direction of the policy gradient estimate

$$\widehat{g}(x, y, y') \stackrel{\text{def}}{=} \nabla_\theta \log \pi_\theta(y|x) \left(\mathcal{P}(y \succ y'|x) - 1/2 - \tau \log \frac{\pi_\theta(y|x)}{\mu(y|x)} \right). \tag{13}$$

Notice we have subtracted the baseline $1/2 = \mathcal{P}(y \succ y|x)$ from the preference $\mathcal{P}(y \succ y'|x)$ (which does not change the expectation of the gradient) as a variance reduction technique that does not require learning a value function as baseline. In practice, when the response y comprises a sequence of tokens $y_{0:N}$, a sample-based estimator to the KL based on the sample response y can be used. Further, this can be decomposed into a sum across token indicies of per-token KL estimators, and the standard policy-gradient variance-reduction trick of only multiplying $\nabla_\theta \log \pi_\theta(y_n|x, y_{0:n-1})$ by KL estimator terms corresponding to indices at least as great as n can be applied.

$\mathcal{P}$-model-free approach. In the case the preference model $\mathcal{P}(y \succ y'|x)$ comes directly from human preferences: $\mathcal{P}(y \succ y'|x) = \mathbb{P}_{Z\sim\nu}(\text{Human } Z \text{ prefers } y \text{ over } y' \text{ given } x)$, where ν is a distribution over humans, and if humans are immediately available to express their preference between any two responses, we can directly estimate the gradient by replacing $\mathcal{P}(y \succ y'|x)$ with $\mathbb{I}\{\text{Human } Z \text{ prefers } y \text{ over } y' \text{ given } x\}$ in Equation (13). This estimate does not require to learn a preference model first and is thus not affected by possible bias coming from an approximate model. Implementation-wise it requires having access to humans preference immediately after having generated the responses y and y'.

In both model-based and model-free approaches, we have that

$$\nabla_\theta \mathcal{P}_\tau(\pi_\theta \succ \underline{\pi'}) = \mathbb{E}_{x\sim\rho, \left\{ \substack{y\sim\pi_\theta(\cdot|x) \\ y'\sim\pi'(\cdot|x)} \right.} \left[\widehat{g}(x, y, y')\right], \tag{14}$$

(where $\underline{\pi'}$ denotes a stop-gradient on π' in the case π' would depend on θ).

F.3. Choice of the alternative policy π'

Now, for the choice of alternative policies π' that are used to generate the second sample y', we will consider two different algorithms **Nash-MD-PG** and **Nash-EMA-PG**, that are inspired by, respectively, the mirror-ascent algorithm Nash-MD introduced in the previous section, and a generalization of fictitious play where we consider an exponential moving average.

Nash-MD-PG. We define the alternative policy $\pi' = \pi_\theta^\beta$ as a geometric-mixture between π_θ and μ in a similar way as the regularized policy is defined in Equation (3):

$$\log \pi_\theta^\beta(y|x) \stackrel{\text{def}}{=} (1-\beta)\log(\pi_\theta(y|x)) + \beta \log(\mu(y|x)) + c(x), \tag{15}$$

where $\beta \in [0, 1]$ is the parameter of the mixture, and $c(x)$ is a constant independent of y. This is inspired by the Nash-MD algorithm described in Section 6, which we have proven to be convergent in Theorem 1. In the case of sequential generation of tokens in LLMs, we apply the one-step-at-a-time version $\tilde{\pi}_\theta^\beta$ of this regularized policy π_θ^β as defined in Subsection F.1. However, the corresponding PG version outlined in Subsection F.2 differs from Nash-MD as defined in Section 6 in a number of ways.

In addition to using a parametric representation of policies instead of a tabular one, it differs from the fact that it is not directly implementing a mirror descent algorithm but a simple gradient descent on the regularized preference model. In a sense this algorithm is only making a gradient step for the inner optimization problem of Equation (4), whereas a more faithful variant of Nash-MD would use a two-time scale algorithm and perform several gradient steps (while keeping π_θ and π_θ^β fixed) until the inner loop has reached an optimum, before updating π_θ and π_θ^β. Another apparent difference is that Nash-MD uses a KL-regularization w.r.t. the mixture policy π_θ^β, whereas Nash-MD-PG uses a KL w.r.t. the reference policy μ. However, we have that

$$\text{KL}(\pi_\theta, \pi_\theta^\beta) = (1-\beta)\text{KL}(\pi_\theta, \pi_\theta) + \beta\text{KL}(\pi_\theta, \mu) - \mathbb{E}_{x\sim\rho}[c(x)] = \beta\text{KL}(\pi_\theta, \mu) - \mathbb{E}_{x\sim\rho}[c(x)],$$

where $c(x)$ is the normalizing constant in Equation (15). Thus, we have

$$\nabla_\theta \text{KL}(\pi_\theta, \underline{\pi_\theta^\beta}) = \beta \nabla_\theta \text{KL}(\pi_\theta, \mu),$$

and since we perform a single step of gradient descent before updating π_θ, regularizing with respect to the mixture π_θ^β (in Nash-MD) is equivalent to regularizing w.r.t. μ (in Nash-MD-PG). Further, we use an additional parameter β (to define the mixture) that can be further tuned independently of τ.

Thus, while it is possible to implement Nash-MD more faithfully, such as by incorporating two-timescale policy gradient versions or exploring variants of regularized policy gradient methods such as PPO (Schulman et al., 2017) or NeuRD

(Hennes et al., 2020), we contend that the essence of Nash-MD is encapsulated in Nash-MD-PG for the following reason: the policy gradient algorithm Equation (14) improves the current policy π_θ by playing against the geometric mixture π_θ^β while preserving regularization with respect to π_θ^β.

Extreme cases for $\beta \in [0,1]$. Consider the alternative policy π_θ^β of Nash-MD-PG when $\beta \in [0,1]$ takes its extreme possible values: $\beta = 0$ or 1. When $\beta = 0$ then $\pi_\theta^{\beta=0} = \pi_\theta$, thus the alternative policy is the current policy, and this algorithm is simply a version of **self-play** (SP) where one improves its policy by playing against oneself. We do not expect this algorithm (even in its tabular form) to enjoy a last-iterate convergence to the Nash equilibrium; see the discussion around the OMD algorithm in Equation (8).

Now, when $\beta = 1$, then the alternative policy is $\pi_\theta^{\beta=1} = \mu$, thus we are improving the current policy against the (fixed) reference policy μ (i.e., optimizing $\pi \mapsto \mathcal{P}_\tau(\pi, \mu)$), thus this a version of **best-response** (BR) against μ. This will generally not converge to the Nash equilibrium either because there is no reason that this BR cannot be exploited.

Nash-EMA-PG. As an alternative to Nash-MD-PG, we consider as alternative policy π' another mixture policy $\pi' \stackrel{\text{def}}{=} \pi_{\bar{\theta}_t}$ where $\bar{\theta}_t$ is a exponential moving average (EMA) of the past values of the parameter $(\theta_s)_{s \leq t}$, defined (recursively) by $\bar{\theta}_t = (1-\beta)\theta_t + \beta\theta_0$. Thus when $\beta = 0$ then $\pi_{\bar{\theta}_t} = \pi_{\theta_t}$ and the algorithm is just self-play, and when $\beta = 1$, then $\pi_{\bar{\theta}_t} = \pi_{\theta_0}$ and the algorithm is a best response again the fixed initial policy π_{θ_0}.

Now for any other $\beta \in (0,1)$ the policy uses as parameter a mixture of past parameters. Because of the non-linearity of the policy representation, there is no guarantee that this policy is the mixture of the corresponding past policies. However, prior work in deep learning (Grill et al., 2020; Wortsman et al., 2022; Busbridge et al., 2023; Rame et al., 2023) suggests that it could be a reasonable first-order approximation to it.

G. Experiments

We now report experiments on a summarisation task and compare several algorithms for NLHF (self-play, best-response against μ, Nash-MD-PG and Nash-EMA-PG) as well as a RLHF baseline.

G.1. Preference models versus reward models

In this section, we compare parametric preference models $\mathcal{P}_\theta$ and reward models r_θ. Preference models assigns a score $\mathcal{P}_\theta(y \succ y'|x) \in [0,1]$ that can be interpreted as the probability of generation y being preferred to generation y' given the context x. The preference $\mathcal{P}_\theta(y \succ y'|x)$ is initialised by using a LLM prompted in the following way:

```
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 -
```

where ⟨text⟩ corresponds to x, ⟨summary1⟩ to y, and ⟨summary2⟩ to y'. We then use the last logit for an arbitrary chosen token and pass it through a sigmoid function to output a single number in $[0,1]$. This number models the preference $\mathcal{P}_\theta(y \succ y'|x)$. We train the LLM to fit the underlying human preference probability $\mathcal{P}(y \succ y'|x)$ by minimizing a cross-entropy loss on a dataset $D = \{(x^k, y_w^k, y_l^k)_{1 \leq k \leq K}\}$, where y_w^k is the preferred generation, y_l^k is the less preferred generation and K is the number of examples:

$$\mathcal{L}_{\mathcal{P}}(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\left(\mathcal{P}_\theta(y_w \succ y_l|x)\right)\right].$$

Reward models assigns a score $r_\theta(x,y) \in \mathbb{R}$ that can be interpreted as the value of a generation y given a context x. The reward $r_\theta(y|x)$ is defined by prompting the LLM in the following way: 'Context - ⟨text⟩, Summary - ⟨summary⟩' where ⟨text⟩ corresponds to x and ⟨summary⟩ to y. We then use the last logit for an arbitrary chosen token to output a single number. This number models the reward $r_\theta(y|x)$. Reward models are trained to fit the underlying human preference probability $\mathcal{P}(y \succ y'|x)$ via a Bradley-Terry model $\mathcal{P}_{BT}(y \succ y'|x) \stackrel{\text{def}}{=} \sigma\left(r_\theta(x,y) - r_\theta(x,y')\right)$ where $\sigma(x)$ is the sigmoid function. They use the same preference dataset D and minimize the following cross-entropy loss:

$$\mathcal{L}_r(\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim D}\left[\log\left(\sigma\left(r_\theta(y_w|x) - r_\theta(y_l|x)\right)\right)\right].$$

In our experiments, we use the summarization dataset described in (Stiennon et al., 2020) that has been built from the TL;DR dataset (Völske et al., 2017). We train our preference and reward models on the train set $D_{\texttt{Train}}$, that contains 92820

examples, and evaluate them on a test set of high confidence data $D_{\texttt{Test}}$. To measure the quality of our models we use the expected agreement, also called *accuracy*, between our models and the human ratings:

$$\mathcal{A}(\mathcal{P}_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim D}\left[\mathbf{1}_{\{\mathcal{P}_\theta(y_w \succ y_l|x)\geq 0.5\}}\right],$$
$$\mathcal{A}(r_\theta) = \mathbb{E}_{(x,y_w,y_l)\sim D}\left[\mathbf{1}_{\{\sigma(r_\theta(y_w|x)-r_\theta(y_l|x))\geq 0.5\}}\right].$$

Our first experiment (see Figure 1) shows the accuracy of preference models with different sizes. Our models are T5X encoder-decoder models (transformer models) that have been described in detail in (Roberts et al., 2023; Roit et al., 2023). We use different sizes: T5X-small (110M), T5X-XL (3B) and T5X-XXL (11B). We see, on the test set, that the bigger the model the better the accuracy. However, there is relatively small gains going from 3B to 11B in this specific summarization task. In the remaining, we therefore run our experiments on T5X-XL models only.

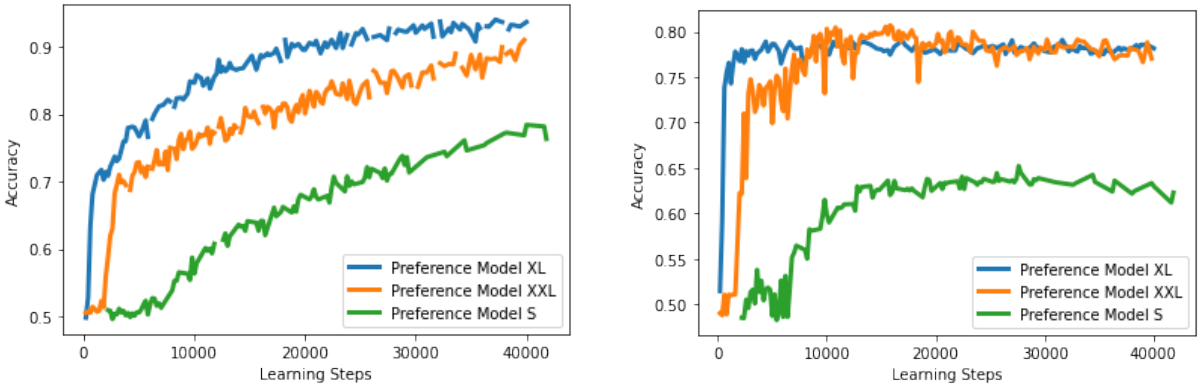

Figure 1. Learning curves showing the accuracy of preference models of different sizes on the train set (left) and on the test set (right).

Our second experiment consists in looking at the accuracy of T5X-XL reward model versus the accuracy of a T5X-XL preference model. We observe that the preference model has a slightly better accuracy than the reward model on the test set (peak accuracy for the preference model is around 0.78 vs 0.76 for the reward model).

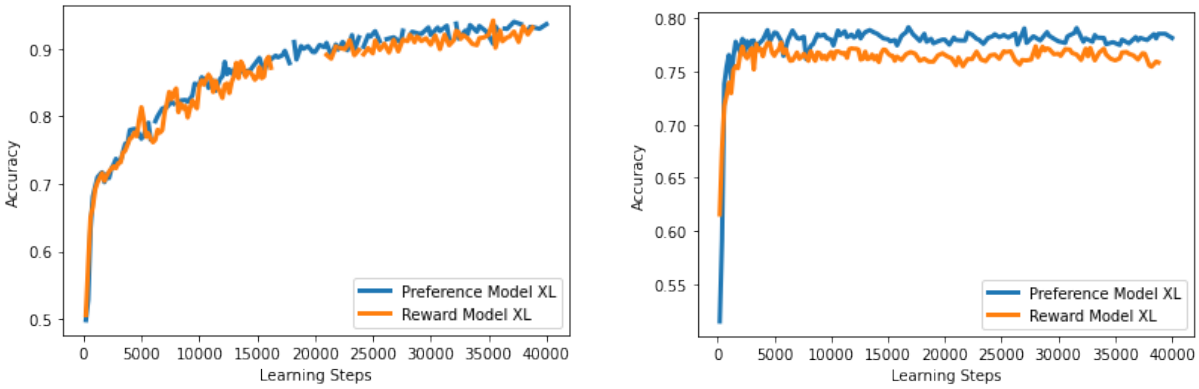

Figure 2. Learning curves showing the accuracy of a preference model versus the accuracy of a reward model of the same size on the train set (left) and on the test set (right).

G.2. Supervised fine-tuned (SFT) initial policy

In all our experiments, we will initialize our policy with a T5X-L model and fine-tune it by supervised learning using the OpenAI dataset described in (Stiennon et al., 2020) that was built from the TL;DR dataset (Völske et al., 2017). We call this supervised fine-tuned model the SFT. In all our experiments, our policies are initialized with this SFT.

For all our policy models, we opted for a T5X-L model, as opposed to T5X-XL, for computational efficiency and to compute the pairwise comparisons across our policies. The primary objective of these experiments is to provide a proof of concept

for the NLHF approach introduced in this paper, rather than striving for state-of-the-art performance in text summarization. Therefore, our aim is to conduct a fair and equitable comparison among the various approaches.

G.3. RLHF baseline

We established a RLHF baseline by initializing our model with the SFT and then updating the policy by doing 10000 steps of a regularized policy gradient update:

$$\mathbb{E}_{x\sim\rho,y\sim\pi_\theta(\cdot|x)}\left[\nabla_\theta \log \pi_\theta(y|x)\left(R(x,y) - \tau \mathrm{KL}(\pi_\theta(\cdot|x),\mu(\cdot|x))\right)\right], \tag{16}$$

where the reward $R(x, y)$ comes from the trained T5X-XL reward model, as described in Subsection G.1. We conducted a sweep across a set of values $0.01, 0.02, 0.05, 0.1, 0.2$ for the parameter τ of the KL-regularization. The value $\tau = 0.05$ has been selected for the pairwise comparison table below.

G.4. NLHF algorithms Nash-MD and Nash-EMA

We initialize our policy with the SFT and update the model by executing the Nash-MD-PG and Nash-EMA-PG algorithms as outlined in Section F. The preference model $\mathcal{P}$ used in these algorithms is derived from the trained T5X-XL model, as described in Subsection G.1.

We conducted a sweep over the values $\tau \in \{0.02, 0.01, 0.008, 0.005\}$ and selected $\tau = 0.008$ for all Nash-MD and Nash-EMA experiments for the pairwise comparison table below.

For Nash-MD-PG we conducted a sweep over the mixing coefficient $\beta \in \{0, 0.125, 0.250, 0.375, 0.5, 0.625, 0.75, 0.875, 1.0\}$ (used in the definition of the alternative policy defined in Section F.3) and for Nash-EMA-PG we have swept over $\beta \in \{0, 0.999, 0.9995, 0.9999, 1.0\}$.

G.5. Pairwise preference between all the models

Here are the list of all the models we considered for pairwise preference comparison.

- SFT: Supervised-fined-tuned, described in Subsection G.2. All models all initialised with this SFT and this SFT is also the policy μ we use for the KL-regularization.
- RLHF described in Subsection G.3 with regularization coefficient $\tau = 0.05$.
- SP (self-play). This corresponds to Nash-MD-PG with mixture coefficient $\beta = 0$ (or equivalently Nash-EMA-PG with $\beta = 0$ as both algorithms are equivalent for $\beta = 0$), described in Subsection G.4. The policy improves by playing against itself (the alternative policy $\pi' = \pi_\theta$ is the current policy).
- MD1 to MD6 is Nash-MD-PG with $\beta \in \{0.125, 0.25, 0.375, 0.5, 0.625, 0.75\}$.
- BR is best-response against SFT. This corresponds to Nash-MD-PG with $\beta = 1$ (or equivalently Nash-EMA-PG with $\beta = 1$). The policy improves by playing against the fixed SFT policy.
- EMA1 and EMA2 are the last-iterate of Nash-EMA-PG (i.e., returns the last policy), with $\beta \in \{0.999, 0.9995\}$.
- EMA1* and EMA* are the EMA policy of Nash-EMA-PG (i.e., returns the policy with average weight) with $\beta \in \{0.999, 0.9995\}$.

All models are trained for 10000 steps. The Nash-MD models (as well as SP and BR) and Nash-EMA are trained with a regularization coefficient of $\tau = 0.008$. The pairwise preference comparisons under $\mathcal{P}_\tau$ are given in Table 2; these figures are estimated based on 1,000 pairwise comparisons, and hence an upper bound on the width of a 95% confidence interval for each is ± 0.032, based on the exact Clopper-Pearson method for Bernoulli proportions (Clopper & Pearson, 1934). Note that the Clopper-Pearson method can be used to deduce a per-element confidence interval which may be considerably narrower in cases where the empirically observed preference rate is close to 0 or 1.

We will analyse these results after the next section where we describe an evaluation of our models based on a preference model build from a much larger LLM.

Table 2. The regularized preference $\mathcal{P}_\tau(\pi_c \succ \pi_r)$ between column policy π_c against row policy π_r

$\mathcal{P}_\tau$	SFT	RLHF	SP	**MD1**	MD2	MD3	MD4	MD5	MD6	BR	EMA1	EMA2	EMA1*	EMA2*
SFT	0.500	0.975	0.981	**0.986**	0.983	0.982	0.979	0.970	0.967	0.933	0.965	0.970	0.971	0.975
RLHF	0.025	0.500	0.741	**0.769**	0.752	0.744	0.661	0.450	0.340	0.167	0.640	0.531	0.617	0.671
SP	0.019	0.259	0.500	**0.547**	0.506	0.509	0.406	0.244	0.185	0.082	0.418	0.338	0.363	0.450
MD1	0.014	0.231	0.453	**0.500**	0.471	0.469	0.354	0.224	0.165	0.079	0.372	0.308	0.348	0.409
MD2	0.017	0.248	0.494	**0.529**	0.500	0.492	0.393	0.231	0.182	0.084	0.426	0.315	0.375	0.454
MD3	0.018	0.256	0.491	**0.531**	0.508	0.500	0.380	0.230	0.153	0.087	0.411	0.328	0.349	0.457
MD4	0.021	0.339	0.594	**0.646**	0.607	0.620	0.500	0.306	0.224	0.088	0.508	0.416	0.458	0.531
MD5	0.030	0.550	0.756	**0.776**	0.769	0.770	0.694	0.500	0.380	0.169	0.682	0.554	0.627	0.697
MD6	0.033	0.660	0.815	**0.835**	0.818	0.847	0.776	0.620	0.500	0.269	0.735	0.644	0.706	0.777
BR	0.067	0.833	0.918	**0.921**	0.916	0.913	0.912	0.831	0.731	0.500	0.856	0.789	0.830	0.875
EMA1	0.035	0.360	0.582	**0.628**	0.574	0.589	0.492	0.318	0.265	0.144	0.500	0.407	0.448	0.507
EMA2	0.030	0.469	0.662	**0.692**	0.685	0.672	0.584	0.446	0.356	0.211	0.593	0.500	0.540	0.627
EMA1*	0.029	0.383	0.637	**0.652**	0.625	0.651	0.542	0.373	0.294	0.170	0.552	0.460	0.500	0.589
EMA2*	0.025	0.329	0.550	**0.591**	0.546	0.543	0.469	0.303	0.223	0.125	0.493	0.373	0.411	0.500

G.6. Evaluation using the PaLM 2 preference model

While the ideal approach for evaluating our models would involve soliciting human preferences between summaries generated by different models, we resort to a proxy method using the highly capable LLM, PaLM 2 Large (Anil et al., 2023). We query this model to obtain a preference signal, which we refer to as the PaLM 2 preference model $\mathcal{P}^*(y \succ y'|x)$, achieved by prompting the LLM in the following manner:

```
'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 -',
```

where $\langle$`text`$\rangle$ corresponds to x, $\langle$`summary1`$\rangle$ to y, and $\langle$`summary2`$\rangle$ to y'.

This evaluation approach shares similarities with the method employed by Lee et al. (2023). To obtain an assessment of the preference $\mathcal{P}^*(\pi \succ \pi')$, we compute the ratio between the total number of token '1' generated and the total number of token '1' or '2' across 2000 samples drawn from the distribution $(x \sim \rho, y \sim \pi(\cdot|x), y' \sim \pi'(\cdot|x))$.

This $\mathcal{P}^*$ serves as an approximate surrogate for human preferences. Notably, it is essential to highlight that the preference model $\mathcal{P}$ utilized during the training of our policies is considerably smaller in size than $\mathcal{P}^*$ and corresponds to a different model. Specifically, $\mathcal{P}$ is based on the T5X-XL model, fine-tuned with TL;DR data, whereas $\mathcal{P}^*$ is derived from the PaLM 2 Large model.

The pairwise preference comparisons under $\mathcal{P}^*$ using the PaLM 2 Large model are given in Table 3. As each element is estimated with 2000 samples, the confidence interval, an upper bound on the 95% confidence interval is given by ± 0.023, based on the exact Clopper-Pearson method for Bernoulli proportions (Clopper & Pearson, 1934).

Table 3. PaLM 2 preference $\mathcal{P}^*(\pi_c \succ \pi_r)$ model between column policy π_c against row policy π_r.

$\mathcal{P}^*$	SFT	RLHF	SP	**MD1**	MD2	MD3	MD4	MD5	MD6	BR	EMA1	EMA2	EMA1*	EMA2*
SFT	0.500	0.990	0.983	**0.982**	0.989	0.987	0.985	0.982	0.965	0.943	0.970	0.961	0.977	0.980
RLHF	0.010	0.500	**0.489**	**0.598**	**0.519**	**0.561**	**0.501**	**0.436**	**0.284**	**0.148**	0.468	0.320	0.477	0.510
SP	0.017	0.511	0.500	**0.592**	0.504	0.545	0.499	0.451	0.310	0.211	0.445	0.362	0.464	0.488
MD1	0.018	0.402	0.408	**0.500**	0.425	0.470	0.369	0.362	0.238	0.163	0.391	0.270	0.400	0.447
MD2	0.011	0.481	0.496	**0.575**	0.500	0.513	0.491	0.434	0.298	0.196	0.460	0.351	0.430	0.496
MD3	0.013	0.439	0.455	**0.530**	0.487	0.500	0.484	0.408	0.273	0.187	0.429	0.323	0.413	0.472
MD4	0.015	0.499	0.501	**0.631**	0.509	0.516	0.500	0.428	0.265	0.161	0.468	0.358	0.437	0.503
MD5	0.018	0.564	0.549	**0.638**	0.566	0.592	0.572	0.500	0.329	0.210	0.532	0.389	0.518	0.539
MD6	0.035	0.716	0.690	**0.762**	0.702	0.727	0.735	0.671	0.500	0.342	0.652	0.548	0.651	0.691
BR	0.057	0.852	0.789	**0.837**	0.804	0.813	0.839	0.790	0.658	0.500	0.743	0.640	0.752	0.774
EMA1	0.030	0.532	0.555	**0.609**	0.540	0.571	0.532	0.468	0.348	0.257	0.500	0.381	0.480	0.556
EMA2	0.039	0.680	0.638	**0.730**	0.649	0.677	0.642	0.611	0.452	0.360	0.619	0.500	0.585	0.659
EMA1*	0.023	0.523	0.536	**0.600**	0.570	0.587	0.563	0.482	0.349	0.248	0.520	0.415	0.500	0.555
EMA2*	0.020	0.490	0.512	**0.553**	0.504	0.528	0.497	0.461	0.309	0.226	0.444	0.341	0.445	0.500

G.7. Analysis of the results

First, let us mention that the RLHF baseline that we have built is a very strong baseline. It beats SFT with a win rate of 99% marking the highest win rate observed against SFT among all models when using the PaLM 2 preference model $\mathcal{P}^*$

Best-response against self-play (BR) does not exhibit strong performance. Despite being trained explicitly to outperform self-play during training, its $\mathcal{P}^*$-evaluation yields a relatively modest score of 94% against self-play. Furthermore, BR performs poorly against RLHF and all other Nash-based approaches. This suggests the possibility of 'preference hacking,' where BR may be overly adapting to the preference model by overfitting to the specific SFT policy.

Self-play (SP) exhibits strong overall performance, with notable exceptions in the $\mathcal{P}^*$ evaluation against RLHF and the Nash-MD models (for $\beta \leq 0.5$). This suggests that enhancing one's policy through self-play could be a promising avenue for improving the initial model. However, it's essential to acknowledge that self-play does not guarantee the attainment of a Nash equilibrium, as cyclic patterns are possible, as discussed in the Theory Section. In particular, SP is found to be vulnerable to exploitation by certain Nash-MD models.

The Nash-MD models, especially those with $\beta \leq 0.5$, exhibit very strong performance. Notably, Nash-MD models with $\beta = 0.125$, $\beta = 0.25$, and $\beta = 0.375$ outperform all other models, including RLHF. Among them, Nash-MD with $\beta = 0.125$ (highlighted in bold as 'MD1') emerges as the top-performing model, surpassing all others in both the training preference model $\mathcal{P}_\tau$ and the evaluation model $\mathcal{P}^*$.

All Nash-EMA models, including EMA1 and EMA2 (representing the last iterate) as well as EMA1* and EMA2* (representing the average policy), are outperformed by Nash-MD for $\beta \leq 0.5$ and RLHF. This observation may suggest that the first-order approximation of the mixture policy as the policy having an average (EMA) weight may not be well-suited in this context, potentially contributing to the overall lower performance.

Examining Nash-MD, which emerges as the most efficient method, it is interesting to note that both extreme values of the mixing parameter $\beta \in [0, 1]$, namely $\beta = 0$ (self-play) and $\beta = 1$ (best-response against SFT), result in suboptimal performance compared to intermediate values of β (particularly $\beta = 0.125$, $\beta = 0.25$, and $\beta = 0.375$). This trend is visible, for instance, in the highlighted blue row showing Nash-MD (for $\beta \in 0, 0.125, 0.25, 0.375, 0.5, 0.625, 0.75, 1.0$) against RLHF. It suggests that improving one's policy by playing against a mixture of the initial policy and the current policy yields superior model improvement compared to interactions with either the initial policy or the current policy in isolation.