DECOUPLING LEARNING FROM NEGATIVE AND POSITIVE FEEDBACK IN PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing preference optimization methods are mainly designed for directly learning from human feedback with the assumption that paired examples (preferred vs. dis-preferred) are available. In contrast, we propose a method that can leverage unpaired preferred or dis-preferred examples, and works even when only one type of feedback (positive or negative) is available. This flexibility allows us to apply it in scenarios with varying forms of feedback and models, including training generative language models based on human feedback as well as training policies for sequential decision-making problems, where learned (value) functions are available. Our approach builds upon the probabilistic framework introduced in (Dayan & Hinton, 1997), which proposes to use expectation-maximization (EM) to directly optimize the probability of preferred outcomes (as opposed to classic expected reward maximization). To obtain a practical algorithm, we identify and address a key limitation in current EM-based methods: when applied to preference optimization, they solely maximize the likelihood of preferred examples, while neglecting dis-preferred samples. We show how one can extend EM algorithms to explicitly incorporate dis-preferred outcomes, leading to a novel, theoretically grounded, preference optimization algorithm that offers an intuitive and versatile way to learn from both positive and negative feedback.

1 INTRODUCTION

The use of preference annotated data for training machine learning models has a long history going back to early algorithms for recommender systems and market research (Guo & Sanner, 2010; Boutilier, 2002; Bonilla et al., 2010). These days preference optimization algorithms are receiving renewed attention since they are a natural candidate for shaping the outputs of deep learning systems, such as large language models (Ouyang et al., 2022; Team et al., 2024) or control policies, via human feedback (Christiano et al., 2017; Rafailov et al., 2023; Azar et al., 2023). Arguably, preference optimization algorithms can also be a natural choice even when direct human feedback is not available but one instead aims to optimize a machine learning model based on feedback from a hand-coded or learned critic function (judging desirability of solutions). Here preference optimization methods are useful since they let us optimize the model to achieve desired outcomes based on relative rankings between outcomes alone (rather than requiring absolute labels or carefully crafted reward functions).

Among preference optimization approaches, those based on directly using preference data – as opposed to casting preference optimization as reinforcement learning from (human) feedback – such as DPO (Rafailov et al., 2023), have emerged as particularly successful since they only require access to an offline dataset of paired preference data, and are fairly robust to application domain and hyperparameter settings. However, algorithms within this class make specific assumptions tailored to their application domain. They were designed to optimize LLMs from human feedback in the form of comparisons of generated sentences and thus, by design, require paired preference data (since they directly model a specific choice of preference distribution). We are interested in finding algorithms that are more flexible, and applicable in settings where the assumptions underlying DPO do not apply.

In this work we take a fresh look at preference optimization from a probabilistic inference perspective that has been used with great success in the literature on KL regularized reinforcement learning (Dayan & Hinton, 1997; Peters et al., 2010; Abdolmaleki et al., 2018). We find that from this perspective a simplified approach to preference optimization can be derived that is intuitive to understand and is

capable of leveraging an arbitrary number of unpaired preferred or dis-preferred outcomes, or even solely one type (positive or negative) of preference feedback. In particular, our method is able to learn even if exclusively positive or negative examples are available. Formally, our method involves an objective consisting of three log likelihood terms that are derived from first principles: maximizing the likelihood of preferred outcomes, minimizing the likelihood of dis-preferred outcomes, while staying close to a reference distribution (see equation 10). We show the effectiveness of our method across a wide range of benchmarks including synthetic benchmarks, training policies for continuous control, and training large language models (LLMs) from human feedback.

2 Related Work

2.1 RL as Inference

Viewing reinforcement learning through the lens of probabilistic inference offers an alternative framing of RL (Dayan & Hinton, 1997). This "RL as inference" perspective has gained considerable attention recently (Levine, 2018) inspiring various expectation-maximization (EM) based RL algorithms (Peters et al., 2010; Abdolmaleki et al., 2018). Essentially, these policy improvement algorithms can be viewed as performing EM to optimize the likelihood of a successful outcome. However, a limitation of these algorithms is their reliance on successes (preferred outcome) data. In this paper, we extend this framework to incorporate dis-preference information; effectively allowing the policy to make unwanted outcomes *less* likely. We show that this alone can have an positive effect on data efficiency and performance on certain tasks, notwithstanding the added flexibility.

2.2 Preference optimization

Preference optimization methods like Direct Preference Optimization (DPO; Rafailov et al., 2023) and Identity Preference Optimization (IPO; Azar et al., 2023) have enjoyed much attention lately, especially in the LLM training literature. This success is mostly due to a so-called *direct* optimization of human preferences, in contrast to reward model training required in RL from human feedback (RLHF) training pipelines. Nevertheless, these preference optimization methods were designed specifically to learn from a particular type of data: pairs of preferred and dis-preferred data, usually coming from humans indicating their preference over a pair of LLM responses to their query. This can be restrictive in scenarios where multiple outcomes need to be considered, and DPO has since been extended to multiple generations and compared to a novel method Efficient Exact Optimization (EXO; Ji et al., 2024), both shown to outperform the RLHF baseline in cases where a reward model is available. In this paper, we leverage the RL as inference framework to generalize preference optimization even further, allowing for more general algorithms derived from first principles. Our approach can not only handle scenarios with multiple generations but it also naturally handles cases where only one type of feedback is accessible (i.e. all generations are failures), which can be particularly useful for challenging task with binary success/failure outcomes (e.g. code, math, safety assurance).

3 Using preferred and dis-preferred outcomes for policy optimization

In this section we present an approach to optimising policies based on preference data and show how it can be used to tackle a variety of problem settings in Machine Learning; e.g. policy optimisation in a standard RL setting or learning from human preference feedback. We build upon a large body of existing work in probabilistic inference for policy optimization. In particular we make use of a formulation underlying many existing KL regularized RL approaches that are motivated from the RL as inference perspective such as REPS (Peters et al., 2010), AWR (Peng et al., 2019) and MPO (Abdolmaleki et al., 2018). We will show that, when applied to preference optimization, the Expectation-Maximization (EM) approach employed by this class of algorithms results in a natural formulation of maximizing (weighted) likelihood of preferred outcomes. Since such a formulation is appealing due to its simplicity but cannot effectively use information about dis-preferred outcomes (unless we make the often unrealistic assumption of having access to the full probability distribution

over preferences) we finally derive a simple extension that enables the use of dis-preferred/disliked data-points.

The resulting algorithm has multiple intriguing properties: it can make use of preference data containing positive and negative outcomes but it does not require paired outcomes (i.e. it can make use of data for which we only know whether it is either good or bad, without knowing about relative preference with respect to other data-points) and can thus also naturally utilize unbalanced datasets (where e.g. we have multiple preferred options for each dis-preferred example, or vice-versa). Due to the close relationship of our algorithm to the existing MPO algorithm (Abdolmaleki et al., 2018) *we refer to it as preference based MPO (PMPO).*

3.1 Background on maximising preferred outcomes

We review the preference based RL formulation common in RLHF (Ziegler et al., 2019; Rafailov et al., 2023) and show how methods from the literature on EM based policy optimization (Rawlik et al., 2013; Peters et al., 2010; Abdolmaleki et al., 2018) can be naturally applied to it.

In the following, x denotes the conditioning variable such as the state/observation in classical RL, or a document and query in the LLM finetuning setting. Providing this information to a model (or policy) produces $\pi(y|x)$, a probability distribution over outputs y; these would be actions in classical RL or responses/generations (sequences of tokens) in the LLM finetuning literature. We will also make use of the definition of a KL divergence between conditional distributions which we define as $\mathrm{KL}(p(\cdot|x) \parallel q(\cdot|x)) = \mathrm{KL}(p, q; x) = \mathbb{E}_{y\sim p(\cdot|x)}[\log p(y|x) - \log q(y|x)]$.

Objective. Define a binary random variable S, which takes a value of 1 in the event of a successful outcome and 0 otherwise. To lighten notation, we will use the shorthand $p(S)$ and $p(S')$ to mean $p(S = 1)$ and $p(S = 0)$, respectively, and similarly for the conditioned distributions. In words, our goal is to optimize the parameters θ of a parametric policy $\pi_\theta(y|x)$ to produce outcomes y that have a high likelihood of being preferred as measured by an unknown preference distribution $p(S|y, x)$, i.e. the event that y is a 'preferred' or 'successful' response to the condition x:

$$\max_\theta \mathop{\mathbb{E}}_{y\sim\pi_\theta} p(S|y, x) \tag{1}$$

Reference model. In addition to the above general formulation we assume access to a reference model π_{ref} that can either consist of a previous iteration of the model we would like to improve, or be the outcome of a pre-training or supervised fine-tuning phase (as routinely employed in RLHF for LLMs). We refer to this model as the reference policy and in general we use the terms model and policy interchangeably.

Preference information. In order to derive a practical sample based algorithm we have to assume some knowledge of the preference distribution $p(S|y, x)$; we distinguish two cases in this paper. In the first case we assume we have pointwise access function $f(y, x)$ that is proportional to the log-probability density function (PDF) of the preference distribution. That is, we have a Boltzmann distribution $p(S|y, x) \propto \exp(f(y, x)/\eta)$ where η is a temperature parameter. For cases where we can query a reward function r or a state-action value-function Q,[1] f can be set to be r or Q, respectively. In this case we only need samples (x, y) for optimization without explicit success/failure labels. In the second case we assume we only have access to a dataset of labelled examples:

$$\mathcal{D} = \left\{x^{(i)},\ y^{(i,j)},\ s^{(i,j)}\right\}_{i,j=1}^{N,M_i}$$

where $y^{(i,j)} \sim \pi_{\text{ref}}(\cdot|x^{(i)})$ and the $s^{(i,j)}$ are binary preference labels, usually obtained from human feedback. In other words, in this second case we only have samples from the real preference distribution $P(S|y, x)$ as opposed to a Boltzmann model for it.

Policy optimization. Let us drop the superscripts (i) for now and only consider the objective on a per-condition basis, ultimately we average over the batch. Then for every conditioning $x = x^{(i)}$, the problem is finding a policy that achieves the highest marginal probability of preferred outcomes. This

[1] Defined as $Q(y, x) = \mathbb{E}[\sum_t \gamma^t r(y_t, x_t)|x_o = x, y_0 = y]$ for a timeseries of observation/action pairs.

amounts to optimizing

$$\log p_\pi(S|x) = \underbrace{\mathop{\mathbb{E}}_{y\sim q}\Big[\log \frac{\pi(y|x)p(S|y,x)}{q(y|x)}\Big]}_{\mathcal{J}(\pi;\, q,\, x)} + \mathrm{KL}(q(y|x)\| \, p_\pi(y|S,x)), \tag{2}$$

where we have used a standard formulation from the probabilistic inference literature (Kingma & Welling, 2013) to decompose the objective into an evidence lower bound $\mathcal{J}(\pi; q, x)$ and a KL term by introducing an auxiliary variational distribution q that we will use as a 'vehicle' to perform stable optimization via expectation maximization (Dayan & Hinton, 1997). The goal of EM is to iteratively find a tight lower bound given the current estimate π_{ref} by optimizing for q (E-Step) and improve the lower bound $\mathcal{J}$ by optimizing for π (M-Step). More concretely, in the E-step, we fix $\pi = \pi_{\text{ref}}$ and find the $\hat{q}$ which minimizes the KL; this tightens the bound. In the M-step, we fix $q = \hat{q}$ and maximize the lower bound $\mathcal{J}(\pi_\theta; \hat{q}, x)$ to update π_θ. This process of tightening the bound and improving the policy constitutes one iteration of policy improvement over π_{ref}.

E-step: Tighten the lower bound by fixing $\pi = \pi_{\text{ref}}$ and minimize $\mathrm{KL}(q(\cdot|x) \parallel p_{\pi_{\text{ref}}}(\cdot|S,x))$. Following prior work (Dayan & Hinton, 1997; Peters et al., 2010; Abdolmaleki et al., 2018), since the KL is minimized when both distributions are equal, the solution can be expressed in closed form as $\hat{q}(y|x) = p_{\pi_{\text{ref}}}(y|S,x)$. Then, according to Bayes rule:

$$p_{\pi_{\text{ref}}}(y|S,x) = \frac{1}{Z_x}\pi_{\text{ref}}(y|x)p(S|y,x), \tag{3}$$

where we used the normalization factor $Z_x = \int \pi_{\text{ref}}(y|x)p(S|y,x)\,\mathrm{d}y$. Recall that $p(S|y,x)$ is still a modelling choice discussed in the *Preference information* section.

M-Step: Optimize the lower bound $\mathcal{J}$ fixing $q = \hat{q}$ from the previous step. Since this problem does not have an analytic solution we use a parametric function approximator π_θ, usually a large neural network, and maximize the following objective via gradient ascent:

$$\mathcal{J}(\pi_\theta; \hat{q}, x) = \mathop{\mathbb{E}}_{y\sim\hat{q}}\Big[\log \frac{\pi_\theta(y|x)p(S|y,x)}{\frac{1}{Z_x}\pi_{\text{ref}}(y|x)p(S|y,x)}\Big] = \mathop{\mathbb{E}}_{y\sim\hat{q}}\Big[\log \pi_\theta(y|x)\Big] + K \tag{4}$$

$$\mathcal{J}(\pi_\theta; x) = \mathop{\mathbb{E}}_{y\sim\pi_{\text{ref}}}\Big[\frac{p(S|y,x)}{Z_x}\log \pi_\theta(y|x)\Big], \tag{5}$$

where K represents all constant terms that are independent of θ and are dropped from the final objective. Notice that this objective amounts to a weighted maximum likelihood with preferences determining the weights and samples coming from π_{ref}. Notice also that the final expression subsumes the closed form E-step solution such that we can safely consider only this objective and introduce the short-hand $\mathcal{J}(\pi_\theta; x)$, dropping the implicit dependence on the E-step solution. In practice, to optimize this objective we need to form a Monte-Carlo approximation of the expectation in Eq. (5). We distinguish the two cases mentioned in the *Preference information* section.

In the first case, we assume access to a function f that is proportional to the preference log-probability, and access to M responses $y^{(j)}$ for each x. We can then set $p(S|y,x) \approx w^{(j)} \propto \exp(f(y^{(j)}, x)/\eta)$ in Eq. (5) (a softmax of f across the responses $y^{(j)}$ to x). This is the case commonly studied in the literature, e.g., in MPO where one uses $f = Q(y^{(j)}, x)$.

It is often unrealistic to assume access to a reliable model of preference labels. For example, preferences often come from human annotations and we thus only have access to samples or we might only have access to a learned and unreliable preference model.[2] To cover this case, let us partition our dataset of labeled examples $\mathcal{D} = \mathcal{D}_a \cup \mathcal{D}_r$ where $\mathcal{D}_a = \{y^{(j)} \ni (s^{(j)} = 1)\}_{i=1:M}$ and $\mathcal{D}_r = \{y^{(j)} \ni (s^{(j)} = 0)\}_{i=1:M}$, denote accepted (preferred) samples and rejected (dis-preferred) samples, respectively. In this case we can still use the objective from Eq. (5), using the binary preferences $s^{(j)}$ as weights:

$$\mathcal{J}(\pi_\theta; x) \approx \mathcal{J}_a(\pi_\theta; x) = \mathop{\mathbb{E}}_{y^{(j)}\sim\mathcal{D}}\big[s^{(j)}\log \pi_\theta(y^{(j)}|x)\big] = \mathop{\mathbb{E}}_{y^{(j)}\sim\mathcal{D}_a}\log \pi_\theta(y^{(i)}|x), \tag{6}$$

which effectively filters rejected generations $\mathcal{D}_r$ out, thus reverting back to the maximum likelihood objective on preferred data.

[2] A case studied in the offline RL literature where the authors realised that using binary weights often works better as in binary CRR (Wang et al., 2020).

3.2 USING DIS-PREFERRED OUTCOMES VIA REGULARISED MINIMUM LIKELIHOOD

We will now derive a simple way to incorporate negative (dis-preferred) samples into the optimization to address the shortcomings of naively applying the EM-based perspective from the previous section. We would like to incorporate these examples without changing the overall objective since it has well established policy improvement guarantees (Rawlik et al., 2013; Abdolmaleki et al., 2018). To accomplish this we take a second look at the non-parametric variational distribution $\hat{q}$ from Eq. (3) that is the solution to our E-step; since it determines the sampling distribution used for the M-step.

We can realise that the restriction to positive/preferred samples stems from the fact that we express $\hat{q}$ directly in terms of the preference distribution $p(S|y,x)$. A natural question then is: can we re-express $\hat{q}$ in terms of dis-preferences? It turns out the answer to this is positive. Recall that S' denotes the complement of the event S i.e. the event that y is not a successful action/response to a conditioning x. Then by definition, $p(S|y,x) = 1 - p(S'|y,x)$ we can equivalently write

$$\hat{q}(y|x) = \frac{1}{Z_x}\pi_{\text{ref}}(y|x)(1 - p(S'|y,x)). \tag{7}$$

We can plug this form of $\hat{q}$ into the evidence lower bound expressed in Eq. (4). After rearranging terms and re-writing in terms of two expectations over π_{ref} this gives the alternative form:

$$\mathcal{J}(\pi_\theta; x) = \mathop{\mathbb{E}}_{y\sim\pi_{\text{ref}}}\Big[-\frac{p(S'|y,x)}{Z_x}\log\pi_\theta(y|x)\Big] - \frac{1}{Z_x}\,\text{KL}(\pi_{\text{ref}}, \pi_\theta; x) + K, \tag{8}$$

where K again denotes terms independent of π_θ. This version of the objective now is expressed in terms of the dis-preference distribution and we can use it to define a Monte-Carlo estimate based on dis-preferred examples as (where the additional constant 1 results in KL regularization):

$$\mathcal{J}(\pi_\theta; x) \approx \mathcal{J}_r(\pi_\theta; x) = \mathop{\mathbb{E}}_{y^{(j)}\sim\mathcal{D}_r}\big[-\log\pi_\theta(y^{(j)}|x)\big] - \beta\,\text{KL}(\pi_{\text{ref}}, \pi_\theta; x), \tag{9}$$

where β is an additional tuning parameter that is typically set high enough to only remove the dis-preferred outcomes from the prior π_{ref}. As before, our use of samples $s^{(j)}$ (labelled data) filters out part of the dataset; in this case, it is the accepted responses which are filtered out, hence the expectation over $\mathcal{D}_r$. We refer to the appendix for a full derivation. This is a fairly intuitive objective to optimize. It tells us to *minimize* the likelihood of dis-preferred examples while staying close to the reference model. Interestingly, compared to the preferred data case, it has an additional KL term that appears as a result of the reparameterization of the variational distribution. We will see in the experiments that this term is required when learning from negative data to avoid arbitrary solutions where probability is assigned to random out-of-data responses. Intuitively, we can think of the objective as modifying the reference distribution such that the negative examples are removed. Interestingly such an additional KL for the M-step has previously been considered in the literature even for the case where we have access to the preference distribution but must perform optimization based on a limited set of sampled responses; as in MPO (Abdolmaleki et al., 2018). However, previous work used the additional KL term to prevent rapid entropy loss. In contrast, our motivation for incorporating the KL term is to learn from negative samples, as suggested by the derivations.

3.3 LEARNING FROM PREFERRED AND DIS-PREFERRED OUTCOMES

Finally, we can form a combined objective from our two M-step estimates – which both optimize the same quantity but can utilize different samples. That is, we combine Eq. (6) and Eq. (9):

$$\mathcal{J}_{ar}(\pi_\theta; x) = \underbrace{\alpha \mathop{\mathbb{E}}_{y\sim\mathcal{D}_a}\big[\log\pi_\theta(y|x)\big]}_{\text{Learning From Accepted Samples}} \underbrace{- (1-\alpha) \mathop{\mathbb{E}}_{y\sim\mathcal{D}_r}\big[\log\pi_\theta(y|x)\big] - \beta\,\text{KL}(\pi_{\text{ref}}, \pi_\theta; x),}_{\text{Learning From Rejected Samples}} \tag{10}$$

where α is a trade-off parameter between the two estimates. Recall that in practice, this objective will be aggregated over an entire dataset of conditions x and corresponding datasets $\mathcal{D}_a$ and $\mathcal{D}_r$. There are a few interesting things to note about this objective. First, we emphasize that our objective assumes categorization of samples into good/bad or preferred/dis-preferred datasets. As a result, it can be used *even when only positive or only negative samples are available* (this is in contrast to e.g. DPO (Rafailov et al., 2023) or IPO (Azar et al., 2023) which require relative scores of paired

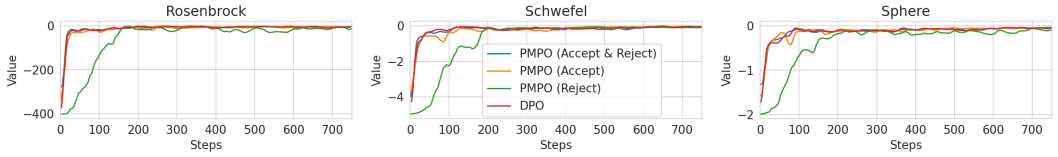

Figure 1: Performance of PMPO and DPO on Benchmark Functions - This figure illustrates the optimization progress of PMPO variants (PMPO-AR, PMPO-A, PMPO-R) on a selection of standard benchmark functions, showcasing their ability to leverage different types of preference feedback.

positive and negative examples for each query x). Furthermore, the objective has also no restriction on the number of positive / negative samples per query x and thus it automatically extends to the multi-sample case for fine-tuning language models. Finally, the objective is intuitive and simple to implement; it amounts simply to maximizing likelihood of good data while minimizing likelihood of bad data and staying close to the reference model. The KL term is implemented in closed form whenever possible. For example, for the autoregressive models used in LLMs, we use the sum of per-token closed-form KL divergences of categorical distributions. This enable us to learn only from a negative feedback without access to positive feedback as suggested by our derivations. See the details in appendix D for KL computation.

4 Extracting preferences from evaluation functions

Our algorithm requires access to preference information, which can come directly from human feedback or be extracted from an evaluation function. This section describes the latter. We consider improving policies within a traditional reinforcement learning (RL) setting; bandit optimization and optimization of language models via RLHF. In each setting our preference-based update rule can be used in the policy improvement step. For this we need to extract preference information from a (possibly learned) evaluation function. This can be achieved in the following way:

Generate Samples: For a given input or state x, sample one or multiple generations y from the current reference policy π_{ref}.

Evaluate Actions: Calculate the evaluation function $f(x, y)$ (e.g. a reward model in RLHF) for each input-generations pair (x, y).

Classify Actions: If $f(x, y) \geq b(x)$, classify the generation y as preferred in state x. Otherwise ($f(x, y) < b(x)$), classify it as dis-preferred.

5 Experiments

We evaluate our algorithm in a variety of different settings, showcasing its utility as a general preference optimization algorithm that can deal with many different forms of preference feedback. We first test it in a Bandit setting (optimizing synthetic benchmark functions) then in a setting where we transform RL on control and robotics tasks into preference optimization. And finally we showcase strong performance for RLHF of large language models. To verify our derivations, we evaluate three different variants of the PMPO algorithm: learning only from accepted samples ($\alpha = 1$), learning only from rejected samples ($\alpha = 0$), and learning from both accepted and rejected samples ($\alpha = 0.5$). We also use MPO (Abdolmaleki et al., 2018) and DPO (Rafailov et al., 2023) as baselines. For all the experiments, we will use a beta value of 0.5 for learning from accept&reject, 0.0 for learning from accept only, and 2.0 for learning from reject only, unless stated otherwise. Furthermore, in all experiments except experiment 5.3, the reference policy for all baselines is updated every N steps to allow for multiple policy improvement steps and demonstrate that our algorithm can effectively optimize the underlying reward function until convergence. For experiment 5.3, we only have access to samples from the reference policy; therefore, we can make only one improvement step, which means the reference policy is effectively fixed. Please note that experiment 5.3 is designed to have access to only positive or negative feedback for each state.

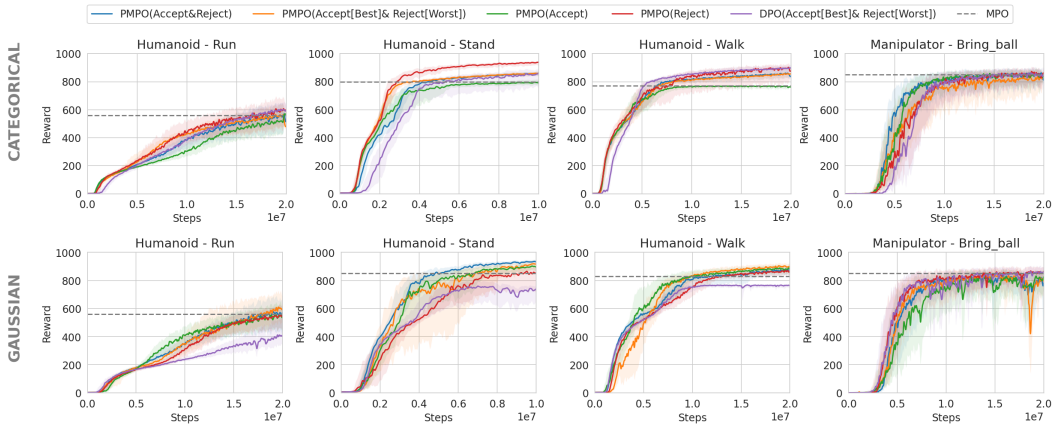

Figure 2: Comparison of PMPO/DPO/MPO for high-dimensional control tasks from the DeepMind Control Suite. We plot average reward over time of training (using 100 episodes for each evaluation).

5.1 Bandit RL: Standard Functions

First, we evaluate our algorithm on a suite of well-known synthetic benchmark functions, including the Rosenbrock, Sphere and Schwefel functions (Hansen et al., 2003); all of which have an optimum value of zero. Each function optimization problem is framed as a preference optimization problem analogous to multi-armed bandit optimization (Auer et al., 2002); i.e. there is no state conditioning x. At each iteration, the policy proposes 4 samples within the function's domain and observes evaluations (function values) as feedback. The reference distribution used for sampling actions is a time lagged version of the policy being optimized (updated every 100 optimization steps). Subsequently, the algorithm utilizes the two top samples as preferred samples and the other two samples as dis-preferred samples. Figure 1 illustrates the performance of our PMPO algorithm under different feedback settings (PMPO-AR: uses all 4 samples and thus accepted and rejected samples, PMPO-A: uses only the accepted samples, PMPO-R: uses only the rejected samples). The results show that all three variants of our algorithm can successfully learn and optimize the objective functions. This demonstrates that we can effectively leverage diverse sources of preference information. It may seem surprising that even just using negative samples is enough to drive optimization towards the optimum but together with the KL constraint towards the slowly changing reference it can be effective. DPO, which uses the best and worst action samples among the 4 sample archives, exhibits similar performance to PMPO-AR.

5.2 Full online RL: control suite

We evaluate our algorithm on a range of control tasks from the DeepMind Control Suite (Tunyasuvunakool et al., 2020). See appendix for details. We cast the setting of optimizing a policy for the control suite as a preference optimization problem by leveraging a learned action-value function (a Q-function)–represented by a separate network trained alongside the policy–to infer preferences for each observed state and action. This is analogous to the actor-critic setting in classical reinforcement learning. Similar to the bandit case, at each iteration, the reference policy proposes four actions for each state in the batch. The top two actions with the highest Q-values are considered preferred samples, while the two actions with the lowest Q-values are treated as dis-preferred samples. We consider two different cases, one where the output of the neural network are mean and standard deviation of a Gaussian control policy and one where the actions are discretized into bins (and the network outputs categorical logits over these bins).

Figure 2 demonstrates that, as in the bandit case, our algorithm can effectively learn from different types of available signals (accept/reject, accept-only, reject-only) to solve high-dimensional tasks, such as controlling humanoid agents to run, stand, and walk, as well as manipulating objects. In all of them PMPO matches or outperforms the strong MPO baseline. Notably, even with only reject signals (PMPO-R), the algorithm is capable of achieving good performance. As predicted by the theory, not using a KL can quickly lead to collapse when using only dis-preferred samples. We also compare

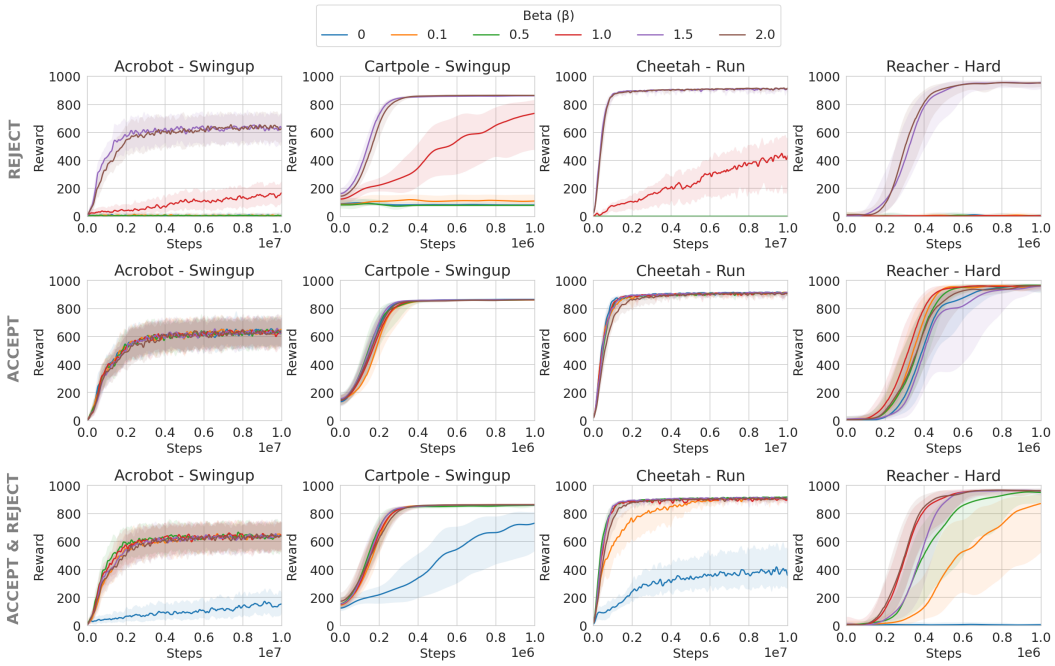

Figure 3: Impact of the KL weight 'beta' on the performance of PMPO. When learning solely from dispreferences across various Control Suite tasks (Reject, $\alpha = 0$), a sufficiently high beta value is required for effective learning. However, when learning from preferences only (Accept) PMPO is robustness to the KL weight 'beta' across different Control Suite tasks, confirming theoretical insights. When both both accept and reject signals are used (Accept & Reject), PMPO shows a partial sensitivity to KL Weight 'beta'. While learning is possible with a wider range of beta values, a beta higher than 0.5 is generally needed for optimal performance.

to an implementation of DPO (Rafailov et al., 2023) which uses the best and worst action sample among the 4 samples. This still results in a strong algorithm that works well when using a discretized action representation. However, in the continuous Gaussian case, DPO requires a very high implicit regularization parameter ($\beta = 20$) which results in slow learning and suboptimal policies. For the sake of fair comparison with DPO that uses the worst and best generation, we also show results for PMPO when only the best is labeled as preferred and the worst is labeled as dispreferred, which is still competitive with DPO.

We further ablate the impact of the KL term on learning solely from dispreferences ($\alpha = 0$), solely from preferences ($\alpha = 1$), and from both ($\alpha = 0.5$). For each of these settings, we sweep over the β parameter in the range $(0.0, 0.5, 1.0, 1.5, 2.0)$. As depicted in Figure 3, when learning exclusively from dispreferences (PMPO-R), the performance is highly sensitive to β. To achieve effective learning, we need to set β sufficiently high (> 1.0), which aligns with our theoretical derivations. In contrast, Figure 3 shows that the algorithm is insensitive to the setting of β when learning only from preferred samples (PMPO-A), again confirming our theoretical insights. When learning from both types of signals (PMPO-AR), as shown in Figure 3, we observe a partial sensitivity to the KL weight β. While the algorithm can learn with a wider range of beta values, a β larger than 0.5 is still necessary to ensure optimal performance across all tasks.

5.3 Offline RL using Advantage Function

In a final set of experiment on control domains we want to show that our algorithm can also be applied to a setting where we have only access to one sample with either a reject or an accept label per state conditioning x. We consider the RGB Stacking benchmark (Lee et al., 2021), a pick-and-place manipulation task with image observations (see appendix for details). We investigate the effect of positive and negative feedback in the context of offline RL to exclude cascading effects from exploration. To this end we take a dataset of 140k episodes from a multi-task RL experiment

trained to convergence (Lampe et al., 2024). We then train a value function on all data and use it to label the transitions in the first 40k episodes as accept (positive advantage) or reject (negative advantage). Different combinations of acceptance, rejection, and BC losses are then compared in order to understand their respective effects. In summary we use: i) the full 140k episodes to train a value function and label the first 40k episodes as accept or reject ; ii) the first 40k episodes to compute the positive weighted part of the loss if labeled as accept and to compute the negatively weighted part of the loss if labeled as reject. The KL part of the loss is calculated on all 140k episodes. Note that the value function is only used to transform the reward annotations into accept and reject labels.

Table 1 shows the achieved reward for different loss combinations. First we run BC on the full 140k episodes and we can observe that the performance is mediocre due to the data containing a significant amount of bad episodes. Using only the accepted transitions for BC training does not result in better performance; this is due to the limited number of positive examples contained in the first 40k episodes. When combining both BC and using the positive (accept) part of the loss, performance does not significantly improve as the large number of negative episodes is not compensated for. On the other hand, combining BC with the negative (reject) part of the loss does significantly improve performance. This is due to the rejection loss successfully pushing the policy away from the negative examples (while keeping close on all other data due to the KL constraint). Finally, best performance is achieved when combining all three losses; and thus effectively utilizing all data. While in this example we have constructed the dataset in a way that the effect is strong, and this might be less the case in more natural settings, it nevertheless shows that using a negative signal can have a significant effect on performance by masking the effect of bad data.

	BC	Accept+BC	Accept	Reject+BC	Accept+Reject+BC
Reward	24	26	27	77	93

Table 1: Comparing different mixtures of acceptance, rejection and BC losses. We measure average reward (over 100 evaluation episodes) across stacking of all 5 triplets. Training with BC is corrupted by bad examples. Training on only accepted examples lacks data. Only when integrating the rejection loss bad data can be masked and performance goes up. Best performance is achieved when combining acceptance, rejection and BC loss signals.

5.4 Language alignment experiments

We apply different versions of the PMPO algorithm to the task of aligning large language models. Specifically, we fine-tune a Gemma 2B pre-trained model using a trained reward model (Gemma Team et al., 2024) using prompts from the LMSYS-chat-1M dataset (Zheng et al., 2023). The reward model has been trained on human preference data with a Bradley-Terry modelisation as explained in (Christiano et al., 2017). In these experiments, we perform one epoch of training, processing a dataset of 500k prompts in approximatively 4000 learner steps, meaning that each batch is composed of 128 prompts and 4 generations per prompt. Similar to the typical RLHF setting, at each iteration, for each prompt in a batch, we sample four generations from the model and rank them based on their reward values. The top two generations are labeled as preferred, and the bottom two as dis-preferred. For the sake of fair comparison with DPO that uses the top one (best) and bottom one (worst) generation, we also show results for PMPO when only the top one is labeled as preferred and the bottom one is labeled as dispreferred. Note that this particular choice could be refined further and tailored to the task. First, Fig. 4 showcases the best PMPO setting, leveraging both accept and reject signals (PMPO-AR) (and we compare to use either feedback signal in isolation). Notably, utilizing both types of feedback leads to faster learning compared to using either signal in isolation (PMPO-A or PMPO-R) and overall our approach is competitive to DPO, which is applicable in this setting by using only the best and worst sample respectively per prompt but would be more restrictive in general (i.e. it cannot naturally make use of unbalanced preference data). As shown on the right, when performing a side by side comparison using GPT-4 (OpenAI et al., 2024) to judge whether our model is preferred over the base Gemma model (using a set of held-out test prompts) the PMPO fine-tuned model wins over the base model. Note in Fig. 4 right, we see some drop indicating some exploitation of the imperfect reward model; known as reward hacking (Skalse et al., 2022). We can see that PMPO-AR is the quickest to "hack the reward"(see Fig. 4 left), it reaches a good performance but then in the middle of training its start hacking the reward and learns a pathological behaviour that makes it performs worse on the independent benchmark. This phenomenon has been observed

consistently in RLHF. Overall, our language alignment experiments provide strong evidence for the effectiveness and versatility of PMPO. Finally, we illustrate in Figure 5 that, again, our algorithm demonstrates the ability to learn effectively from various preference signals, including scenarios with only accept (PMPO-A), only reject (PMPO-R), or both accept/reject (PMPO-AR) feedback. These results highlight the versatility of our approach to different preference acquisition settings. The results also underline the critical role of the KL term in enabling learning exclusively from dis-preferred generations (PMPO-R). As predicted by our derivation, a sufficiently high value $\beta > (1 - \alpha)$ is necessary to stabilize learning in this scenario. In contrast, when learning solely from preferred samples (PMPO-A), the algorithm is insensitive to the value of β in terms of stability.

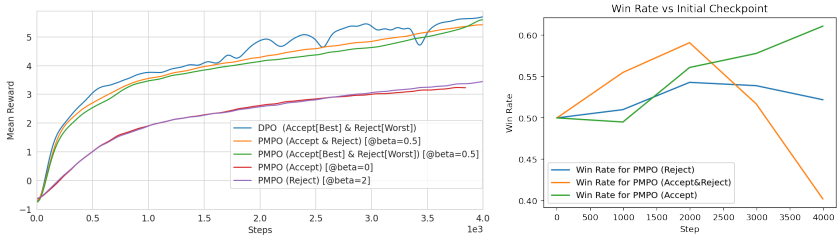

Figure 4: Left: Impact of Combining Accept and Reject Signals - The plot demonstrates the learning progress of PMPO-AR (using both accept and reject signals) compared to PMPO-A and PMPO-R, showcasing faster learning when leveraging both types of feedback in language alignment task and is competitive with DPO. Right: Win-rate when doing A/B comparisons on held-out prompts for PMPO against the base Gemma checkpoint as judged by GPT-4.

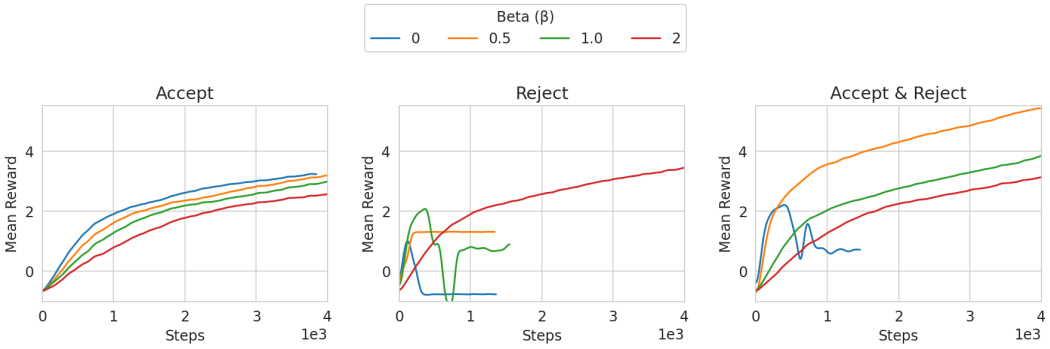

Figure 5: Rewards obtained by the policy at each training step, averaged over the batch and smoothed. Each curve corresponds to a configuration of β specified in the legend. This figure illustrates the ability of PMPO to learn effectively from various preference signals (accept-only, reject-only, or both) in language alignment tasks. highlighting its adaptability to different preference acquisition settings.

6 Conclusion

We propose a novel algorithm for policy optimization from preference feedback derived from the perspective of RL as probabilistic inference. Our policy improvement algorithm has a clear and intuitive objective: it maximizes the likelihood of preferred data while minimizing the likelihood of dis-preferred data. We show that doing the latter in a stable way requires a regularization term forcing the policy to stay close to a reference model. This regularization term follows naturally from the derivation. The main advantage of our algorithm over existing preference optimization algorithms such as DPO is that it does not rely on defining/fitting an explicit model of the preferences and can thus use data containing partial preference information; i.e. we can use data where instead of comparisons between samples we only have accept (or only reject) labels and make no further assumptions on their distribution. In a large number of settings, ranging from classical continuous control to modern LLM finetuning tasks, we show that our method is effective at training policies from such binary preference feedback, without requiring a balanced dataset of positive and negative examples.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *ArXiv*, abs/2310.12036, 2023.

Edwin V. Bonilla, Shengbo Guo, and Scott Sanner. Gaussian process preference elicitation. In *Neural Information Processing Systems*, 2010.

Craig Boutilier. A pomdp formulation of preference elicitation problems. *Proceedings of the National Conference on Artificial Intelligence*, 05 2002.

Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *arXiv preprint arXiv:1706.03741*, 2017.

Peter Dayan and Geoffrey E Hinton. Using expectation-maximization for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

Shengbo Guo and Scott Sanner. Real-time multiattribute bayesian preference elicitation with pairwise comparison queries. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 289–296, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/guo10b.html.

Nikolaus Hansen, Sibylle D Müller, and Petros Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003.

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. Towards efficient and exact optimization of language model alignment. *arXiv preprint arXiv:2402.00856*, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas Lampe, Abbas Abdolmaleki, Sarah Bechtle, Sandy H Huang, Jost Tobias Springenberg, Michael Bloesch, Oliver Groth, Roland Hafner, Tim Hertweck, Michael Neunert, et al. Mastering stacking of diverse shapes with large-scale iterative reinforcement learning on real robots. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7772–7779. IEEE, 2024.

Alex X. Lee, Coline Devin, Yuxiang Zhou, Thomas Lampe, Konstantinos Bousmalis, Jost Tobias Springenberg, Arunkumar Byravan, Abbas Abdolmaleki, Nimrod Gileadi, David Khosid, Claudio Fantacci, José Enrique Chen, Akhil Raju, Rae Jeong, Michael Neunert, Antoine Laurens, Stefano Saliceti, Federico Casarini, Martin A. Riedmiller, Raia Hadsell, and Francesco Nori. Beyond pick-and-place: Tackling robotic stacking of diverse shapes. *CoRR*, abs/2110.06192, 2021. URL https://arxiv.org/abs/2110.06192.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018. URL https://arxiv.org/abs/1805.00909.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano,

Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *CoRR*, abs/1910.00177, 2019. URL http://arxiv.org/abs/1910.00177.

Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 24, pp. 1607–1612, 2010.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. In *R:SS 2021*, 2013.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, Jack Krawczyk, Cosmo Du, Ed Chi, Heng-Tze Cheng, Eric Ni, Purvi Shah, Patrick Kane, Betty Chan, Manaal Faruqui, Aliaksei Severyn, Hanzhao Lin, YaGuang Li, Yong Cheng, Abe Ittycheriah, Mahdis Mahdieh, Mia Chen, Pei Sun, Dustin Tran, Sumit Bagri, Balaji Lakshminarayanan, Jeremiah Liu, Andras Orban, Fabian Güra, Hao Zhou, Xinying Song, Aurelien Boffy, Harish Ganapathy, Steven Zheng, HyunJeong Choe, Ágoston Weisz, Tao Zhu, Yifeng Lu, Siddharth Gopal, Jarrod Kahn, Maciej Kula, Jeff Pitman, Rushin Shah, Emanuel Taropa, Majd Al Merey, Martin Baeuml, Zhifeng Chen, Laurent El Shafey, Yujing Zhang, Olcan Sercinoglu, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam

Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Gaurav Singh Tomar, Evan Senter, Martin Chadwick, Ilya Kornakov, Nithya Attaluri, Iñaki Iturrate, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Xavier Garcia, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Ravi Addanki, Antoine Miech, Annie Louis, Denis Teplyashin, Geoff Brown, Elliot Catt, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaliy Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer, Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Caglar Unlu, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze

Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Sidharth Mudgal, Romina Stella, Kevin Brooks, Gautam Vasudevan, Chenxi Liu, Mainak Chain, Nivedita Melinkeri, Aaron Cohen, Venus Wang, Kristie Seymore, Sergey Zubkov, Rahul Goel, Summer Yue, Sai Krishnakumaran, Brian Albert, Nate Hurley, Motoki Sano, Anhad Mohananey, Jonah Joughin, Egor Filonov, Tomasz Kępa, Yomna Eldawy, Jiawern Lim, Rahul Rishi, Shirin Badiezadegan, Taylor Bos, Jerry Chang, Sanil Jain, Sri Gayatri Sundara Padmanabhan, Subha Puttagunta, Kalpesh Krishna, Leslie Baker, Norbert Kalb, Vamsi Bedapudi, Adam Kurzrok, Shuntong Lei, Anthony Yu, Oren Litvin, Xiang Zhou, Zhichun Wu, Sam Sobell, Andrea Siciliano, Alan Papir, Robby Neale, Jonas Bragagnolo, Tej Toor, Tina Chen, Valentin Anklin, Feiran Wang, Richie Feng, Milad Gholami, Kevin Ling, Lijuan Liu, Jules Walter, Hamid Moghaddam, Arun Kishore, Jakub Adamek, Tyler Mercado, Jonathan Mallinson, Siddhinita Wandekar, Stephen Cagle, Eran Ofek, Guillermo Garrido, Clemens Lombriser, Maksim Mukha, Botu Sun, Hafeezul Rahman Mohammad, Josip Matak, Yadi Qian, Vikas Peswani, Pawel Janus, Quan Yuan, Leif Schelin, Oana David, Ankur Garg, Yifan He, Oleksii Duzhyi, Anton Älgmyr, Timothée Lottaz, Qi Li, Vikas Yadav, Luyao Xu, Alex Chinien, Rakesh Shivanna, Aleksandr Chuklin, Josie Li, Carrie Spadine, Travis Wolfe, Kareem Mohamed, Subhabrata Das, Zihang Dai, Kyle He, Daniel von Dincklage, Shyam Upadhyay, Akanksha Maurya, Luyan Chi, Sebastian Krause, Khalid Salama, Pam G Rabinovitch, Pavan Kumar Reddy M, Aarush Selvan, Mikhail Dektiarev, Golnaz Ghiasi, Erdem Guven, Himanshu Gupta, Boyi Liu, Deepak Sharma, Idan Heimlich Shtacher, Shachi Paul, Oscar Akerlund, François-Xavier Aubet, Terry Huang, Chen Zhu, Eric Zhu, Elico Teixeira, Matthew Fritze, Francesco Bertolini, Liana-Eleonora Marinescu, Martin Bölle, Dominik Paulus, Khyatti Gupta, Tejasi Latkar, Max Chang, Jason Sanders, Roopa Wilson, Xuewei Wu, Yi-Xuan Tan, Lam Nguyen Thiet, Tulsee Doshi, Sid Lall, Swaroop Mishra, Wanming Chen, Thang Luong, Seth Benjamin, Jasmine Lee, Ewa Andrejczuk, Dominik Rabiej, Vipul Ranjan, Krzysztof Styrc, Pengcheng Yin, Jon Simon, Malcolm Rose Harriott, Mudit Bansal, Alexei Robsky, Geoff Bacon, David Greene, Daniil Mirylenka, Chen Zhou, Obaid Sarvana, Abhimanyu Goyal, Samuel Andermatt, Patrick Siegler, Ben Horn, Assaf Israel, Francesco Pongetti, Chih-Wei "Louis" Chen, Marco Selvatici, Pedro Silva, Kathie Wang, Jackson Tolins, Kelvin Guu, Roey Yogev, Xiaochen Cai, Alessandro Agostini, Maulik Shah, Hung Nguyen, Noah Ó Donnaile, Sébastien Pereira, Linda Friso, Adam Stambler, Adam Kurzrok, Chenkai Kuang, Yan Romanikhin, Mark Geller, ZJ Yan, Kane Jang, Cheng-Chun Lee, Wojciech Fica, Eric Malmi, Qijun Tan, Dan Banica, Daniel Balle, Ryan Pham, Yanping Huang, Diana Avram, Hongzhi Shi, Jasjot Singh, Chris Hidey, Niharika Ahuja, Pranab Saxena, Dan Dooley, Srividya Pranavi Potharaju, Eileen O'Neill, Anand Gokulchandran, Ryan Foley, Kai Zhao, Mike Dusenberry, Yuan Liu, Pulkit Mehta, Ragha Kotikalapudi, Chalence Safranek-Shrader, Andrew Goodman, Joshua Kessinger, Eran Globen, Prateek Kolhar, Chris Gorgolewski, Ali Ibrahim, Yang Song, Ali Eichenbaum, Thomas Brovelli, Sahitya Potluri, Preethi Lahoti, Cip Baetu, Ali Ghorbani, Charles Chen, Andy Crawford, Shalini Pal, Mukund Sridhar, Petru Gurita, Asier Mujika, Igor Petrovski, Pierre-Louis Cedoz, Chenmei Li, Shiyuan Chen, Niccolò Dal Santo, Siddharth Goyal, Jitesh Punjabi, Karthik Kappaganthu, Chester Kwak, Pallavi LV, Sarmishta Velury, Himadri Choudhury, Jamie Hall, Premal Shah, Ricardo Figueira, Matt Thomas, Minjie Lu, Ting Zhou, Chintu Kumar, Thomas Jurdi, Sharat Chikkerur, Yenai Ma, Adams Yu, Soo Kwak, Victor Ähdel, Sujeevan Rajayogam, Travis Choma, Fei Liu, Aditya Barua, Colin Ji, Ji Ho Park, Vincent Hellendoorn, Alex Bailey, Taylan Bilal, Huanjie Zhou, Mehrdad Khatir, Charles Sutton, Wojciech Rzadkowski, Fiona Macintosh, Konstantin Shagin, Paul Medina, Chen Liang, Jinjing Zhou, Pararth Shah, Yingying Bi, Attila Dankovics, Shipra Banga, Sabine Lehmann, Marissa Bredesen, Zifan Lin, John Eric Hoffmann, Jonathan Lai, Raynald Chung, Kai Yang, Nihal Balani, Arthur Bražinskas, Andrei Sozanschi, Matthew Hayes, Héctor Fernández Alcalde, Peter Makarov, Will Chen, Antonio Stella, Liselotte Snijders, Michael Mandl, Ante Kärrman, Paweł Nowak, Xinyi Wu, Alex Dyck, Krishnan Vaidyanathan, Raghavender R, Jessica Mallet, Mitch Rudominer, Eric Johnston, Sushil Mittal, Akhil Udathu, Janara Christensen, Vishal Verma, Zach Irving, Andreas Santucci, Gamaleldin Elsayed, Elnaz Davoodi, Marin Georgiev, Ian Tenney, Nan Hua, Geoffrey Cideron, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Dylan Scandinaro, Heinrich Jiang, Jasper Snoek, Mukund Sundararajan, Xuezhi Wang, Zack Ontiveros, Itay Karo, Jeremy Cole, Vinu Rajashekhar, Lara Tumeh, Eyal Ben-David, Rishub Jain, Jonathan Uesato, Romina Datta, Oskar Bunyan, Shimu Wu, John Zhang, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam

Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Jane Park, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Geoffrey Irving, Edward Loper, Michael Fink, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Ivan Petrychenko, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Evan Palmer, Paul Suganthan, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Ginger Perng, Elena Allica Abellan, Mingyang Zhang, Ishita Dasgupta, Nate Kushman, Ivo Penchev, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Daniel Andor, Pedro Valenzuela, Minnie Lui, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Ken Franko, Anna Bulanova, Rémi Leblond, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Mark Omernick, Colton Bishop, Rachel Sterneck, Rohan Jain, Jiawei Xia, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Daniel J. Mankowitz, Alex Polozov, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Matthieu Geist, Ser tan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Kathy Wu, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Saaber Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Yeqing Li, Nir Levine, Ariel Stolovich, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Charlie Deck, Hyo Lee, Zonglin Li, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Sho Arora, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Lynette Webb, Sahil Dua, Dong Li, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Evgenii Eltyshev, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Christof Angermueller, Xiaowei

Li, Anoop Sinha, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Denny Zhou, Komal Jalan, Dinghua Li, Blake Hechtman, Parker Schuh, Milad Nasr, Kieran Milan, Vladimir Mikulik, Juliana Franco, Tim Green, Nam Nguyen, Joe Kelley, Aroma Mahendru, Andrea Hu, Joshua Howland, Ben Vargas, Jeffrey Hui, Kshitij Bansal, Vikram Rao, Rakesh Ghiya, Emma Wang, Ke Ye, Jean Michel Sarr, Melanie Moranski Preston, Madeleine Elish, Steve Li, Aakash Kaku, Jigar Gupta, Ice Pasupat, Da-Cheng Juan, Milan Someswar, Tejvi M., Xinyun Chen, Aida Amini, Alex Fabrikant, Eric Chu, Xuanyi Dong, Amruta Muthal, Senaka Buthpitiya, Sarthak Jauhari, Nan Hua, Urvashi Khandelwal, Ayal Hitron, Jie Ren, Larissa Rinaldi, Shahar Drath, Avigail Dabush, Nan-Jiang Jiang, Harshal Godhia, Uli Sachs, Anthony Chen, Yicheng Fan, Hagai Taitelbaum, Hila Noga, Zhuyun Dai, James Wang, Chen Liang, Jenny Hamer, Chun-Sung Ferng, Chenel Elkind, Aviel Atias, Paulina Lee, Vít Listík, Mathias Carlen, Jan van de Kerkhof, Marcin Pikus, Krunoslav Zaher, Paul Müller, Sasha Zykova, Richard Stefanec, Vitaly Gatsko, Christoph Hirnschall, Ashwin Sethi, Xingyu Federico Xu, Chetan Ahuja, Beth Tsai, Anca Stefanoiu, Bo Feng, Keshav Dhandhania, Manish Katyal, Akshay Gupta, Atharva Parulekar, Divya Pitta, Jing Zhao, Vivaan Bhatia, Yashodha Bhavnani, Omar Alhadlaq, Xiaolin Li, Peter Danenberg, Dennis Tu, Alex Pine, Vera Filippova, Abhipso Ghosh, Ben Limonchik, Bhargava Urala, Chaitanya Krishna Lanka, Derik Clive, Yi Sun, Edward Li, Hao Wu, Kevin Hongtongsak, Ianna Li, Kalind Thakkar, Kuanysh Omarov, Kushal Majmundar, Michael Alverson, Michael Kucharski, Mohak Patel, Mudit Jain, Maksim Zabelin, Paolo Pelagatti, Rohan Kohli, Saurabh Kumar, Joseph Kim, Swetha Sankar, Vineet Shah, Lakshmi Ramachandruni, Xiangkai Zeng, Ben Bariach, Laura Weidinger, Tu Vu, Alek Andreev, Antoine He, Kevin Hui, Sheleem Kashem, Amar Subramanya, Sissie Hsiao, Demis Hassabis, Koray Kavukcuoglu, Adam Sadovsky, Quoc Le, Trevor Strohman, Yonghui Wu, Slav Petrov, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2024. URL https://arxiv.org/abs/2312.11805.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. ISSN 2665-9638. doi: https://doi.org/10.1016/j.simpa.2020.100022. URL https://www.sciencedirect.com/science/article/pii/S2665963820300099.

Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, and Nando de Freitas. Critic regularized regression. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7768–7778. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/588cb956d6bbe67078f29f8de420a13d-Paper.pdf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric. P Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. Lmsys-chat-1m: A large-scale real-world llm conversation dataset, 2023.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.