Scaling Laws for Reward Model Overoptimization in Direct Alignment Algorithms

Anonymous Author(s)
Affiliation
Address
email

Abstract

Reinforcement Learning from Human Feedback (RLHF) has been crucial to the recent success of Large Language Models (LLMs), however, it is often a complex and brittle process. In the classical RLHF framework, a reward model is first trained to represent human preferences, which is in turn used by an online reinforcement learning (RL) algorithm to optimize the LLM. A prominent issue with such methods is *reward over-optimization* or *reward hacking*, where performance as measured by the learned proxy reward model increases, but true quality plateaus or even deteriorates. Direct Alignment Algorithms (DDAs) like Direct Preference Optimization have emerged as alternatives to the classical RLHF pipeline by circumventing the reward modeling phase. However, although DAAs do not use separate proxy reward model, they still commonly deteriorate from over-optimization. While the so-called reward hacking phenomenon is not well-defined for DAAs, we still uncover similar trends: at higher KL-budgets, DAA algorithms exhibit similar degradation patterns to their classic RLHF counterparts. In particular, we find that DAA methods deteriorate not only across a wide range of KL-budgets, but also often before even a single epoch of the dataset is completed. Through extensive empirical experimentation, this work formulates and formalizes the reward over-optimization or hacking problem for DAAs and explores its consequences across objectives, training regimes, and model scales.

1 Introduction

Recent advancements in Large Language Models (LLMs) have broadened their capabilities significantly, enabling applications in code generation, mathematical reasoning, tool use, and interactive communication. These improvements have popularized LLMs across various domains. Reinforcement Learning from Human Feedback (RLHF) has been instrumental in these advances and is now integral to sophisticated LLM training regimes [10, 55]. Before alignment, LLMs, trained on vast text corpses to predict subsequent tokens [45, 8] are often unwieldy and hard to use. Today, leading LLMs incorporate variants of the RLHF framework [14, 68, 36] to align them with human intent, which generally involves a multi-stage process. Specifically, users evaluate model responses to assorted prompts in order to train a reward model that encapsulates human preferences [10, 55, 71, 5, 61]. Then, the refined LLM maximizes the expected learned reward function using a reinforcement learning (RL) algorithm [50, 1, 64]. Despite its efficacy, this procedure is complex and computationally intensive, particularly in its latter stages.

Goodhart's Law [25, 11], that "when a measure becomes a target, it ceases to be a good measure", has often been cited as a core shortcoming of RLHF. Standard RLHF methods optimize a learned, but imperfect reward function which ends up amplifying the reward model's shortcomings. Empirically, this phenomena was first extensively characterized by Gao et al. [21], who coined the term "reward

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

over-optimization", and has been seen consistently in recent findings [61, 16, 14]. While reward over-optimization has been studied in the context of the aforementioned RLHF procedure, recent contemporary methods for aligning LLMs circumvent the reward learning procedure, necessitating a new characterization of the over-optimization phenomena.

This new broad class of algorithms, which we refer to as Direct Alignment Algorithms (DAAs), bypass the traditional RLHF pipeline by re-parameterizing the reward model directly through the optimal policy derived during the reinforcement learning phase. DAA methods, like Direct Preference Optimization [46], have gained popularity [14, 28] as they often reduce computational demands. Yet, despite not fitting a reward function, DAAs still exhibit over-optimization trends similar to those of traditional RLHF methods using a learned reward function. In some sense, this is puzzling: DAAs can be viewed as simply learning a reward function with supervised learning from which the optimal policy is deterministically mapped, however more seems to be at play than simple supervised learning.

In this work we investigate the over-fitting phenomena present in DAA algorithms through extensive experimentation. First, we unify a number of different recent methods [46, 67, 4] under the DAA framework. Then, across different model scales and hyper-parameters, we show that DAAs exhibit a type of reward over-optimization consistent with that previously observed in RLHF [21]. Specifically, we find that at different KL-divergence budgets DAAs exhibit degradation patterns similar to those found in RLHF. Interestingly, we also find that performance within a single epoch is not always consistent as expected for DAAs. Finally, we explain why this happens by appealing to the under-constrainted nature of the optimization problem used in DAAs.

2 Preliminaries

In this section, we first outline the core components of the standard RLHF pipeline [71, 55, 5, 41]). Then, we examine prior literature to characterize the reward over-optimization exhibited by standard RLHF methods. Finally, we provide a unifying view of direct alignment algorithms (DAAs) which will guide our analysis of their training dynamics in the next section.

2.1 Reinforcement Learning From Human Feedback

The standard RLHF pipeline consists of three distinct stages with the goal of aligning the LLM with human preferences.

Supervised Fine Tuning (SFT): First, a dataset of prompts x and high-quality answers y are used to train an LLM for instruction following via maximum likelihood estimation over next-tokens. We refer to the resultant model as $\pi_{\text{SFT}}(y|x)$ and consider the entire prompt and answer strings to be single variables.

Reward Modeling: Second, the SFT model $\pi_{\text{SFT}}(y|x)$ is used to learn a reward function over human preferences. Specifically, the SFT model is queried to produce pairs of answers $(y_1, y_2) \sim \pi_{\text{SFT}}(y|x)$, for every prompt x in a dataset. Then, users select their preferred answers, resulting in ranking $y_w \succ y_l \mid x$ where y_w and y_l are the preferred and dispreferred answers respectively. Typically, user rankings are assumed to be distributed according to the Bradley-Terry (BT) model [7]

$$p(y_1 \succ y_2 \mid x) = \frac{\exp\left(r(x, y_1)\right)}{\exp\left(r(x, y_1)\right) + \exp\left(r(x, y_2)\right)} = \sigma(r(x, y_1) - r(x, y_2)) \tag{1}$$

where the preference distribution p results from an unobserved latent reward $r(x, y)$, and σ is the logistic function. Given this model and a dataset of rankings, denoted $\mathcal{D} = \left\{x^{(i)}, y_w^{(i)}, y_l^{(i)}\right\}_{i=1}^{N}$, we can train a parameterized model $r_\phi(x, y)$ to predict the unobserved reward using maximum likelihood estimation. This yields the following loss function,

$$\mathcal{L}_{\text{rew}}(r_\phi) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]. \tag{2}$$

Reinforcement Learning (RL): The final stage of the standard RLHF pipeline uses the learned reward model $r_\phi(x, y)$ to update the LLM π_θ with an on-policy RL algorithm like PPO [50], optimizing the model to provide responses more preferred by human raters. The most common objective is

$$\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(.|x)}\left[r_\phi(x, y)\right] - \beta \mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x) \mid\mid \pi_{\text{ref}}(y|x)\right] \tag{3}$$

which enforces a Kullback-Leibler (KL) divergence penalty with a reference distribution $\pi_{\text{ref}}(y|x)$ (usually taken to be $\pi_{\text{SFT}}(y|x)$) to prevent the LLM π_θ from straying too far from its initialization. Thus, the hyper-parameter β directly trades off exploiting the reward function and deviating from $\pi_{\text{ref}}(y|x)$.

2.2 Reward Exploitation in RLHF

Unfortunately, repeating the above procedure without careful tuning of the RL phase can lead to disastrous performance. This is because in the context of RLHF the LLM policy is optimizing the surrogate reward estimate $r_\phi(x, y)$ and not the true reward function as is often the case in other domains. Thus, prior works have observed that while the LLM's expected reward according to eq. (3) increases the actual quality of the model's outputs can decrease [54, 43, 9, 34]. This particular instantiation of the reward exploitation or hacking problem [3] is often referred to as reward "over-optimization" in RLHF literature and has been studied empirically in both controlled experiments [21] and user studies [14]. There are two prevailing explanations for why this phenomena occurs.

1. OOD Robustness: In the classical RLHF pipeline, the RL objective (eq. (3)) is optimized using on-policy samples from π_θ. This means that the reward function is continuously queried using unseen model samples which are potentially out-of-distribution. Beyond the support of the reward modeling distribution, r_ϕ may assign high reward to sub-par responses, leading the policy to believe it is doing well when it may not be. While the KL-regularization term is designed to prevent the model from drifting too far out of distribution, this term alone has proven inadequate to prevent reward hacking [21].

2. Reward Mis-specification. Learned reward functions may exhibit spurious correlations that cause them to prefer unintended behaviors. While this issue is not at the forefront of LLM research, it is known to be pervasive in RL [43, 34]. Most efforts to address these problems exist at the intersection of robustness and offline RL literature [13, 66, 16] and use measures of epistemic uncertainty to penalize the predicted reward.

2.3 Direct Alignment Algorithms

Due to its complex multi-step nature, recent works have sought alternatives to the classic RLHF pipeline. A new class of algorithms, which we broadly classify as Direct Alignment Algorithms (DAAs), directly update the LLM's policy π_θ using user feedback instead of fitting a reward function to it and then employing an RL algorithm. Perhaps the most known example is Direct Preference Optimization (DPO). DPO as well as other DAAs are derived using the closed form solution to the RLHF objective in eq. (3) [70], $\pi^*(y|x) \propto \pi_{\text{ref}}(y|x)e^{r(x,y)/\beta}$, where $r(x, y)$ is the ground-truth reward. By isolating $r(x, y)$ in this relationship and substituting it into the reward optimization objective in eq. (2), we arrive at a general objective that allows us to train the LLM directly using feedback data:

$$\mathcal{L}_{\text{DAA}}(\pi_\theta; \pi_{\text{ref}}) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[g\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right] \tag{4}$$

where g is a convex loss function. Using $g(x) = -\log \sigma(x)$ coincides with the standard Bradley-Terry model and the original DPO objective. Other methods choose different loss functions: IPO [4] uses the quadratic objective $g(x) = (x - 1)^2$ and SLiC-HF [67, 38] uses the hinge loss $g(x) = \max(0, 1 - x)$. Additional objectives were also considered in [59], but due to limited computational resources, we focus on the three objectives outlined above.

Crucially, the DAA approach allows us to recover the optimal policy using a straightforward classification loss without the need for learning a reward function, on-policy sampling, or RL, which can be notoriously difficult to tune and computationally expensive. Because of this, DAAs have emerged as a popular alternative. However, just like classical RLHF methods, DAAs exhibit strong over-fitting and even reward-hacking like behaviors. For example, Park et al. [44] show that LLMs trained with DPO generate responses with increasing length throughout the course of training, but do not improve in ground-truth win-rate after a certain point. Since DAAs do not explicitly learn a reward function, it is unclear how "reward-overoptimization" fits into the picture. In this work, we aim to shed some light on this phenomena in DAAs.

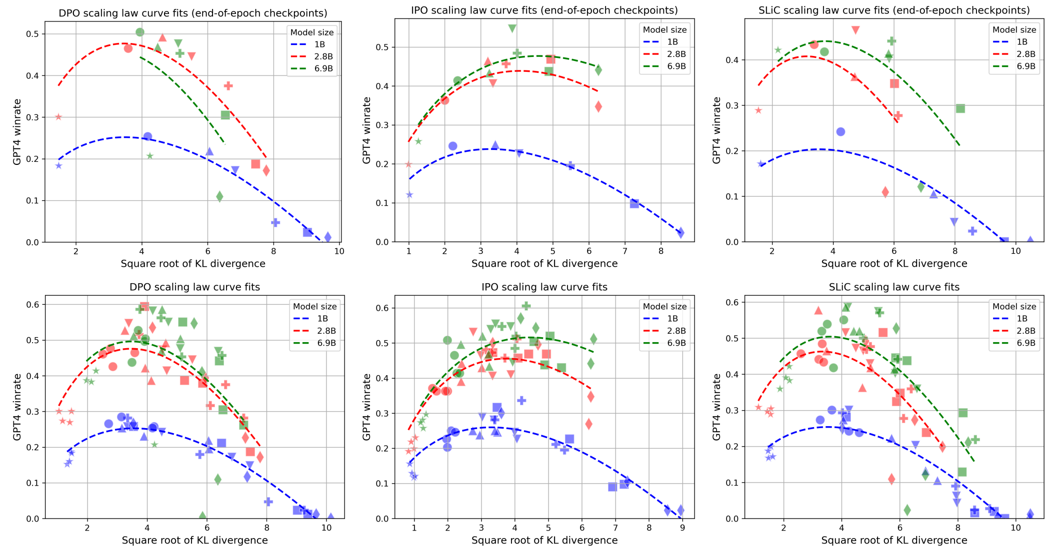


Figure 1: Results on over-optimization in Direct Alignment Algorithms for DPO, IPO and SLiC. Results shows model win-rates over the dataset summary on an evaluation set of prompts as judged by GPT-4. The top row shows final performance after 1 epoch of training, while the second row also includes 4 intermediate checkpoints as well. The fitted dotted curves utilize scaling laws from [21] applied to direct alignment, with GPT4 winrates taking the place of the gold reward model score.

3 Empirical Analysis of Overoptimization in DAAs

First, we examine the over-optimization problem in DAAs and compare it to those observed in traditional RLHF methods. All our experiments are carried using the Reddit TL;DR summarization dataset [55] and the Pythia family of Large Language Models [6].

3.1 Evaluating Model-Overoptimization

In our first set of experiments we evaluate the reward model over-optimization phenomenon. We evaluate three training objectives DPO, IPO and SLiC using seven β parameters, representing different KL budgets at three model size - 1B, 2.8B and 6.9B. Our main results are shown in Fig. 1 which presents results for different configurations after 1 epoch of training (row 1) and including 4 uniform intermediate checkpoints (row 2). We include additional results on the training dynamics in Fig. 2, which shows win rates and KL bounds for intra-epoch training. We present our findings below.

Model Over-Optimization: We see clear over-optimization for all objectives as performance exhibits a hump-shaped pattern, where an additional increase in the KL budget leads to decreasing model performance. Moreover in Fig. 2 we observe similar intra-epoch training dynamics patterns as configurations with wider KL budgets achieve their best performance after training on only 25% of the data, after which performance starts decreasing in conjunction with increasing KL divergence metrics.

Effect of Training Objective: In the IPO work [4] the authors present theoretical arguments that due to the monotone sigmoid objective in the DPO formulation, the KL constraint is not effectively enforced and propose the quadratic fixed-margin loss as an alternative. Across all objectives, there are clear dependencies between the β parameter and the corresponding KL achieved at the end of training. While DPO and SLiC exhibit similar performance, IPO indeed seems to be less prone to over-optimization and in general achieve lower KLs under the same constraint. Our observations with IPO also align with prior works in preference-based RL and imitation learning where imposing a fixed margin led to more stable and performant methods [48, 51].

Effect of Model Size: The results also show strong parameter count scaling effect. The Pythia 1B model achieves low performance under the same set of constraints it reaches much higher KL values, while almost immediately exhibiting signs of over-optimization. This behavior holds under all three

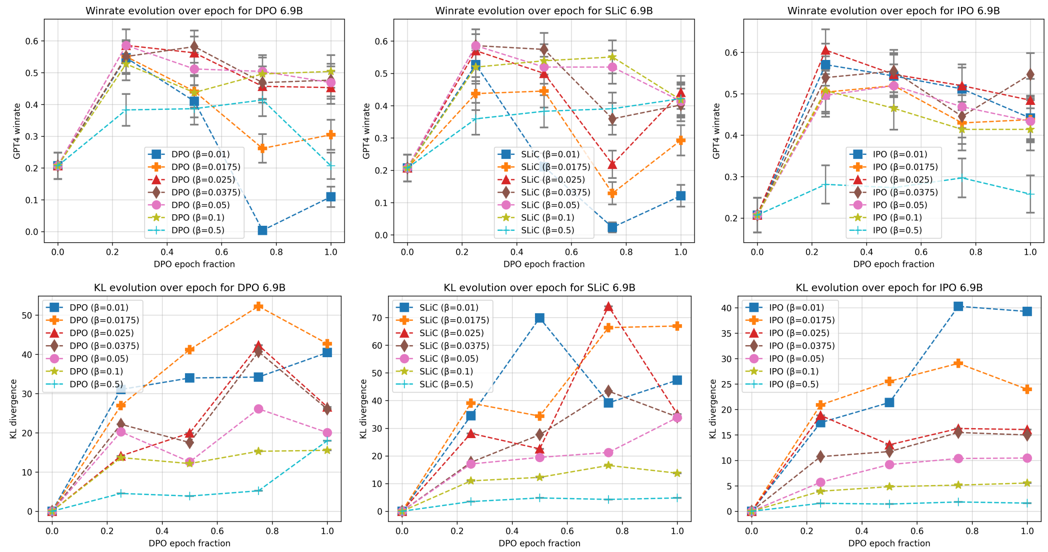


Figure 2: Results on intra-epoch optimization dynamics. The top row shows win-rates against fraction of an epoch so far, while the bottom row shows the corresponding KL values. Under a lower KL constraint most experiments reach their best performance in the first 25% of the epoch and degrade over the remaining of training, while the model deviates from the reference under increasing KL. All models are 6.9B and vary across DPO, SLiC, and IPO loss formulations.

objectives. At larger scales, the 6.9B Pythia model tends to exhibit more win-rate - KL trade-offs and be less prone to over-optimization, with both models significantly outperforming the 1B model. In the case of the IPO objective, the 6.9B also exhibits significantly better control over the KL objective and shows little to no over-optimization behavior.

3.2 Scaling Law Fits

Given we have established a framework for evaluating over-optimization in DAAs and empirically validated it (section 3.1), we now develop scaling laws for this phenomenon. Previous work in classical RLHF have established such scaling laws for reward model scores as a function of the KL divergence between initial and optimized policies [21]. The relevant functional of the reward $R(d)$ is

$$R(d) = d(\alpha - \beta \log d) \tag{5}$$

where α, β are constants dependent on the size of the reward model dataset and parameter count, and $d = \sqrt{D_{\text{KL}}(\pi || \pi_{\text{ref}})}$. As DAAs do not train a proxy reward model, we treat GPT4 winrates over dataset completions as a proxy for gold reward. Somewhat surprisingly, we find that this scaling law accurately relates d and winrates for DAAs. Compared to a quadratic fit between $D_{\text{KL}}(\pi || \pi_{\text{ref}})$ and winrates, this scaling law halves the RMSE. It is worth noting, however, that a quadratic fit between d and winrates yields similar error compared to Equation 5.

3.3 Length Correlations

Prior work [44] has shown that the DPO algorithm is prone to length exploitation as it amplifies verbosity biases in preference datasets. Here we show that length is not the only dimension on which exploitation can occur. Our experimental results are shown in Fig. 3. On the left we show results for the 2.8B Pythia model with standard training plus the length-regularization approach from [44]. Both approaches suffer from over-optimization, but the dynamics differ depending on the KL budget. Moreover, even though the regularized model achieves higher win rates on a length-correct basis, it under-performs the model trained with the standard objective in the lower KL constraint region.

Recent work [27] has also shown that DAAs prioritize features of the data based on their complexity and prevalence (with length a clear example of human datasets). [44] further showed that models trained with the DPO algorithm extrapolate significantly based on length. We extend this analysis in

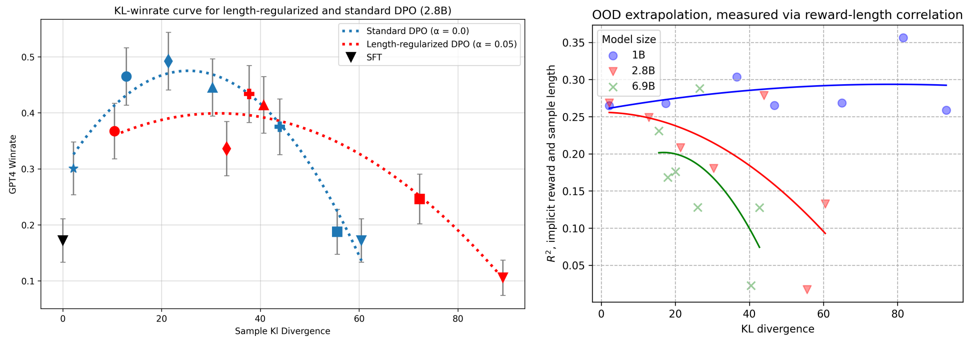


Figure 3: **Left:** KL budget versus win-rates (over dataset human answer) with and without length-regularization [44]. While including a length-correction in the optimization objective changes the KL-win-rate Pareto Frontier, it does not alleviate reward over-optimization and might even exacerbate it. **Right:** Scaling behaviour for length extrapolation - smaller capacity models (either by size or KL budget) extrapolate more strongly on simpler features such as length.

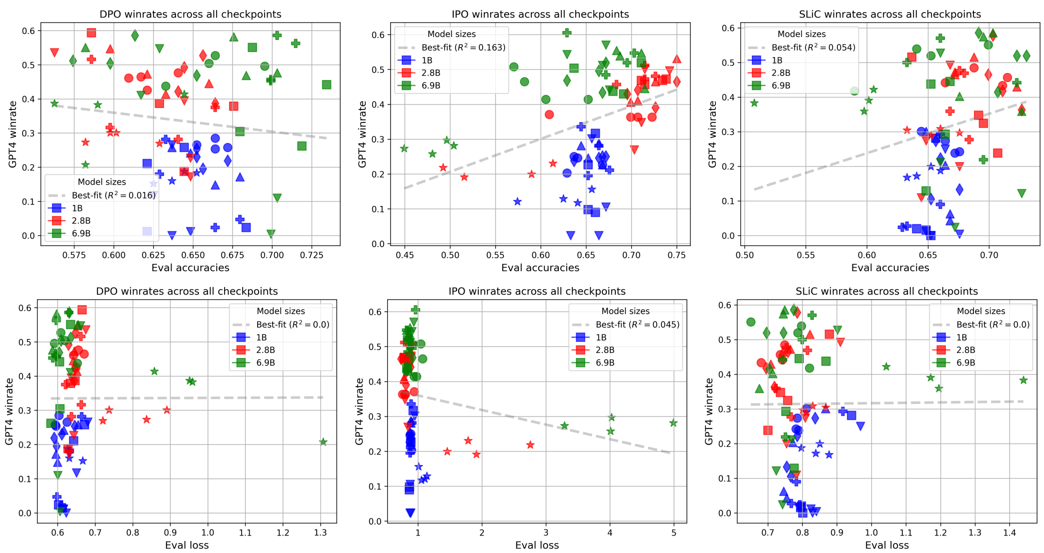


Figure 4: **Top:** We plot the DAA implicit reward accuracy in preference classification versus win rates. **Bottom:** DAA optimization loss versus checkpoint win rates. Model training statistics, do not exhibit strong relationship with downstream performance.

Fig, 3 (right). We consider a linear regression of the form

$$\log \frac{\pi_\theta(y^{(i)}|x^{(i)})}{\pi_{ref}(y^{(i)}|x^{(i)})} = \hat{\gamma}|y^{(i)}| + \epsilon^{(i)} \tag{6}$$

where $x^{(i)}$ are held-out prompts and $y^{(i)}$ are samples from the corresponding model between the DPO implicit reward and length. We fit a different regression for each model size and checkpoint and plot the corresponding R^2 values. We observe two main effects; first there is a clear scaling law behaviour. Weaker models extrapolate across the simple length feature to a much higher degree than stronger ones. This is especially clear comparing the behaviour of the Pythia 1B versus the 2.8B and 6.9B models. Second, we see significant effects based on the KL budget - under a smaller budget all model sizes exhibit higher extrapolation behaviour. Based on these results we formulate the hypothesis that under limited capacity, either from model capability or limited KL budgets the model will extrapolate more strongly based on simpler features, which can lead to OOD issues.

3.4 Reward Metrics Correlations

Prior works have measured reward model quality in ranking settings by classification accuracy. We evaluate the relationship between the DAA implicit reward model accuracy and policy performance

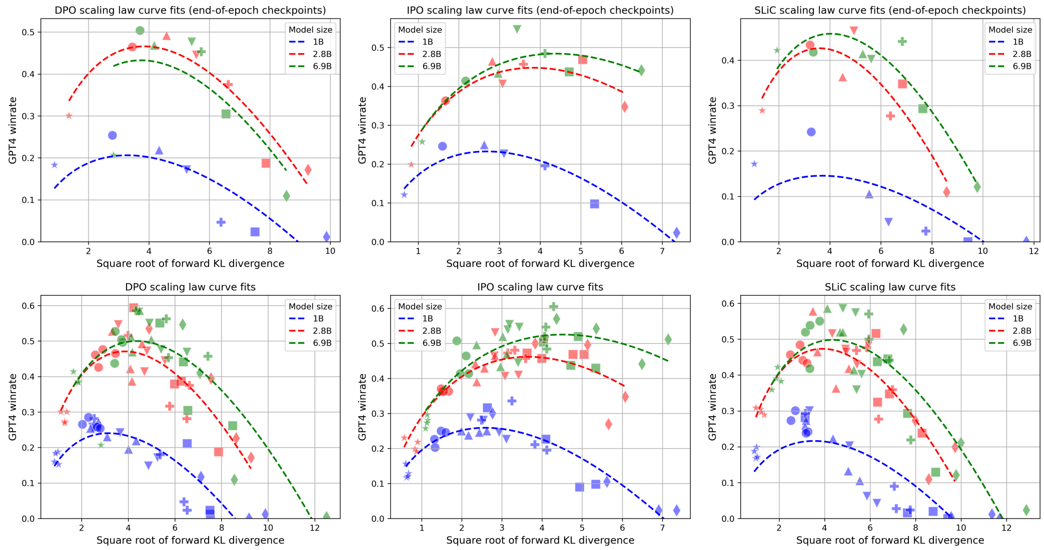


Figure 5: Over-optimization results for $\sqrt{\text{Forward KL}}$ vs. winrates. The top row shows final performance after 1 epoch of training, while the second row also includes 4 intermediate checkpoints. The fitted dotted curves are scaling laws from [21] applied to DAAs, with GPT4 winrates taking the place of the gold reward model score.

in Figure 4. The DPO and SLiC algorithms exhibit little to no correlation between reward model accuracy and downstream model performance. The IPO model shows a weak positive relationship, but upon further examinations, this is entirely due to model size scaling - stronger models both fit the data better and produce better generations as well, however within each particular model size, there is no discernible relationship between the DAA implicit reward accuracy and the actual policy performance. Similar observations hold when comparing the empirical DAA loss with model performance, which is contrary to observations in supervised pre-training and instruction tuning [30].

3.5 Decreasing Likelihoods and Model Performance

A number of recent works have observed that the implicit DAA rewards of both preferred and dis-preffered responses decrease doing training, which may be counter-intuitive. In [47] the authors make a counter-point that in offline training of DAAs π_{ref} is usually pre-trained with SFT on the preferred response and thus

$$\mathbb{E}_{p_{\mathcal{D}}(y_w|x)}\left[\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right] \approx \mathbb{E}_{\pi_{\text{ref}}(y_w|x)}\left[\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right] = -\mathbb{D}_{\text{KL}}\big[\pi_{\text{ref}}(y|x)\ ||\ \pi_\theta(y\mid x)\big] \quad (7)$$

where $p_{\mathcal{D}}(y^w|x)$ is the dataset distribution of preferred answers. That is the expected implicit reward represent a forward KL divergence between the reference policy and the optimisation policy, thus it is expected to be negative and decrease with training as the optimisation model moves away from the reference. In this section we study whether this empirical phenomenon presents a challenge for DAA learning. Similar to Fig. 1 we plot the win rates against the square-root-transformed (negative) expected implicit reward of the preferred response (evaluated on a held-out evaluation dataset), which as stated above approximates the (square-root-transformed) forward KL $\mathbb{D}_{\text{KL}}\big[\pi_{\text{ref}}(y|x)\ ||\ \pi_\theta(y\mid x)$. Results are included in Fig. 5, which follow closely the pattern in Fig. 1 with performance initially increasing before it starts dipping down after a certain threshold. This indicates that under the standard DAA training pipeline decreasing likelihoods are not necessary an issue for performance, and are even necessary for improvement, but exhibit a non-linear over-optimization dynamics.

4 Reward Exploitation in Direct Alignment Algorithms

While the phenomena observed in the previous section echo those observed in classical RLHF, their underlying causes may be distinct. Reward over-optimization in classical RLHF is largely attributed to querying a proxy reward function that is potentially OOD, while DAAs do not train a separate reward model. Instead, DAAs are generally understood as fitting an "implicit" reward model to preference data with the parameterization $r_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ using the objective in eq. (2). Therefore, the OOD behavior of the policy is inextricably linked to the OOD behavior of the implicit reward model. Here we demonstrate that the reward modeling objective used is heavily under-constrained, allowing for a potentially large number of solutions that can place weight on OOD responses. This is especially problematic for DAAs which deterministically map the optimal policy from the "implicit" reward.

Rank Deficiency with Finite Preferences. In DAAs, the language modeling problem is treated as contextual bandit. However, the space of possible prompts $x \in \mathcal{X}$ and answers $y \in \mathcal{Y}$ are both exponentially large in sequence length. However, as highlighted by Tang et al. [59], DAAs often assume full support of the reference distribution when mapping from the implicit reward to the optimal policy π by eq. (10). However, in practice such coverage is impossible. Instead, preference datasets cover a minuscule portion of the prompt-response space. Unfortunately, as DAA objectives are not strictly convex, this means that many optima of eq. (4) can place a high weight on OOD responses.

We can demonstrate this using the regression interpretation from Hejna et al. [23]. Consider re-writing the DAA objective from eq. (4) in terms of preference query vectors q which select the win response pair (x, y^w) and the loss response pair (x, y^l) from the prompt-response space. Each vector q represents both the preferred and dis-preferred responses, with the entree corresponding to (x, y^w) being +1 and the entree corresponding to (x, y^l) being -1. We can then write the generalized DAA loss function with finite preference data as

$$\mathcal{L}_{\text{DAA}}(\pi_\theta, \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} g\left(\beta q_i^\top (\log \pi(y|x) - \log \pi_{\text{ref}}(y|x))\right), \text{ where } q_i[x, y] = \begin{cases} 1 & \text{if } (x^{(i)}, y_w^{(i)}) = (x, y) \\ -1 & \text{if } (x^{(i)}, y_l^{(i)}) = (x, y) \\ 0 & \text{otherwise} \end{cases}$$

where the policy π is a single vector of size $|\mathcal{X} \times \mathcal{Y}|$. In practice, the distributional constraint of π also applies. Choosing g to be the negative log sigmoid above recovers DPO with finite preferences, but also logistic regression with a data matrix Q of shape $|\mathcal{D}|$ by $|\mathcal{X} \times \mathcal{Y}|$ constructed by stacking the aforementioned query vectors q. As $|\mathcal{X} \times \mathcal{Y}| >> |\mathcal{D}|$, this matrix is likely to have a non-trivial null-space, making the problem not strictly convex. Thus, there are many possible policies π that can thus achieve the same optima, some of which can place a high weight on out-of-distribution responses [23, 69]. For more details and constructions, we defer to Hejna et al. [23].

Understanding OOD behavior for DAA algorithms with a Toy MDP: To illustrate that DAA algorithms, in general and not an artifact of training LLM's, end up placing probability mass on OOD sequences during training we design a simple Tree MDP (shown in Figure 6) to mimic the token-level MDP in LLMs. We use a dataset containing a single preference between two trajectories and follow the standard procedure of running SFT on preferred responses before updating an RNN policy using a DAA. Figure 7 shows that even in this simple setup, popular DAAs (DPO/IPO/SLiC) end up extrapolating incorrectly out of distribution revealing a fundamental shortcoming. Unlike in standard RLHF, the non-strict convexity of the reward function in DAAs ends up directly affecting the policy. Detailed experimental details can be found in Appendix E.

5 Related Work

Broadly, over-optimization has been a widely studied phenomena across different settings [60, 18]. Over-fitting can be characterized as over-optimization in the supervised learning setting [39, 32], which can harm generalization [19, 12, 24] or lead to susceptibility to adversarial attacks [56, 37, 15]. Reward hacking in reinforcement learning (RL) [54], where an agent maximizes its reward through behavior that deviates from the intended goal, can be viewed as a different type of over-optimization, commonly observed in prior work [43, 3, 22].

We study over-optimization in the context of aligning LLMs with human feedback, for which the most common approach is RLHF as outlined in section 2.1. Similar RLHF techniques were originally pioneered for control [31, 2, 10]. Standard RLHF methods suffer from both potential over-fitting of the reward function and reward exploitation by the RL algorithm. Several works have considered how to reduce over-fitting or increase the robustness of learned reward functions using ensembles [13, 66, 16] or data smoothing [69]. Other approaches, like Moskovitz et al. [40] consider how reward exploitation can be reduced by using different optimization techniques in the RL stage. Much

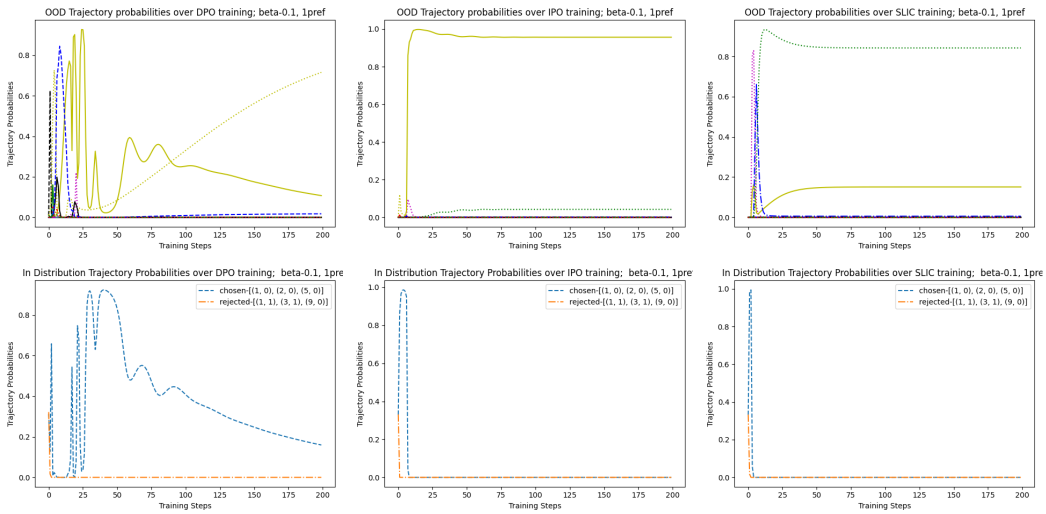


Figure 7: (Top row) Probability of OOD trajectories. DAA algorithms end up placing a substantial probability mass of some of the OOD trajectories during training. (Bottom row) Probability of in-distribution (preference-pair) trajectories decrease during training.

.

of this work is motivated by Gao et al. [21], which first characterized and provided scaling laws for over-optimization in RLHF.

Unlike Gao et al. [21], we consider the over-optimization problem in DAAs, which differ significantly from the standard RLHF pipeline. Different DAAs have been derived theoretically [47, 46, 67, 4, 63], and applied to problems beyond language modeling like image generation [62] and control [23]. In all of these scenarios, over-optimization problems have persisted. Park et al. [44] show that DAAs commonly over-fit to length and the expense of performance, which has been linked to inherent bias in training data [53, 29]. Other works have tried to allow DAAs to use more types of data like demonstrations [49] or ratings [17] to get better performance. Recently, incorporating online data has proven critical to improving performance [65, 26, 57]. Concurrent to our work, Tang et al. [58] study the differences between offline DAAs and standard RLHF methods. Unlike us, they focus on comparisons with online sampling whereas we focus on the purely offline setting.

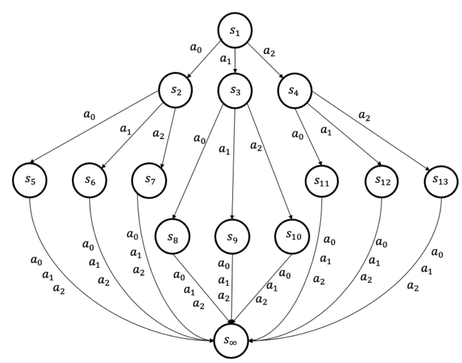


Figure 6: An illustration of the Tree MDP. At each state, we can choose one of 3 actions (a_0, a_1, a_2), which deterministically maps to the next state. Furthermore, all the leaf nodes in this tree MDP, transition to the terminal absorbing state s_∞, irrespective of the chosen action

6 Conclusion

In this work we present an analysis of the over-optimization problem in Direct Alignment Algorithms. Through extensive experimentation on different algorithms (DPO, IPO, SLIC) and at different model scales (1B, 2.8B, 6.9B), we observe consistent over-optimization trends at different KL-divergence budgets. While our analysis is a first step, it is not a complete picture of understanding the over-optimization phenomena. More work can be done characterizing this effect at larger model scales, which we were unable to do due to computational limitations. Nevertheless, we believe our work sheds light on important problems in Direct Alignment Algorithms that can spur future research.

References

[1] A. Ahmadian, C. Cremer, M. Gallé, M. Fadaee, J. Kreutzer, A. Üstün, and S. Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

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

[3] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

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

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

[6] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O'Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, A. Skowron, L. Sutawika, and O. van der Wal. Pythia: A suite for analyzing large language models across training and scaling, 2023.

[7] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: https://doi.org/10.2307/2334029.

[8] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] S. Casper, X. Davies, C. Shi, T. K. Gilbert, J. Scheurer, J. Rando, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Bıyık, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023.

[10] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf`.

[11] J. Clark and D. Amodei. Faulty reward functions in the wild, 2016. URL `https://openai.com/research/faulty-reward-functions`.

[12] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1282–1289. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/cobbe19a.html`.

[13] T. Coste, U. Anwar, R. Kirk, and D. Krueger. Reward model ensembles help mitigate overoptimization, 2023.

[14] Y. Dubois, X. Li, R. Taori, T. Zhang, I. Gulrajani, J. Ba, C. Guestrin, P. Liang, and T. B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024.

[15] J. Ebrahimi, D. Lowd, and D. Dou. On adversarial examples for character-level neural machine translation. *arXiv preprint arXiv:1806.09030*, 2018.

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

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

[18] T. Everitt, V. Krakovna, L. Orseau, M. Hutter, and S. Legg. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1705.08417*, 2017.

[19] J. Farebrother, M. C. Machado, and M. Bowling. Generalization and regularization in dqn. *arXiv preprint arXiv:1810.00123*, 2018.

[20] S. Fujimoto, D. Meger, and D. Precup. Off-policy deep reinforcement learning without exploration, 2019.

[21] L. Gao, J. Schulman, and J. Hilton. Scaling laws for reward model overoptimization. *International Conference on machine Learning*, 2023.

[22] D. Hadfield-Menell, S. Milli, P. Abbeel, S. J. Russell, and A. Dragan. Inverse reward design. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/32fdab6559cdfa4f167f8c31b9199643-Paper.pdf.

[23] J. Hejna, R. Rafailov, H. Sikchi, C. Finn, S. Niekum, W. B. Knox, and D. Sadigh. Contrastive preference learning: Learning from human feedback without reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=iX1RjVQODj.

[24] D. Hernandez, J. Kaplan, T. Henighan, and S. McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

[25] K. Hoskin. The 'awful idea of accountability': inscribing people into the measurement of objects. *Accountability: Power, ethos and the technologies of managing*, 265, 1996.

[26] A. Hosseini, X. Yuan, N. Malkin, A. Courville, A. Sordoni, and R. Agarwal. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*, 2024.

[27] S. Im and Y. Li. Understanding the learning dynamics of alignment with human feedback, 2024.

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

[29] S. Kabir, D. N. Udo-Imeh, B. Kou, and T. Zhang. Who answers it better? an in-depth analysis of chatgpt and stack overflow answers to software engineering questions, 2023.

[30] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei. Scaling laws for neural language models, 2020.

[31] W. B. Knox and P. Stone. Tamer: Training an agent manually via evaluative reinforcement. In *2008 7th IEEE international conference on development and learning*, pages 292–297. IEEE, 2008.

[32] V. Krakovna and R. Kumar. Classifying specification problems as variants of goodhart's law, 8 2019. URL https://vkrakovna.wordpress.com/2019/08/19/classifying-specification-problems-as-variants-of-goodharts-law/.

[33] A. Kumar, A. Zhou, G. Tucker, and S. Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

[34] N. Lambert and R. Calandra. The alignment ceiling: Objective mismatch in reinforcement learning from human feedback, 2023.

[35] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

[36] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, B. Newman, B. Yuan, B. Yan, C. Zhang, C. Cosgrove, C. D. Manning, C. Ré, D. Acosta-Navas, D. A. Hudson, E. Zelikman, E. Durmus, F. Ladhak, F. Rong, H. Ren, H. Yao, J. Wang, K. Santhanam, L. Orr, L. Zheng, M. Yuksekgonul, M. Suzgun, N. Kim, N. Guha, N. Chatterji, O. Khattab, P. Henderson, Q. Huang, R. Chi, S. M. Xie, S. Santurkar, S. Ganguli, T. Hashimoto, T. Icard, T. Zhang, V. Chaudhary, W. Wang, X. Li, Y. Mai, Y. Zhang, and Y. Koreeda. Holistic evaluation of language models, 2023.

[37] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun. Tactics of adversarial attack on deep reinforcement learning agents. *arXiv preprint arXiv:1703.06748*, 2017.

[38] T. Liu, Y. Zhao, R. Joshi, M. Khalman, M. Saleh, P. J. Liu, and J. Liu. Statistical rejection sampling improves preference optimization, 2024.

[39] D. Manheim and S. Garrabrant. Categorizing variants of goodhart's law, 2019.

[40] T. Moskovitz, A. K. Singh, D. Strouse, T. Sandholm, R. Salakhutdinov, A. Dragan, and S. M. McAleer. Confronting reward model overoptimization with constrained RLHF. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=gkfUvn0fLU`.

[41] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. F. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf`.

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

[43] A. Pan, K. Bhatia, and J. Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *International Conference on Learning Representations*, 2022.

[44] R. Park, R. Rafailov, S. Ermon, and C. Finn. Disentangling length from quality in direct preference optimization, 2024.

[45] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners, 2019. OpenAI.

[46] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. 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`.

[47] R. Rafailov, J. Hejna, R. Park, and C. Finn. From r to q^*: Your language model is secretly a q-function, 2024.

[48] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736, 2006.

[49] M. Rita, F. Strub, R. Chaabouni, P. Michel, E. Dupoux, and O. Pietquin. Countering reward over-optimization in llm with demonstration-guided reinforcement learning. *arXiv preprint arXiv:2404.19409*, 2024.

[50] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

[51] H. Sikchi, A. Saran, W. Goo, and S. Niekum. A ranking game for imitation learning. *arXiv preprint arXiv:2202.03481*, 2022.

[52] H. Sikchi, Q. Zheng, A. Zhang, and S. Niekum. Dual rl: Unification and new methods for reinforcement and imitation learning. *arXiv preprint arXiv:2302.08560*, 2023.

[53] P. Singhal, T. Goyal, J. Xu, and G. Durrett. A long way to go: Investigating length correlations in rlhf, 2023.

[54] J. Skalse, N. H. R. Howe, D. Krasheninnikov, and D. Krueger. Defining and characterizing reward hacking, 2022.

[55] N. Stiennon, L. Ouyang, J. Wu, D. M. Ziegler, R. Lowe, C. Voss, A. Radford, D. Amodei, and P. Christiano. Learning to summarize from human feedback, 2022.

[56] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

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

[58] Y. Tang, D. Z. Guo, Z. Zheng, D. Calandriello, Y. Cao, E. Tarassov, R. Munos, B. Á. Pires, M. Valko, Y. Cheng, et al. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*, 2024.

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

[60] J. Taylor. Quantilizers: A safer alternative to maximizers for limited optimization. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[61] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[62] B. Wallace, M. Dang, R. Rafailov, L. Zhou, A. Lou, S. Purushwalkam, S. Ermon, C. Xiong, S. Joty, and N. Naik. Diffusion model alignment using direct preference optimization, 2023.

[63] J. Watson, S. Huang, and N. Heess. Coherent soft imitation learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL `https://openreview.net/forum?id=kCCD8d2aEu`.

[64] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL `https://doi.org/10.1007/BF00992696`.

[65] R. Yuanzhe Pang, W. Yuan, K. Cho, H. He, S. Sukhbaatar, and J. Weston. Iterative reasoning preference optimization. *arXiv e-prints*, pages arXiv–2404, 2024.

[66] Y. Zhai, H. Zhang, Y. Lei, Y. Yu, K. Xu, D. Feng, B. Ding, and H. Wang. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles, 2023.

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

[68] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *Conference on Neural Information Processing Systems Track on Datasets and Benchmarks.*, 2023.

[69] B. Zhu, M. I. Jordan, and J. Jiao. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*, 2024.

[70] B. D. Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

[71] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving. Fine-tuning language models from human preferences, 2020.

A Limitations and Societal Impacts

Our discussion highlights a number of issues with direct alignment algorithms used widely as means to align to human values. This work has mostly focused on pointing out those issues along with a theoretical underpinning of the issue, but does not provide a way to resolve these issues. We still assume an underlying model of human preferences, which is an ongoing research area as no model is perfect in explaining the ways humans give preferences. Our work aims to drive the push towards better alignment algorithms that do not overoptimize and generate models that safe to be deployed in our society. We believe only through understanding and demonstrating the shortcomings of current methods we can develop better alignment methods.

B Experiment Details

We largely follow the DPO setup unless otherwise mentioned and build on their code (https://github.com/eric-mitchell/direct-preference-optimization) without changing any hyperparameters unless otherwise mentioned.

For all DAA experiments, we used the curated OpenAI TL;DR dataset with 92K preferred-dispreferred summary completions [55]. Each prompt is a Reddit post belonging to one of several topic forums, with title/post metadata included. 256 prompts sampled from the held-out set are used for all evaluations (e.g. loss, accuracy, KL, winrates, length), with temperature 1.0 and max length 512.

Model sizes include 1B, 2.8B, and 6.9B and were initialized from the base Pythia pre-trained weights. All models underwent supervised fine-tuning on TL;DR prior to direct alignment. Across all SFT and DAA runs, we used a batch size of 128 (8 gradient accumulation steps), and RMSProp with a learning rate of 0.5×10^{-6} (linear warmup for 150 steps) for 1 epoch. 1B models were trained on 2 NVIDIA A40 GPUs, 2.8B models were trained on 4 NVIDIA A40 GPUs, and 6.9B models were trained on 4 NVIDIA A100 GPUs. All evaluations were computed with "gpt-4-turbo-2024-04-09" as judge, with random positional flips to avoid known bias.

C Appendix A: Complete Intra-Epoch Training Dynamics

This appendix contains similar intra-epoch KL divergence and winrate evolution results as in Fig. 2, across all model sizes.

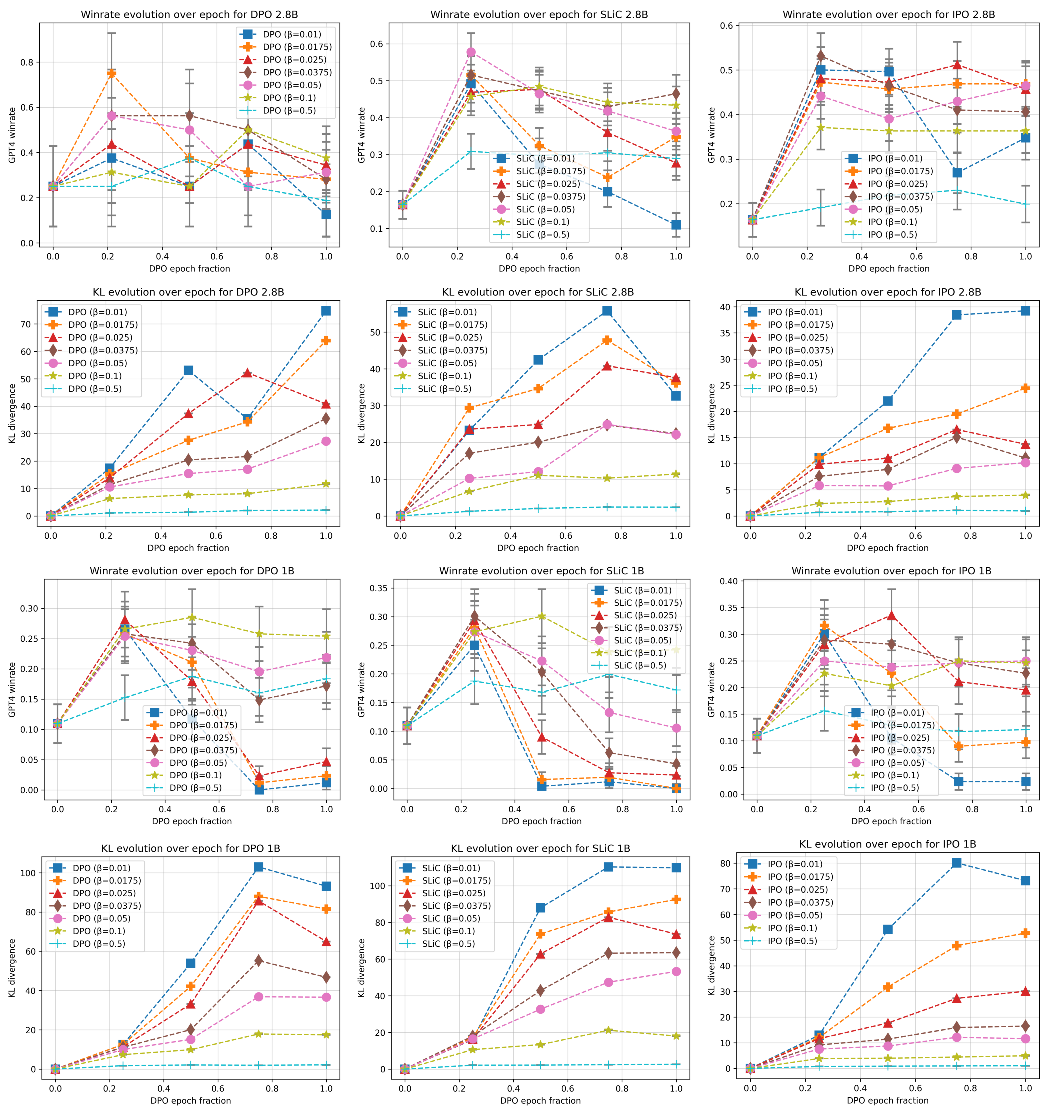


Figure 8: KL divergence and GPT4 winrate evolution for 2.8B and 1B models across DPO, SLiC, and IPO losses. Similar to the 6.9B models, performance tends to degrade after the first quarter epoch, particularly under a low KL budget, while KL increases almost monotonically.

D Overoptimization from the lens of Implicit Bootstrapping

Reward over-optimization is well understood in the classical RLHF setting, with a consensus that is driven by two main components - using a proxy reward function that is trained on limited data and continuous querying with new, potentially OOD samples during PPO training. At first glance none of these conditions hold in DAAs as we do not train a separate proxy reward model or generate new data during training. Therefore, understanding reward over-optimization in DAAs requires a new theory. We will base our analysis on [47] using the token-level MDP and corresponding (soft) Q-learning formulation. Consider the class of dense per-token reward functions $r_\theta(x, y_{\leq i})$, where $y_{\leq i}$ denotes the first i tokens of y, with sequence level-reward $r_\theta(x, y) = \sum_{i=1}^{|y|} r_\theta(x, y_{\leq i})$. This is a strictly more general class than the sparse reward function which returns a single score at the end of the sequence, since we can set all intermediate rewards as 0. Within the framework of [47] given a DAA-trained policy π_θ, there exists a dense per-token reward r_θ, that minimizes the reward modelling objective in Eq. 2 and satisfy the below.

The (soft) Bellman Equation holds:

$$Q^*(y_i, (x, y_{<i})) = \begin{cases} r(x, y_{\leq i}) + \beta \log \pi_{\text{ref}}(y_i | (x, y_{<i})) + V^*((x, y_{\leq i})), & \text{if } y_i \text{ is not } \textbf{EOS} \\ r(x, y_{\leq i}) + \beta \log \pi_{\text{ref}}(y_i | (x, y_{<i})), & \text{if } y_i \text{ is } \textbf{EOS} \end{cases} \tag{8}$$

where V^* is the corresponding soft-value function:

$$V^*((x, y_{<i})) = \beta \log \sum_{y \in |V|} e^{Q^*(y, (x, y_{<i}))/\beta} \tag{9}$$

then the DAA policy π_θ satisfies:

$$\pi_\theta(y_i | (x, y_{<i})) = \exp(\frac{1}{\beta} Q^*(y_i, (x, y_{<i})) - V^*((x, y_{<i}))) \tag{10}$$

in this interpretation, the LLM logits $l_\theta[i] = Q^*(y_i, (x, y_{<i}))/\beta$ represent Q-values. With a direct substitution we then have

$$Q^*(y_i, (x, y_{<i})) = r(x, y_{\leq i}) + \beta \log \pi_{\text{ref}}(y_i | (x, y_{<i})) + \underbrace{\beta \log \sum_{y_i \in |V|} e^{Q^*(y, (x, y_{<i}))/\beta}}_{\text{OOD bootstrapping}} \tag{11}$$

That is in this framework DAAs may suffer from the classical OOD bootstrapping issue in offline RL [20, 35, 33, 52]. In this case even though the objective is trained fully offline we still effectively query the model on the values of unseen tokens. This interpretation also provides further insight into the effect of the β coefficient and the training dynamics. For small values of beta the estimate

$$\beta \log \sum_{y_i \in |V|} e^{Q^*(y, (x, y_{<i}))/\beta} \approx \max_{y \in |V|} Q^*(y, (x, y_{<i})) \tag{12}$$

that is smaller parameter values yield a more optimistic estimate, which results in higher level of OOD bootstrapping. This interpretation would also explain the somewhat counter-intuitive results of section 3.4. While the implicit reward function can adequately fit and model the data, the resulting LLM might behave sub-optimally, due to OOD bootstrapping in the corresponding Q-value estimate.

E Understanding Behavior of DAAs on OOD sequences

We have established that common DAA objectives allow for placing a high-likelihood on OOD data. In practice, while one might expect the likelihood of preferred responses to increase during training, it has been observed that algorithms like DPO decrease the likelihood of both the preferred and dis-preferred responses [42]. In fact, this is expected from a max-entropy RL perspective [47]. Since the total probability mass must sum to one, the probability of OOD responses must increase during the course of training. A small amount of extrapolation may be necessary to reach the optimal policy, however, too much is potentially detrimental to performance. Because they are not adequately constrained to the reference distribution, current DAA objectives allow this to happen.

To understand how DAAs allocate probability mass out of distribution, we use a toy Markov Decision Process (MDP), that mimics the LLM setting. The MDP is modeled as a tree, originating from a single start state, featuring deterministic transitions. The Toy MDP is illustrated in fig. 6.

E.1 Designing a toy LLM MDP

The MDP is modeled as a tree, originating from a single start state. This configuration mirrors the token-level MDP in Direct Preference Optimization (DPO) [47], or the scenario where both preferred and dispreferred responses are conditioned on the same prompt in the broader Large Language Model alignment context. Each leaf node in the MDP transitions deterministically to a terminal absorbing state, regardless of the action taken. The deterministic transitions resemble the LLM setting, where the current state is represented by the sequence of encountered tokens $(s_1, s_2, ..., s_i)$, and the action corresponds to predicting the next word s_{i+1} from the vocabulary, given the context. In this simplified MDP, the deterministic transition is akin to a concatenation function, advancing the state to the next step $(s_1, s_2, ..., s_i, s_{i+1})$. Employing a toy MDP enables us to systematically evaluate the trajectory probabilities for all feasible paths within the MDP, shedding light on the allocation of probability mass by Direct Alignment Algorithms (DAAs) towards out-of-distribution (OOD) trajectories.

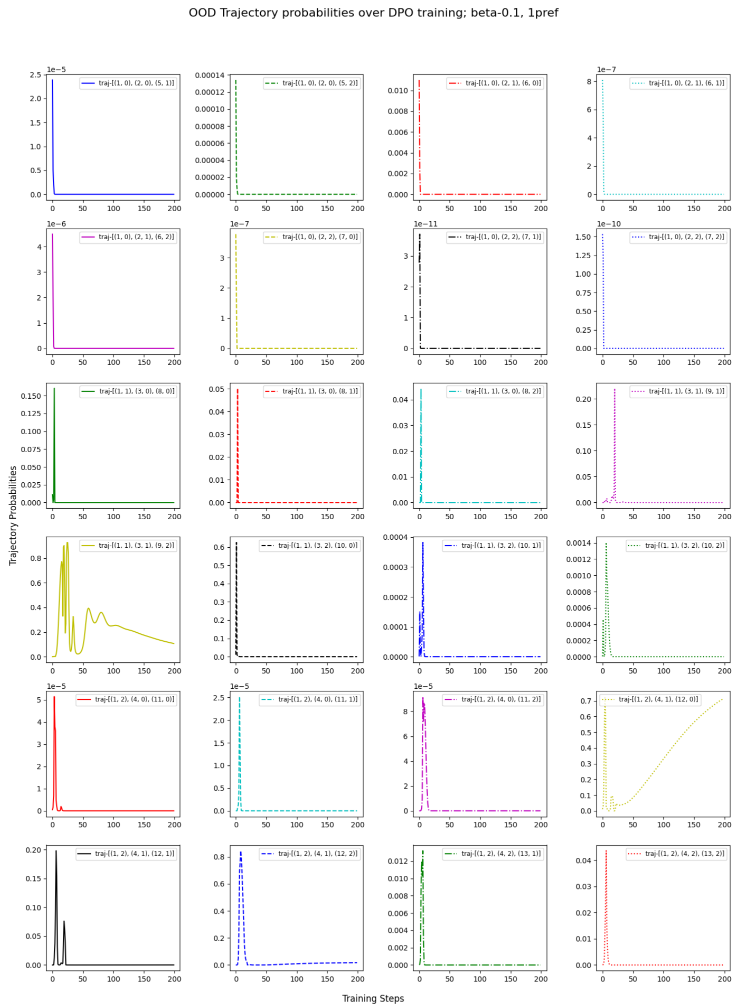
The Experimental Setup. We adhere to the standard direct alignment protocol [46][41], encompassing two key stages:

1. **Supervised Fine-tuning (SFT) / Behavioral Cloning (BC):** This phase involves fine-tuning the policy based on a limited number of trajectories. Specifically, we utilize three demonstrations for SFT: $(s_1, a_0, s_2, a_0, s_5, a_0, s_\infty)$, $(s_1, a_1, s_3, a_1, s_9, a_0, s_\infty)$, and $(s_1, a_2, s_4, a_2, s_{13}, a_2, s_\infty)$.
2. **Alignment with Preferences:** In this stage, preferences extracted from trajectories are employed to align the policy. Notably, we have only one preference available: $(s_1, a_1, s_3, a_1, s_9, a_0, s_\infty) \succ (s_1, a_0, s_2, a_0, s_5, a_0, s_\infty)$. This deliberate constraint exaggerates a scenario with limited data, enabling us to gauge the probability mass allocated to out-of-distribution (OOD) trajectories under such conditions. Insights garnered from this exaggerated low-data scenario hold relevance for Large Language Model (LLM) settings where preference datasets are notably smaller compared to the scale of LLM models deployed.

We utilize a Recurrent Neural Network (RNN) policy to navigate through the MDP, facilitating a closer resemblance to real-world language modeling scenarios.

Subsequently, we explore three distinct direct alignment loss functions: Direct Preference Optimization (DPO) [46], Identity Preference Optimization (IPO) [4], and Sequence Likelihood Calibration (SLiC) [67]. Additionally, we investigate how the selection of the KL penalty coefficient β influences the distribution of probability mass on OOD trajectories. This exploration encompasses three values of β: $(0.01, 0.1, 0.5)$.

In general, the plots illustrate that Direct Alignment Algorithms (DAAs) tend to allocate a significant proportion of the probability mass to out-of-distribution (OOD) trajectories during the alignment process. While Figure **??** may suggest that Direct Preference Optimization (DPO) can retain a substantial amount of probability mass on the selected trajectory in the preference dataset, it's noteworthy that the plots for DPO exhibit considerable noise. To provide further insight, Figure 18 displays the plots resulting from three additional repetitions of the DPO experiment. This elucidates the unconstrained nature of the DPO problem: multiple solutions exist for the DPO loss, each distributing varying amounts of probability mass to OOD trajectories. In the experiments with IPO and SLiC, it's observed that the probability mass allocated to in-distribution trajectories diminishes substantially over the course of training. Notably, the probability mass becomes concentrated on a select few out-of-distribution trajectories. Moreover, consistent trends are discernible across various values of β. All our experiments with Toy-MDP can be found in the following figures 12, 9, 15, 13, 10, 16, 14, 11, 17.



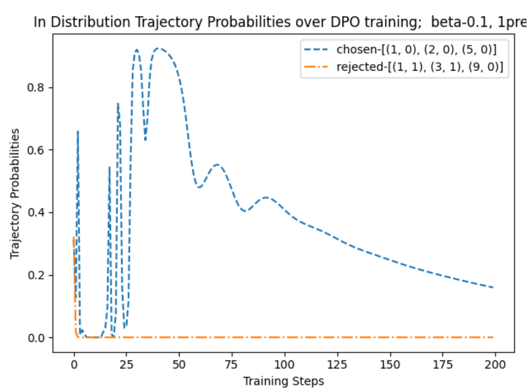
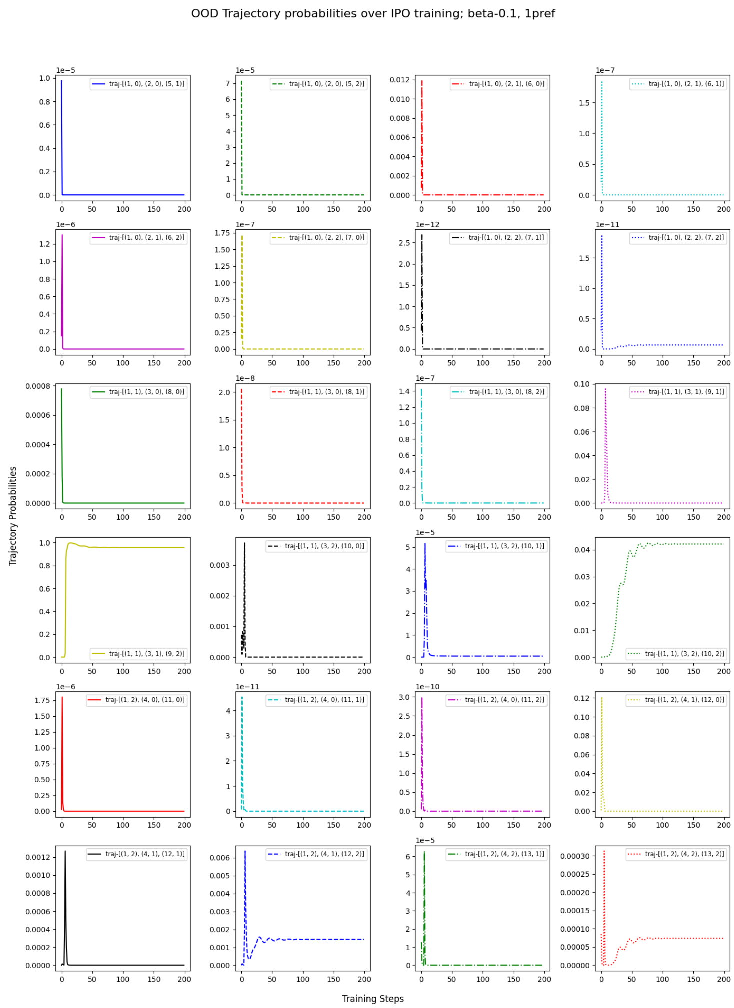


Figure 9: Trajectory probabilities throughout DPO training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training

.



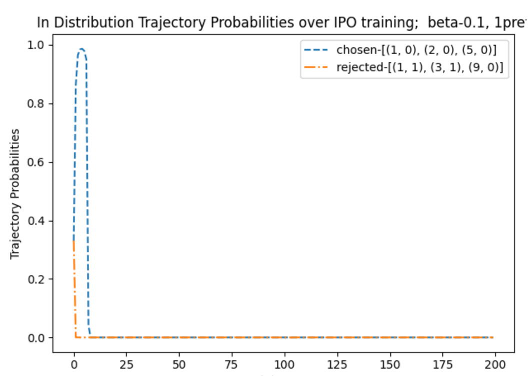
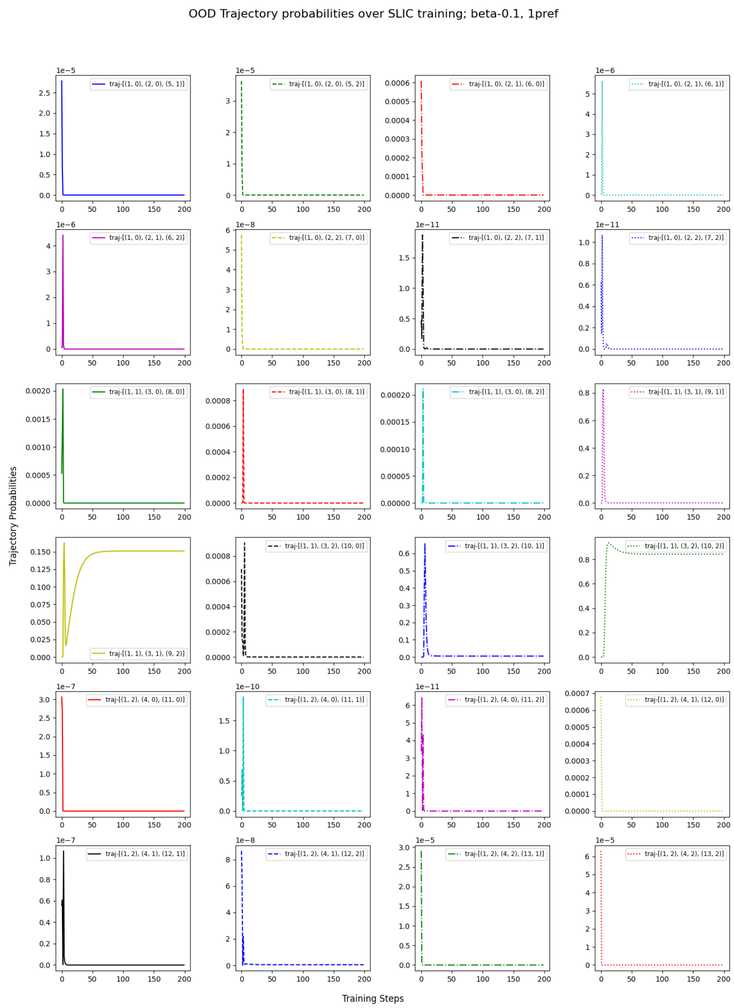


Figure 10: Trajectory probabilities throughout IPO training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training



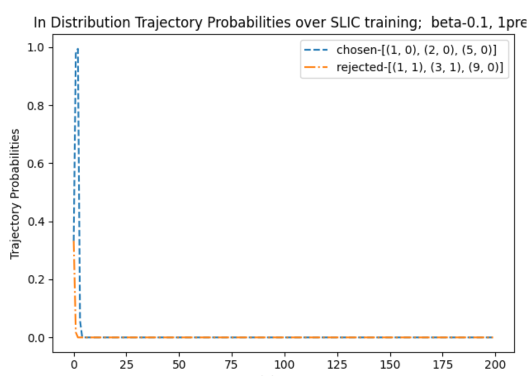


Figure 11: Trajectory probabilities throughout SLiC training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training

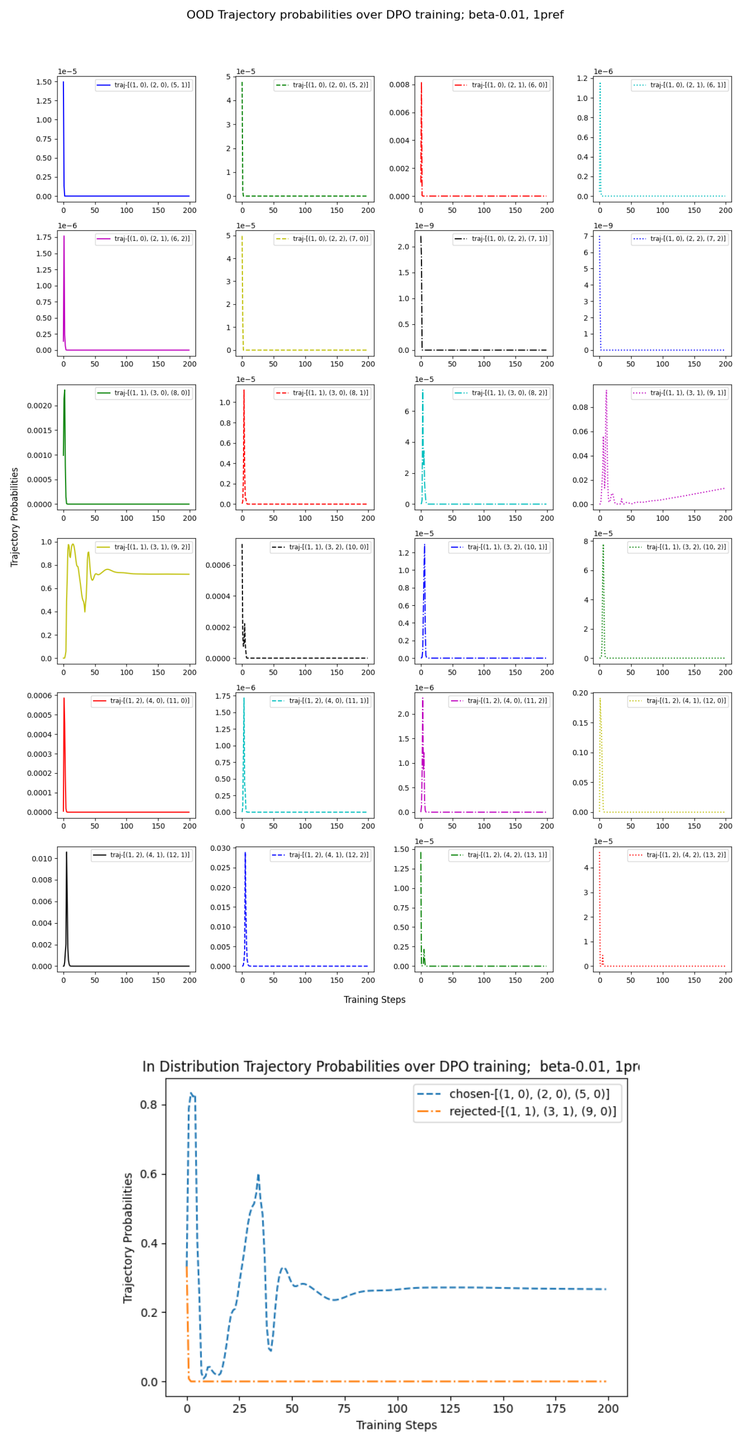


Figure 12: Trajectory probabilities throughout DPO training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training

.

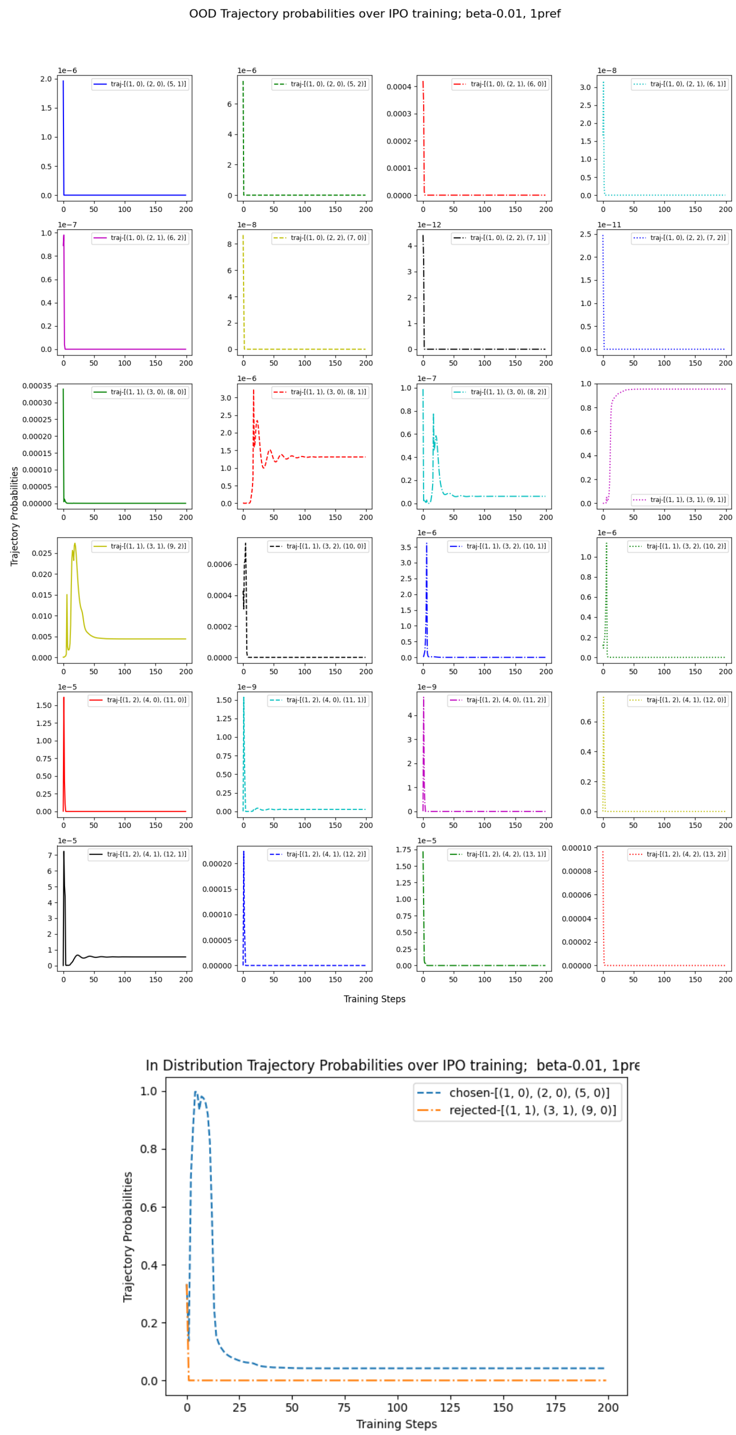
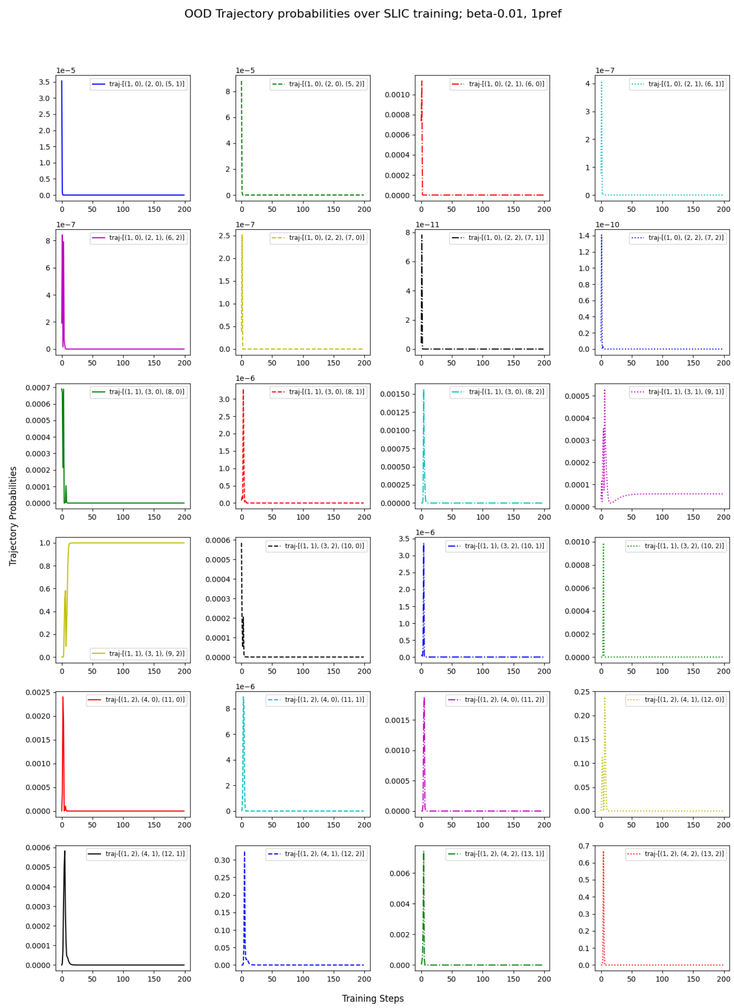


Figure 13: Trajectory probabilities throughout IPO training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training



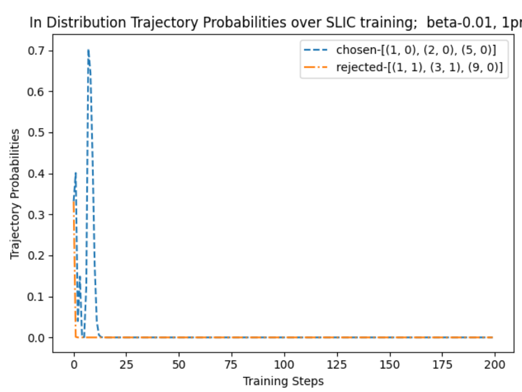
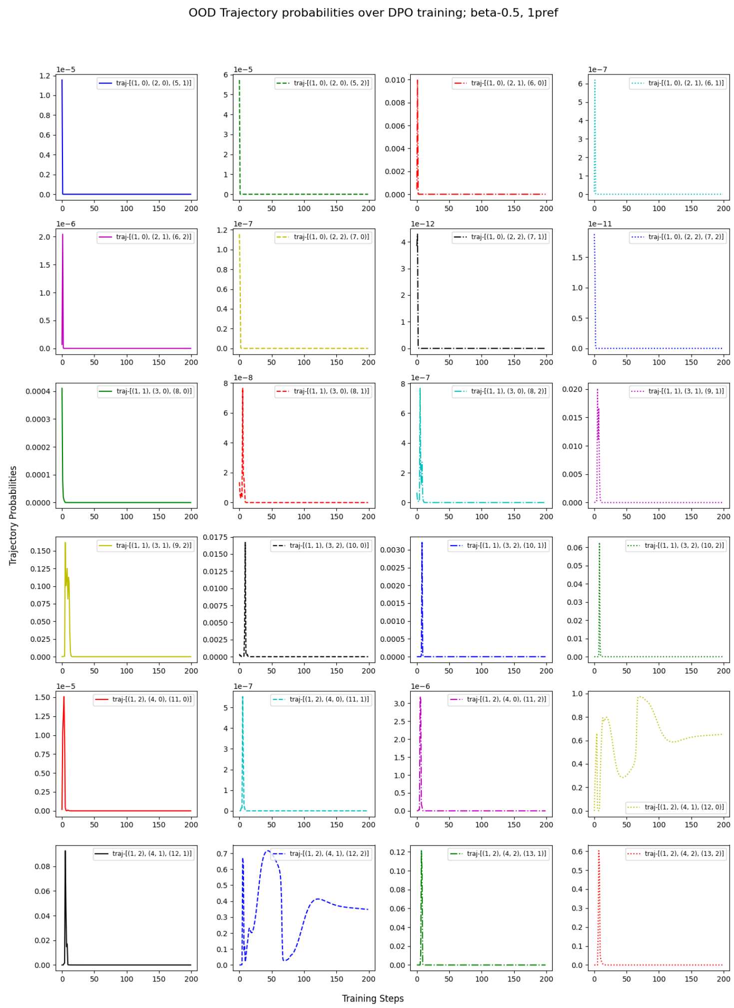


Figure 14: Trajectory probabilities throughout SLiC training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training



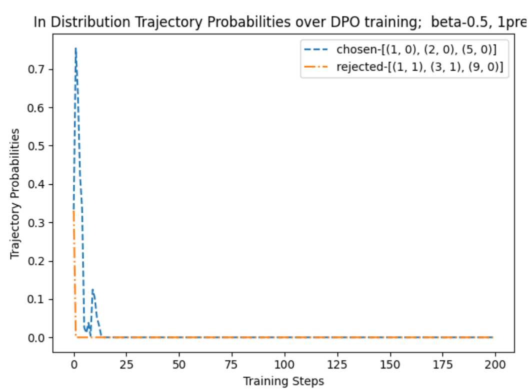
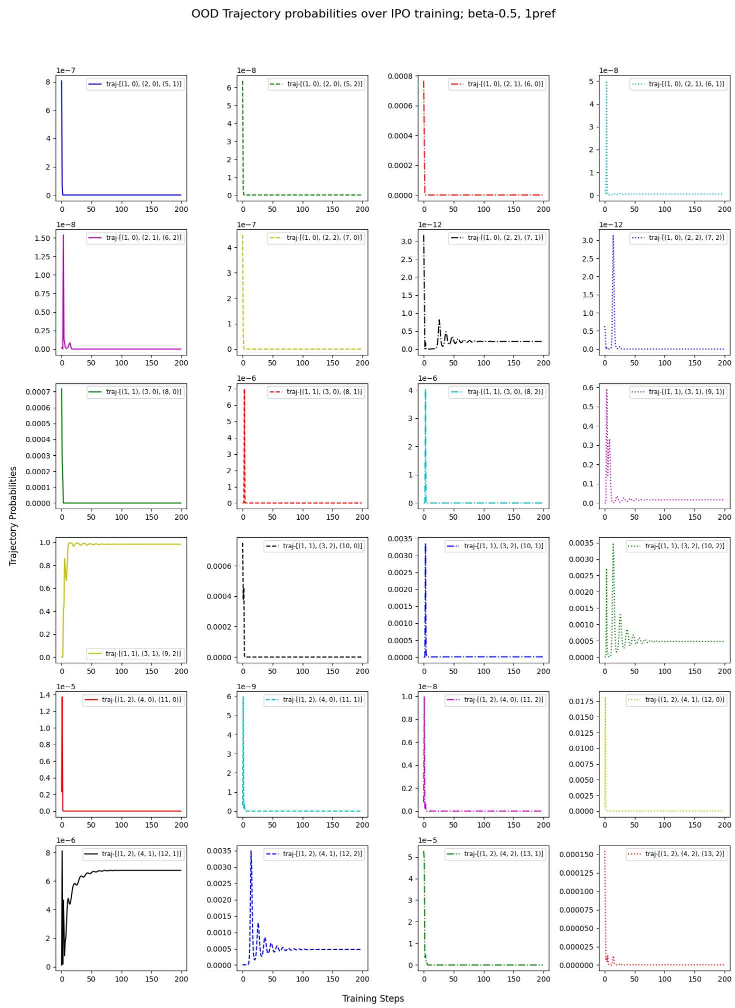


Figure 15: Trajectory probabilities throughout DPO training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training

.



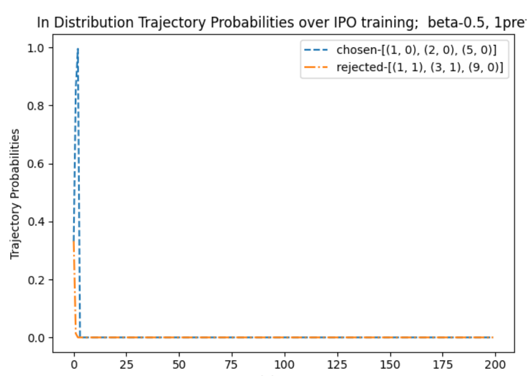
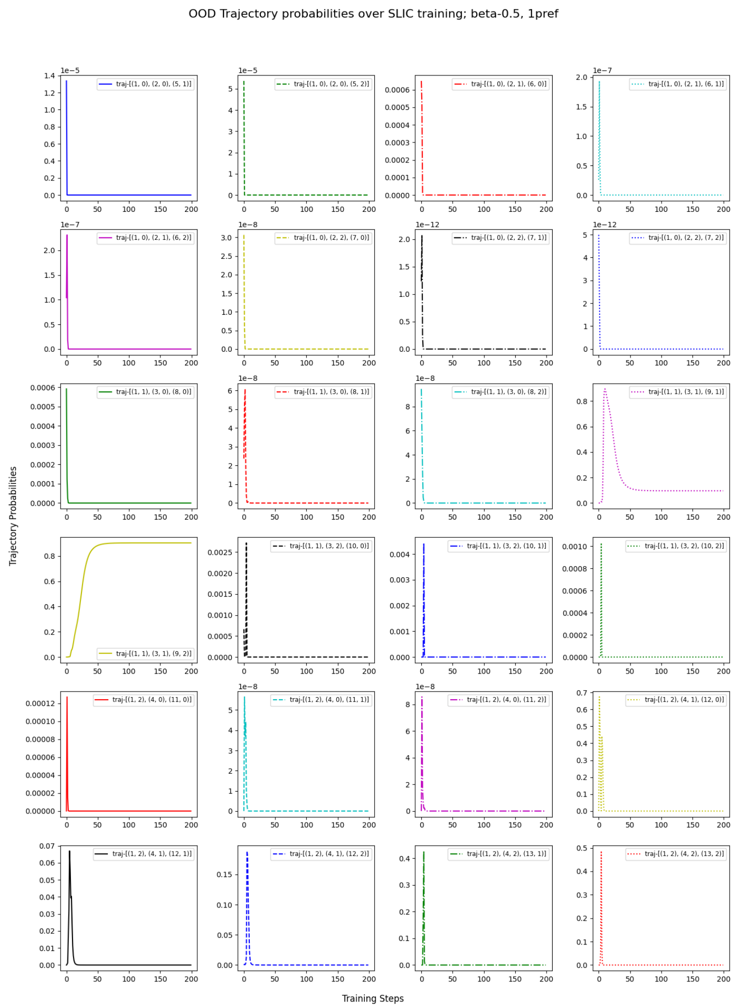


Figure 16: Trajectory probabilities throughout IPO training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training



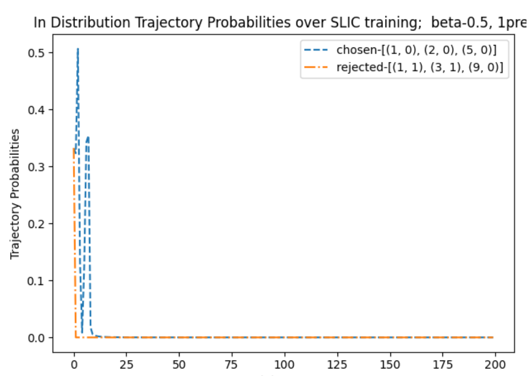


Figure 17: Trajectory probabilities throughout SLiC training. The top plot shows how the probability mass of different OOD trajectories, changes throughout training. The bottom plot shows how the probability mass of the trajectories in our preference dataset (size 1) changes over training

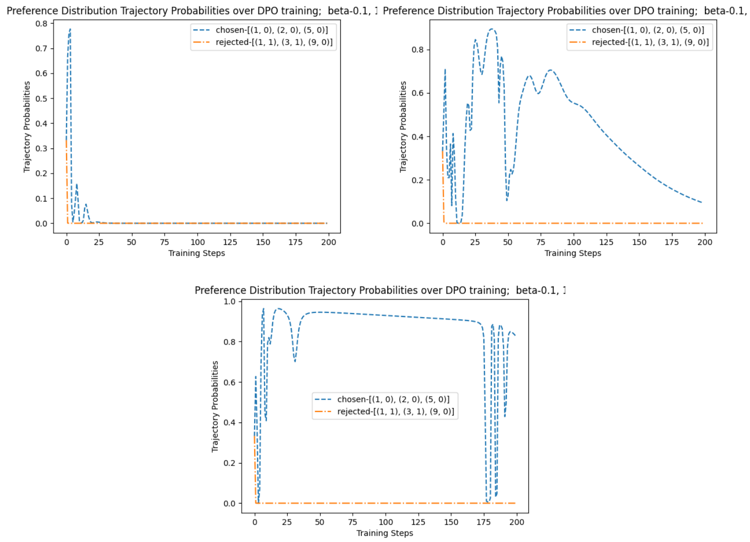


Figure 18: Trajectory probabilities throughout DPO training, over multiple runs
.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The paper faithfully adheres to the claims and motivation in the abstract and provides proof and detailed empirical studies in support.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A discussion of our limitations can be found as a separate section at the beginning of the appendix.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide proofs and empirical evidence to support all our theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide detailed guidance on reproducibility by specifying all datasets, code, and hyperparameters used in this work.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have only used open-source models with open-source datasets for all aspects of the work. Please refer to section B for details on reproducing the results.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We list detailed information about the training and test details in Section **??**. Our experiments use open-source datasets and models.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Training Large Language models is time-consuming and compute-intensive. Our experiments do not run multiple seeds on one configuration due to limited computing and financial budget. Instead, the focus of this work is extensive evaluation across multiple configurations which we spent all our compute resources into. Our evaluation protocol is similar to prior influential works in RLHF [46, 21].

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide information on compute resources in the experimental details section B in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: We abide by the code of ethics in every respect.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss societal impacts in Section A of the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We use public models that are fine-tuned for alignment on open-source datasets. Our models do not contribute any additional risk over the base models as we are explicitly training for alignment.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The pretrained models in this work come from the Pythia family all of which are classified under Apache License https://huggingface.co/EleutherAI/pythia-2.8b/tree/main. The TL;DR comparison dataset used in this work uses a modified MIT License https://github.com/openai/summarize-from-feedback/blob/master/LICENSE.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We use open-source pretrained models and provide details to reproduce our fine-tuning experiments.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: We do not use crowdsourcing or research with human subjects in this work

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not use crowdsourcing or research with human subjects in this work.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.