Degeneration-free Policy Optimization: RL Fine-Tuning for Language Models without Degeneration

Youngsoo Jang [1] Geon-Hyeong Kim [1] Byoungjip Kim [1] Yu Jin Kim [1] Honglak Lee [1] Moontae Lee [1 2]

Abstract

As the pre-training objectives (e.g., next token prediction) of language models (LMs) are inherently not aligned with task scores, optimizing LMs to achieve higher downstream task scores is essential. One of the promising approaches is to fine-tune LMs through reinforcement learning (RL). However, conventional RL methods based on PPO and a penalty of KL divergence are vulnerable to text degeneration where LMs do not generate natural texts anymore after RL fine-tuning. To address this problem, we provide **Degeneration-free Policy Optimization (DfPO)** that can fine-tune LMs to generate texts that achieve improved downstream task scores, while preserving the ability to generate natural texts. To achieve this, we introduce *KL-masking* which masks out the actions that potentially cause deviation from the reference policy when its likelihood is increased or decreased. Then, we devise *truncated advantage functions* for separately performing likelihood maximization and minimization to improve the task performance. In the experiments, we provide the results of DfPO and baseline algorithms on various generative NLP tasks including text continuation, text detoxification, and commonsense generation. Our experiments demonstrate that DfPO successfully improves the downstream task scores while preserving the ability to generate natural texts, without requiring additional hyperparameter search.

1. Introduction

Although pre-trained language models (LMs) have achieved remarkable success in various NLP tasks (Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022; Stiennon et al., 2020; Nakano et al., 2022), fine-tuning LMs on downstream tasks is essential to achieve higher task scores. Since the pre-training and supervised fine-tuning objectives (e.g., next token prediction (Radford et al., 2019)) of LMs are inherently not maximizing the task scores, LMs fail to learn an optimal behavior. One of the promising approaches to fine-tune the LMs is reinforcement learning (RL) (Christiano et al., 2017). Recently, reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Liang et al., 2022; Kim et al., 2023; Ouyang et al., 2022) methods, which learn a reward model from human feedback and then fine-tune LMs through reinforcement learning, has successfully achieved to fine-tune the LMs using RL (Ouyang et al., 2022; Glaese et al., 2022; Bai et al., 2022; Stiennon et al., 2020; Nakano et al., 2022). However, optimizing LMs against a given reward model through RL is yet challenging due to the *degeneration problem* which generates unnatural responses diverging from human language.

When optimizing LMs through RL, most of the existing online RL methods mainly use PPO (Schulman et al., 2017) for optimizing an LM policy, and a penalty of KL divergence between the reference LM and an optimized LM for preserving the ability to generate natural texts (Ouyang et al., 2022; Ramamurthy et al., 2023). However, conventional RL methods based on PPO and a KL divergence penalty are vulnerable to the text degeneration problem that LMs do not generate natural texts anymore after RL fine-tuning. As illustrated in Figure 1, we can observe the text degeneration problem through the diverging perplexity as the sentiment score increases. We carefully conjecture that a penalty of KL divergence often does not work with respect to the penalty ratio β, since PPO is a simplified algorithm from TRPO (Schulman et al., 2015) that guarantees to maximize target task rewards. In other words, it is important to balance two different objectives: (1) maximizing task rewards and (2) minimizing the KL divergence (i.e. preserving the ability to generate natural texts), while preventing one objective from dominating the overall quantity. Especially in language generation tasks, confining the probability distribution of LMs within a certain level is more important than simply maximizing the task rewards.

To address this problem, we present **Degeneration-free**

[1]LG AI Research [2]University of Illinois Chicago. Correspondence to: Moontae Lee <moontae.lee@lgresearch.ai>.

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

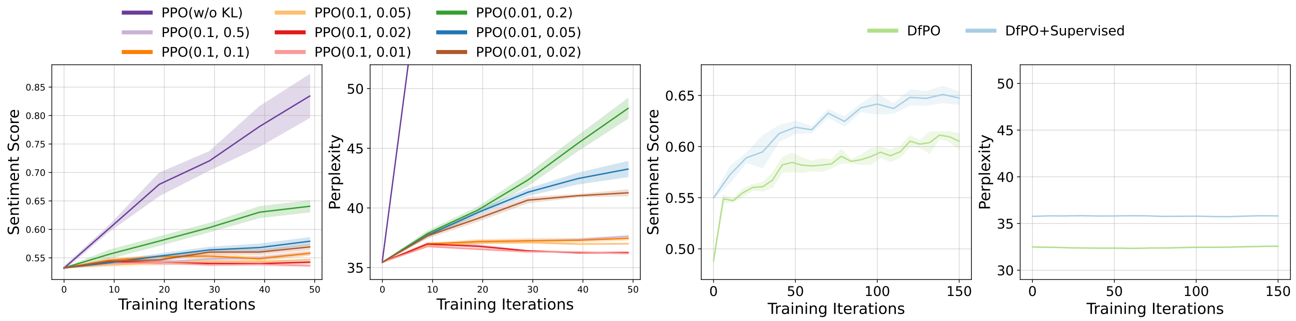


Figure 1. Averaged learning curves over 5 runs of **(Left)** PPO for the varying the coefficient of KL-regularization penalty and **(Right)** DfPO on IMDB text continuation task. PPO (β, $\text{KL}_{\text{target}}$) indicates the PPO that considers the KL-regularization as a reward penalty with KL coefficient β and target KL. The goal of the IMDB text continuation task is to learn a policy that maximizes the sentiment score (i.e. task rewards) while preserving the perplexity score (i.e. naturalness). As shown in the results, PPO with KL-regularization is highly sensitive to hyperparameters, impacting both sentiment and perplexity scores. In contrast, our proposed algorithm, DfPO, enhances sentiment scores while maintaining initial perplexity scores, without any additional hyperparameter search. DfPO and DfPO+Supervised indicate the DfPO training starts from a pre-trained language model and a supervised fine-tuning model as initial policies, respectively.

Policy Optimization (DfPO) that can fine-tune LMs to generate texts that effectively achieve higher downstream task scores, while preserving the ability to generate natural texts. To achieve this, we first investigate why the degeneration problem occurs in policy gradient algorithms under the perspective of likelihood max/minimization. Then we reformulate policy optimization as stable likelihood max/minimization with *KL-masking* and *truncated advantage functions* that eventually mitigates excessive task reward optimization. Consequently, DfPO does not require any sensitive hyperparameters such as the penalty ratio of KL divergence used in conventional RL methods. In the experiments, we provide the results of DfPO and baseline algorithms on various generative NLP tasks including text continuation (Maas et al., 2011), commonsense generation (Lin et al., 2020), and text detoxification (Gehman et al., 2020) using diverse LMs. Our experiments demonstrate that DfPO successfully improves the downstream task scores while preserving the ability to generate natural texts, without requiring additional hyperparameter search.

The contributions of this paper can be summarized as follows:

- We find that conventional RL fine-tuning methods that use PPO (Schulman et al., 2017) and a penalty of KL divergence are vulnerable to result in the text degeneration problem that LMs do not generate natural texts anymore after RL fine-tuning (see Figure 1).
- We introduce Degeneration-free Policy Optimization (DfPO) that can fine-tune LMs to generate texts that achieve higher downstream task scores while preserving the ability to generate natural texts. To achieve this, we reformulate policy optimization as stable likelihood max/minimization with *KL-masking* and *truncated advantage functions* (see Eq. 5, 6, and 7).
- We mathematically analyse the obejctive of DfPO and provide the connection to the objective of PPO with KL-regularization penalty (see Appendix A).
- We demonstrate that, **even DfPO does not perform hyperparameter search**, it achieves similar performance to PPO (Schulman et al., 2017) and NLPO (Ramamurthy et al., 2023) which require additional hyperparameter search for the penalty ratio of KL divergence on various generative NLP tasks including text continuation, text detoxification, and commonsense generation (see Figure 1, 3, 4, 5 and Table 2, 4, 5, 6, 7).
- We also show that DfPO can fine-tune the **large language model (GPT-J-6B)** to improve the task performance while preserving the ability to generate natural texts (see Figure 4 and 9).

2. Preliminaries

2.1. Reinforcement Learning for Language Models

We consider the generative NLP tasks that can be modeled as a Markov decision process (MDP) defined by tuple $M = \langle S, A, T, r, p_0, \gamma \rangle$ (Sutton & Barto, 1998), where S is the set of states s (a sequence of word tokens), A is the set of actions a (a next word token), $T : S \times A \to \Delta(S)$ is the transition probability, $r : S \times A \to \mathbb{R}$ is the reward function, $p_0 \in \Delta(S)$ is the distribution of the initial state, and $\gamma \in [0, 1)$ is the discount factor. The policy $\pi(a|s)$ is a mapping from state to a probability distribution over A, which can be naturally modeled as language models. The value function, action-value function, and advantage function are defined as $V^\pi(s) := \mathbb{E}_{s_0,a_0,\cdots\sim\pi}[\sum_t^\infty \gamma^t r(s_t, a_t)|s_0 = s]$, $Q^\pi(s, a) := r(s, a) + \gamma\mathbb{E}_{s'\sim T(s,a)}[V^\pi(s')]$, and $A^\pi(s, a) := Q^\pi(s, a) - V^\pi(s)$, respectively. Unlike standard RL problems that aim to maximize only cumulative rewards, the goal of generative NLP tasks is to find an optimal policy that maximizes cumulative rewards while preserving the ability to generate natural texts.

2.2. Policy Gradient with KL-regularization penalty

Policy gradient algorithms are widely employed in reinforcement learning to maximize the expected cumulative rewards $\mathbb{E}_{s_0,a_0,\cdots\sim\pi}[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)]$. In this paper, we use the following formulation based on the advantage function, which represents the most commonly used form of the policy gradient estimator:

$$\nabla_\theta J(\theta) = \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\left[A^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a|s)\right], \qquad (1)$$

where $d^{\pi}(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s)$ is a stationary distribution with $s_0 \sim p_0$ and the actions are chosen according to π, and $d^{\pi}(s,a) := d^{\pi}(s)\pi(a|s)$. However, conventional RL algorithms that focus solely on maximizing the task rewards easily result in *reward hacking* behaviors, which correspond to the *degeneration problems* in generative NLP tasks. To optimize the language model while preserving the ability to generate natural texts (i.e. preventing the degeneration problems), the current best-performing RL algorithm for generative NLP tasks utilizes a KL-divergence penalty between the current policy π_θ and the reference policy π_0 for preserving the naturalness by not much deviating from the reference policy as follows:

$$\begin{aligned} &\underset{\theta}{\text{maximize}} && \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\left[A^{\pi_\theta}(s,a)\right] && (2)\\ &\text{subject to} && \mathbb{E}_{s\sim d^{\pi_\theta}}\left[\mathrm{KL}(\pi_\theta(\cdot|s)\|\pi_0(\cdot|s))\right] \le \delta, && (3)\end{aligned}$$

where $\delta > 0$ is a hyperparameter. Based on the Lagrangian, we obtain the following unconstrained optimization for the constrained optimization problem in (2-3):

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\ \mathbb{E}_{d^{\pi_\theta}}\left[A^{\pi_\theta}(s,a) - \beta\mathrm{KL}(\pi_\theta(\cdot|s)\|\pi_0(\cdot|s))\right]\\ &\qquad = \underset{\theta}{\text{maximize}}\ \mathbb{E}_{d^{\pi_\theta}}\left[A^{\pi_\theta}(s,a) + \beta R^{\pi_\theta}(s,a)\right], \qquad (4)\end{aligned}$$

where $\beta \ge 0$ is a fixed hyperparameter rather than a Lagrangian multiplier as in previous studies, and $R^{\pi_\theta}(s,a) := \log\frac{\pi_0(a|s)}{\pi_\theta(a|s)}$. However, as discussed in prior works (Ramamurthy et al., 2023; Ouyang et al., 2022; Ziegler et al., 2019), optimizing the policy through the KL-penalized objective (Eq. 4) is very sensitive to the hyperparameter β. Figure 1 shows the results of PPO with various β for sentiment score (i.e. task scores) and perplexity (i.e. naturalness) on the IMDB text continuation task. This sensitivity to hyperparameters can be especially troublesome for fine-tuning the large-scale language models that require *massive computational costs* and also cause *ambiguity in model selection*, as shown in Figure 1.

	$A^{\pi_\theta}(s,a) > 0$	$A^{\pi_\theta}(s,a) < 0$
$R^{\pi_\theta}(s,a) > 0$	Desirable Samples (Likelihood Maximization)	Degeneration or Task Scores ↓
$R^{\pi_\theta}(s,a) < 0$	Degeneration or Task Scores ↓	Desirable Samples (Likelihood Minimization)

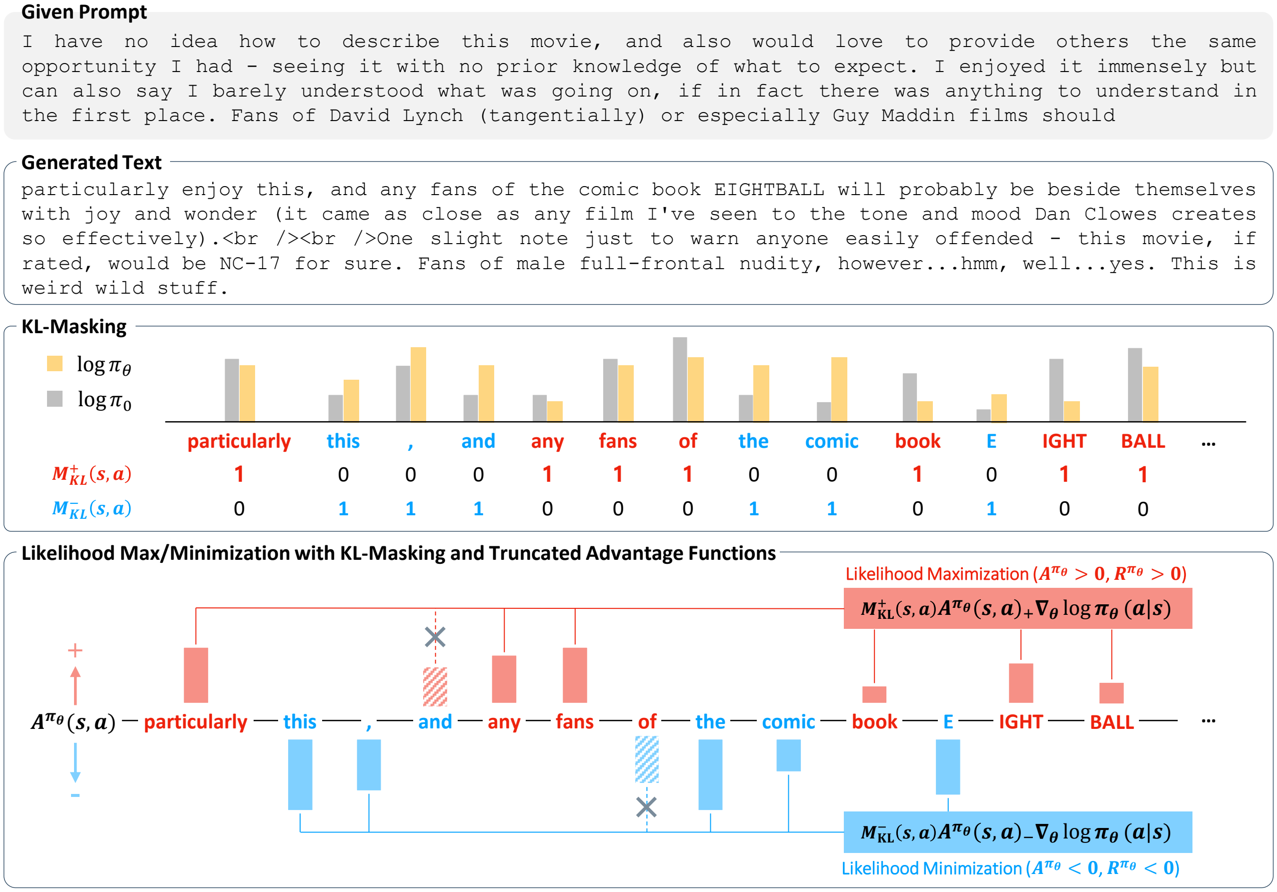
Table 1. Summary for all types of samples used in policy gradient update. The green area represents the state-action pairs that both advantage A^{π_θ} and log ratio R^{π_θ} are positive or negative, which are desirable samples for improving task scores while not deviating from reference policy. On the other hand, the red area represents the state-action pairs that have opposite directions to improve task scores and avoid deviating from the reference policy, which are undesirable samples that can cause degeneration problems in the policy update.

3. Degeneration-free Policy Optimization

In this section, we present **Degeneration-free Policy Optimization (DfPO)** that can fine-tune LMs to generate texts that achieve improved downstream task scores, while preserving the ability to generate natural texts, *without any additional costs for hyperparameter search*. First, we investigate why the degeneration problem occurs in policy gradient algorithms under the perspective of likelihood max/minimization. Then we reformulate policy optimization as stable likelihood max/minimization that eventually mitigates excessive task reward optimization with 1) *KL-masking* which masks out the samples that cause deviating from the reference policy when likelihood max/minimization, and 2) *truncated advantage functions* which separately perform likelihood max/minimization with KL-masking to improve the downstream task scores.

3.1. Understanding Policy Gradient via Likelihood Max/Minimization

Before presenting our algorithm, we first investigate why the degeneration problem occurs in policy gradient algorithms under the perspective of likelihood max/minimization. Intuitively, the policy gradient update can be interpreted as the likelihood max/minimization: gradient update through Eq. (1) that increases the likelihood of actions that are better than the average behavior of the current policy (i.e. positive advantages) and decreases the likelihood of actions that are worse than the average behavior of the current policy (i.e. negative advantages). However, when considering regularized optimization as in Eq. (4), it is challenging to determine whether the actions are better or worse than the average behavior of the current policy in terms of both task scores and naturalness (i.e. penalty of KL-regularization).



Given Prompt

```
I have no idea how to describe this movie, and also would love to provide others the same
opportunity I had - seeing it with no prior knowledge of what to expect. I enjoyed it immensely but
can also say I barely understood what was going on, if in fact there was anything to understand in
the first place. Fans of David Lynch (tangentially) or especially Guy Maddin films should
```

Generated Text

```
particularly enjoy this, and any fans of the comic book EIGHTBALL will probably be beside themselves
with joy and wonder (it came as close as any film I've seen to the tone and mood Dan Clowes creates
so effectively).<br /><br />One slight note just to warn anyone easily offended - this movie, if
rated, would be NC-17 for sure. Fans of male full-frontal nudity, however...hmm, well...yes. This is
weird wild stuff.
```

KL-Masking

	particularly	this	,	and	any	fans	of	the	comic	book	E	IGHT	BALL	...
$M^+_{KL}(s,a)$	1	0	0	0	1	1	1	0	0	1	0	1	1	
$M^-_{KL}(s,a)$	0	1	1	1	0	0	0	1	1	0	1	0	0	

$\log \pi_\theta$
$\log \pi_0$
Likelihood Max/Minimization with KL-Masking and Truncated Advantage Functions
Likelihood Maximization ($A^{\pi_\theta} > 0, R^{\pi_\theta} > 0$)
$M^+_{\mathrm{KL}}(s,a)A^{\pi_\theta}(s,a)_+\nabla_\theta \log \pi_\theta(a|s)$
+
$A^{\pi_\theta}(s,a)$ — particularly — this — , — and — any — fans — of — the — comic — book — E — IGHT — BALL — ...
-
$M^-_{\mathrm{KL}}(s,a)A^{\pi_\theta}(s,a)_-\nabla_\theta \log \pi_\theta(a|s)$
Likelihood Minimization ($A^{\pi_\theta} < 0, R^{\pi_\theta} < 0$)

Figure 2. Illustrative example of the main mechanism of DfPO on IMDB text continuation task. The figure shows the process of sampling the action sequence (i.e. generated text) and updating the policy for the initial state s_0 (i.e. given prompt). Our policy optimization consists of two processes: (1) KL-Masking and (2) truncated advantage functions. Red/blue colored tokens indicate unmasked tokens due to positive/negative KL-masking, and X mark above the dashed box indicates undesirable samples that are truncated and not used in the policy update due to the different signs of advantage A^{π_θ} and log ratio R^{π_θ} (i.e. samples of red area in Table 1). The likelihood maximization (red box) and the likelihood minimization (blue box) correspond to the first and second terms in Eq. (7), respectively.

Table 1 summarizes cases of all state-action samples used in the policy gradient update, according to the signs of the advantage A^{π_θ} and the log ratio R^{π_θ}. First, state-action pairs corresponding to the green area, where both A^{π_θ} and R^{π_θ} are positive or negative, are desirable samples that can update the policy to improve task performance while preserving the naturalness (i.e. not deviating from reference policy) through the policy gradient. On the other hand, state-action pairs corresponding to the red area, which have opposite directions for improving task performance and preserving naturalness, are undesirable samples that can cause degeneration problems or deteriorate task performance in the policy gradient update. Therefore, if we perform a policy gradient update using only desirable samples in the green area, we can learn a policy that maximizes task rewards while preserving the naturalness of the generated texts, without any dependency on hyperparameters.

3.2. Defining KL-Masking with Reference Policy

In order to perform a policy gradient update using only desirable samples corresponding to the green area in Table 1, we introduce *KL-masking*, which masks out the actions that cause deviating from the reference policy when its likelihood is increased or decreased. More formally, we define *positive KL-masking* M^+_{KL} and *negative KL-masking* M^-_{KL} according to the relationship between the log-probabilities of the reference policy and current policy as follows:

$$M^+_{\mathrm{KL}}(s,a) := \begin{cases} 1 & \text{if} \quad R^{\pi_\theta}(s,a) > 0 \\ 0 & \text{otherwise} \end{cases}, \tag{5}$$

$$M^-_{\mathrm{KL}}(s,a) := \begin{cases} 1 & \text{if} \quad R^{\pi_\theta}(s,a) < 0 \\ 0 & \text{otherwise} \end{cases}, \tag{6}$$

where $R^{\pi_\theta}(s,a) := \log \pi_0(a|s) - \log \pi_\theta(a|s)$ and π_0 is a reference policy. Here, we assume that the reference policy in DfPO is a language model that can generate natural responses like a human. Intuitively, positive KL-masking masks out the actions that can cause degeneration problems when their likelihood is increased, and negative KL-masking masks out the actions that can cause degeneration problems when their likelihood is decreased. In other words, the positive/negative KL-masking aims to consider only actions for which the updated policy does not deviate from the reference policy even if the policy is updated to increase/decrease the likelihood of the action. Unlike existing methods that use the KL divergence between the current policy and reference policy as the reward penalty to prevent the degeneration problem, we adopt it to determine whether increasing/decreasing the likelihood of the actions cause the degeneration problem in the policy gradient updates.

3.3. Likelihood Max/Minimization with KL-Masking and Truncated Advantage Functions

Using the KL-masking defined before, we can separately consider state-action pairs that can increase or decrease the likelihood while preserving the naturalness. Now, for each state-action pair filtered by KL-masking, it is necessary to determine whether the likelihood should be increased or decreased to improve the task performance. To achieve this, we can easily determine the state-action pairs that need to maximize or minimize the likelihood in terms of task performance by truncating the advantage function A^{π_θ}, then update the policy with likelihood maximization and minimization as follows:

$$\begin{aligned}\nabla_\theta J(\theta) = & \underbrace{\mathbb{E}_{d^{\pi_\theta}}\left[M^{+}_{\text{KL}}(s,a)A^{\pi_\theta}(s,a)_{+}\nabla_\theta \log \pi_\theta(a|s)\right]}_{\substack{\text{Likelihood maximization for actions}\\ \text{with both positive advantage and log ratio}}} \\ & + \underbrace{\mathbb{E}_{d^{\pi_\theta}}\left[M^{-}_{\text{KL}}(s,a)A^{\pi_\theta}(s,a)_{-}\nabla_\theta \log \pi_\theta(a|s)\right]}_{\substack{\text{Likelihood minimization for actions}\\ \text{with both negative advantage and log ratio}}},\end{aligned} \tag{7}$$

where $(\cdot)_+ = \max(\cdot, 0)$ and $(\cdot)_- = \min(\cdot, 0)$. The first term in Eq. (7) corresponds to the likelihood maximization in Table 1, and it increases the likelihood of actions with positive advantages among actions with a lower likelihood in the current policy than in the reference policy. On the other hand, the second term in Eq. (7) corresponds to the likelihood minimization in Table 1, and it decreases the likelihood of actions with negative advantages among actions with a higher likelihood in the current policy than in the reference policy.

Connections to PPO The current best-performing online RL algorithm is PPO which optimizes the policy through the KL-penalized objective (i.e. Eq. (4)). The main difference between DfPO and PPO with a KL-penalized objective is the use of the log ratio between the reference policy and the current policy. In contrast to the PPO which uses the log ratio as a reward penalty, DfPO uses the log ratio to determine whether the policy gradient update causes degeneration and then excludes it for policy gradient updates. We also provide a mathematical connection to PPO in Appendix A. This result demonstrates that, to prevent the new policy from deviating from the reference policy, DfPO directly modifies the clipping technique in PPO, whereas PPO-based RLHF algorithms additionally incorporate KL regularization. This difference in the use of the log ratio allows DfPO can improve the task performance while preserving the naturalness without additional hyperparameter search.

3.4. Degeneration-free Policy Optimization

Finally, we present Degeneration-free Policy Optimization (DfPO), a new policy optimization method that can fine-tune LMs to generate texts that achieve improved downstream task scores, while preserving the naturalness of the generated texts. Figure 2 shows an illustrative example of the main mechanism of DfPO, and our algorithm can be summarized as follows: 1) KL-masking separately considers state-action pairs that do not deviate from the reference policy even if the likelihood is increased or decreased, 2) among the state-action pairs of the positive KL-masking, the promising actions with a positive advantage are increased which corresponds to the likelihood maximization part (red box) in Figure 2. Similarly, among the state-action pairs of the negative KL-masking, the unpromising actions with a negative advantage are decreased which corresponds to the likelihood minimization part (blue box) in Figure 2. The pseudocode for the whole process of DfPO can be found in Appendix B.4. Since our algorithm updates the policy to maximize or minimize the likelihood of generated sentences (i.e. sentence-level policy optimization), even if it is not provided in the form of Gym-like (Brockman et al., 2016) learning environment, it can be easily applied to any dataset if only the reward function is provided. For the advantage estimation, we use Generalized Advantage Estimation (GAE) (Schulman et al., 2018), but any other advantage estimation method can be used.

4. Related Work

Reinforcement learning for language models Training a language model to improve the downstream task score can be naturally considered as an RL problem (Ramamurthy et al., 2023; Snell et al., 2023; Jang et al., 2022). Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Liang et al., 2022; Kim et al., 2023; Ouyang et al., 2022; Rafailov et al., 2023) is one of the representative successes of fine-tuning LMs through reinforcement learning. However, optimizing LMs against the reward model by using RL is yet challenging due to the *degeneration problem* which generates responses diverging from human language (Lewis et al., 2017; Jang et al., 2020). Recently, as a benchmark of evaluating RL algorithms for fine-tuning language models, Ramamurthy et al. (2023) introduced (1) RL4LMs which is a modular library for optimizing LMs with RL, and (2) GRUE benchmark that is a set of generative NLP tasks with reward functions. Our work is based on RL4LMs and GRUE, and aims to address the degeneration in optimizing text generation with RL.

Direct policy optimization from preference dataset Recently, with a provided preference dataset, preference-based RL algorithms without an explicit reward have been proposed to achieve promising performance concurrently in RL (Hejna & Sadigh, 2023; Hejna et al., 2023) and NLP tasks (Rafailov et al., 2023). Among these studies, Rafailov et al. (2023) introduced *direct preference optimization* (DPO), directly addressing the KL-regularized reward maximization problem using only a simple binary classification loss. While DPO is more stable and efficient compared to previous PPO-based RLHF algorithms, it may also be sensitive to hyperparameters, as indicated by empirical observations presented in Appendix C.1.2. Additionally, the use of only a pre-collected dataset may pose challenges for improving performance in an online setting.

Stabilizing policy gradient methods Stabilizing policy gradient (PG) methods (Peters & Schaal, 2008) is essential to successfully optimize a policy, since PG methods use an *estimator* of the gradient of the expected return. TRPO (Schulman et al., 2015) provides a practical algorithm by making approximations to the theoretically justified algorithm that guarantees policy improvements. PPO (Schulman et al., 2017) is a simplified method from TRPO by introducing a clipped probability ratio, while attaining the data efficiency and reliability of TRPO. Unlike these methods, NLPO (Ramamurthy et al., 2023) is introduced by mainly considering text generation tasks that have much larger action space (i.e., a large number of tokens to select) than conventional decision-making tasks. NLPO mitigates the instability of policy optimization with action masking that learns to invalidate less relevant tokens. Unlike NLPO, our algorithm DfPO updates the policy only with desirable samples that can simultaneously improve the task score and avoid deviating from the reference policy.

5. Experiments

In this section, we show the experimental results of DfPO on generative NLP tasks including **text continuation** (IMDB) (Maas et al., 2011), **text detoxification** (REALTOXICITYPROMPTS) (Gehman et al., 2020), and **commonsense generation** (CommonGen) (Lin et al., 2020). The details of the experimental settings for each task can be found in Appendix B.

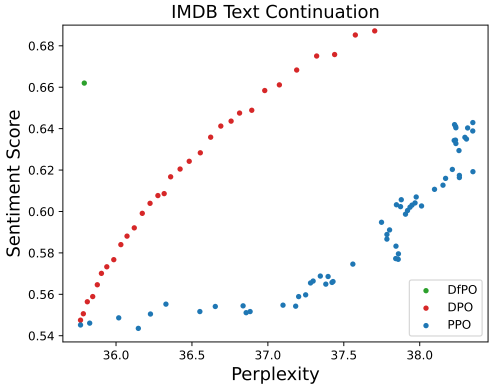


Figure 3. Frontier of sentiment score and perplexity for each algorithm on **IMDB text continuation task** with GPT-2 as a policy.

Baselines We compare the following algorithms to evaluate whether DfPO improves task performance while preserving the naturalness of the generated texts: 1) Zero Shot, a pre-trained language model without any fine-tuning of downstream tasks, 2) Supervised, a supervised fine-tuning model with datasets for downstream tasks, 3) PPO (Schulman et al., 2017), a policy gradient method that is state-of-the-art in discrete action space, 4) NLPO (Ramamurthy et al., 2023), a PPO-based policy optimization method for NLP tasks that effectively reduces the combinatorial action space with action masking, 5) DPO (Rafailov et al., 2023), a direct preference optimization method that directly optimizes the policy from the preference dataset. For PPO, NLPO, and DPO, KL divergence from the reference policy was used as a reward penalty, and *all results were obtained by hyperparameter tuning for the coefficient of the KL-regularization penalty*. Unlike baseline algorithms which **require massive computational costs for hyperparameter search**, DfPO aims to improve task performance while preserving the naturalness of the generated texts **without additional hyperparameter search**.

5.1. Evaluation on IMDB Text Continuation Task

We evaluate our algorithm on the IMDB text continuation task, which is one of the representative generative NLP tasks for evaluating the RL algorithms. The IMDB text continuation task aims to positively complete the movie review when given a partial review as a prompt. In this task, we use GPT-2 (117M parameters) and GPT-J (6B parameters) as a policy, and a trained sentiment classifier DistilBERT (Sanh et al., 2019) is provided as a reward function to train the policy and evaluate its task performance. The naturalness of the trained model is evaluated with a perplexity score.

Learning curves Figure 1 shows the learning curves for PPO and DfPO with GPT-2 as a policy. We also provide the learning curves for NLPO and DPO in Appendix C.1.1 and C.1.2. Here, DfPO and DfPO+Supervised indicate that the results of DfPO trained starting from a pre-trained language model and supervised fine-tuning model as initial policies, respectively. The results of baseline algorithms (PPO, NLPO, and DPO) show very sensitive performance on both sentiment score and perplexity to the coefficient of the KL-regularization penalty. This sensitivity to hyperparameters can be especially troublesome for fine-tuning the large-scale language models that require **massive computational costs** and also cause **ambiguity in model selection** after learning. In contrast to the sensitive results of baseline algorithms, DfPO shows results that successfully improve the sentiment score (i.e. task performance) while preserving their initial perplexity score (i.e. naturalness) without degeneration problem. Since DfPO maximizes only the sentiment score while preserving naturalness, DfPO 1) **does not require additional costs for hyperparameter search** and 2) **can simply select the final model with the best sentiment score** on the validation dataset, without any ambiguity in model selection. We further investigate whether DfPO is adoptable even when using a large language model as a policy. Figure 9 in Appendix C.1.3 shows the learning curves for DfPO with GPT-J (6B parameters) as a policy. Similar to the learning curve of DfPO with GPT-2, the results show that DfPO with a large language model successfully improves the sentiment score while preserving their initial perplexity score without additional hyperparameter search. Furthermore, we also provide an ablation study of DfPO to investigate the role of each part of DfPO in Appendix C.4.

Frontier of sentiment score and perplexity To investigate the effectiveness of DfPO in constrained policy optimization (i.e. maximizing sentiment score while preserving the perplexity), we evaluate the result of DfPO with the frontier of sentiment score and perplexity for PPO and DPO. Since DfPO has no ambiguity in model selection unlike other baseline algorithms, we select the model with the highest sentiment score on the validation dataset and evaluate it on the test dataset as a final result of DfPO. Figure 3 shows that DfPO achieves a better performance than the frontiers of PPO and DPO, even though DfPO was trained without hyperparameter search. Since PPO is sensitive to hyperparameters, the result shows the worst frontier even with sufficient hyperparameter search. DPO achieves a better frontier than PPO, but shows a worse frontier than DfPO due to its sensitivity to hyperparameters and limitation of offline learning.

Numerical comparison We provide the numerical results of the final model for each algorithm in Appendix C.1.4. Table 4 summarizes the performance of DfPO and baseline algorithms on the IMDB text continuation task. The results of baseline algorithms except DPO are from Ramamurthy et al. (2023), and the results of PPO, NLPO, and DPO are obtained by a massive hyperparameter search for the coefficients of the KL penalty. As shown in Table 4, even though the results of DfPO are obtained without additional hyperparameter search, DfPO outperforms or matches baseline algorithms in optimizing sentiment scores while preserving the perplexity of initial policy (i.e. Zero Shot model for DfPO, and Supervised model for DfPO+Supervised).

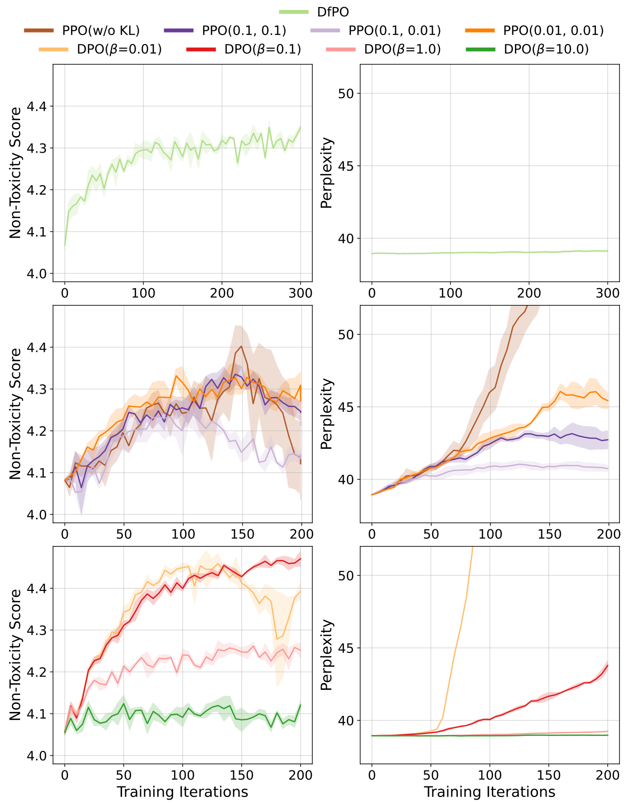


Figure 4. Learning curves of **(Top)** DfPO, **(Middle)** PPO, and **(Bottom)** DPO on **Text Detoxification task**. PPO (β, $\mathrm{KL}_{\text{target}}$) indicates the PPO that considers the KL-regularization as a reward penalty with KL coefficient β and target KL. All results are averaged over 3 runs, and the shaded area represents the standard error.

Results of diversity We also evaluate the diversity of DfPO, which is one of the most important factors when fine-tuning LMs by using reinforcement learning. Table 5 in Appendix C.1.5 summarizes the results for diversity metrics on the IMDB text continuation task. The results show that DfPO can generate more diverse sentences than baseline algorithms for all diversity metrics. Unlike existing RL methods that easily suffer from overoptimization issues as the policy deviates far from the reference policy, DfPO maintains diversity by optimizing the policy without deviating from the reference policy. As a result, DfPO effectively optimizes task scores while preserving the ability to generate diverse sentences.

5.2. Evaluation on Text Detoxification Task

We also evaluate our algorithm on the text detoxification task, REALTOXICITYPROMPTS, where the goal is to complete the sentence without toxicity. In this task, we use

Algorithms	Rouge-1	Rouge-2	Rouge-L	Rouge-LSum	Meteor	BLEU	BertScore	Cider	Spice
Zero Shot*	0.415	0.016	0.270	0.270	0.179	0.0	0.854	0.640	0.231
Supervised*	0.503 ± 0.001	0.175 ± 0.001	0.411 ± 0.001	0.411 ± 0.001	0.309 ± 0.001	0.069 ± 0.001	0.929 ± 0.000	1.381 ± 0.011	0.443 ± 0.001
PPO+Sup*	0.540 ± 0.005	0.204 ± 0.005	0.436 ± 0.004	0.436 ± 0.004	0.329 ± 0.003	0.076 ± 0.003	0.930 ± 0.001	1.474 ± 0.022	0.447 ± 0.004
NLPO+Sup*	0.537 ± 0.003	0.201 ± 0.004	0.431 ± 0.002	0.431 ± 0.002	0.326 ± 0.002	0.074 ± 0.003	0.930 ± 0.000	1.464 ± 0.025	0.448 ± 0.002
DfPO+Sup	**0.555 ± 0.003**	**0.221 ± 0.002**	**0.443 ± 0.001**	**0.443 ± 0.001**	**0.336 ± 0.001**	0.080 ± 0.001	0.929 ± 0.001	1.465 ± 0.028	0.445 ± 0.002

Table 2. Experimental results on **commonsense generation task**. The results with * are from the original paper (Ramamurthy et al., 2023). The table shows experimental results on the commonsense generation task with averages and standard errors. All results indicate averages and standard errors over 5 independent runs.

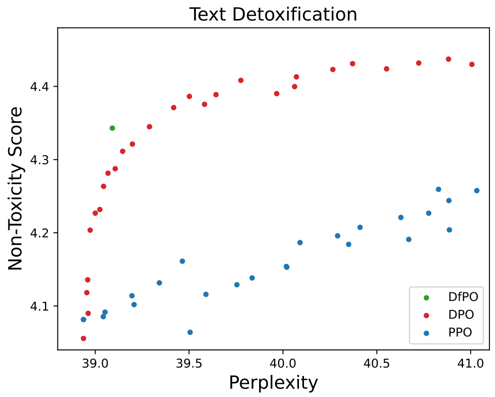


Figure 5. Frontier of sentiment score and perplexity on **text detoxification task** with GPT-J as a policy.

GPT-J (6B parameters) as a policy, and a fine-tuned toxicity classifier RoBERTa (Liu et al., 2019) model is provided as a reward function to train the policy and evaluate their non-toxicity score. The naturalness of the trained model is evaluated with a perplexity score.

Learning curves Figure 4 shows the learning curves for DfPO, PPO, and DPO on the text detoxification task. The results show that PPO and DPO are sensitive to the coefficient of KL-regularization penalty, then easily fail to preserve their initial perplexity (i.e. naturalness). In contrast to the sensitive results of baseline algorithms, DfPO successfully improves the non-toxicity score (i.e. task performance) while preserving their initial perplexity score (i.e. naturalness) without degeneration problem.

Frontier of non-toxicity score and perplexity Figure 5 shows that DfPO achieves a better performance than the frontiers of PPO and DPO, even though DfPO was trained without additional hyperparameter search. Since PPO is sensitive to the coefficient of the KL-regularization penalty, the result shows the worst frontier even with sufficient hyperparameter search. Although DPO achieves a similar frontier to DfPO, the ambiguity of model selection still remains a limitation.

Numerical comparison We provide the numerical results of the final model for each algorithm in Appendix C.2. Table 6 summarizes the performance of DfPO and baseline algorithms on the text detoxification task. The results show that DfPO successfully improves the non-toxicity score while preserving the perplexity of the initial policy.

5.3. Evaluation on Commonsense Generation Task

We additionally evaluate our algorithm on the commonsense generation task, where the goal is to generate a sentence describing a scene using given concepts. In this task, we use T5 (220M parameters) as a policy, and use a METEOR (Banerjee & Lavie, 2005) score as a reward function for training and evaluation of task performance, and BERTScore (Zhang et al., 2019) and SPICE (Anderson et al., 2016) as evaluation metrics for naturalness scores. Therefore, the goal is to successfully improve the METEOR score without degrading other task scores and naturalness scores. Table 2 summarizes the overall results of each algorithm on the commonsense generation task. As shown in Table 2, DfPO successfully achieves a high score on METEOR while not degrading other task scores and the naturalness scores. Furthermore, we provide the diversity results of DfPO and baseline algorithms in Appendix C.3.1. As shown in Table 7, DfPO shows similar diversity in all metrics for diversity only except MSTTR compared to other algorithms.

6. Conclusion

In this paper, we introduce Degeneration-free Policy Optimization (DfPO) that can fine-tune LMs to generate texts that achieve improved downstream task scores, while preserving the naturalness of the generated texts. The basic idea is to update the policy only with desirable samples that can simultaneously improve the task score and preserve naturalness. To achieve this, we develop KL-masking which masks out the samples that potentially cause deviating from the reference policy when likelihood max/minimization. Then, we devise a policy gradient method that separately performs likelihood maximization and minimization by using truncated advantage functions. We demonstrated the effectiveness of DfPO in optimizing task scores and preserving the naturalness of generated texts on various generative NLP tasks using diverse LMs. Although we assume a sufficiently fluent language model as a reference model, we believe that DfPO can be applied to various downstream tasks by leveraging recent high-performance sLLMs. We leave a more extensive attempt to apply DfPO to the RLHF and RLAIF framework with recent LLMs and sLLMs to future work. We expect that DfPO will serve as an important step towards providing a stable and robust RL method for LLM fine-tuning research.

Impact Statement

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

References

Anderson, P., Fernando, B., Johnson, M., and Gould, S. Spice: Semantic propositional image caption evaluation. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14*, pp. 382–398. Springer, 2016.

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

Banerjee, S. and Lavie, A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pp. 65–72, 2005.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Cohn, T., He, Y., and Liu, Y. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301. URL https://aclanthology.org/2020.findings-emnlp.301.

Glaese, A., McAleese, N., Trebacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Hejna, J. and Sadigh, D. Inverse preference learning: Preference-based RL without a reward function. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=gAP52Z2dar.

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

Jang, Y., Lee, J., and Kim, K.-E. Bayes-adaptive monte-carlo planning and learning for goal-oriented dialogues. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7994–8001, Apr. 2020. doi: 10.1609/aaai.v34i05.6308. URL https://ojs.aaai.org/index.php/AAAI/article/view/6308.

Jang, Y., Lee, J., and Kim, K.-E. GPT-critic: Offline reinforcement learning for end-to-end task-oriented dialogue systems. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=qaxhBG1UUaS.

Kim, C., Park, J., Shin, J., Lee, H., Abbeel, P., and Lee, K. Preference transformer: Modeling human preferences using transformers for rl, 2023.

Lewis, M., Yarats, D., Dauphin, Y. N., Parikh, D., and Batra, D. Deal or no deal? end-to-end learning for negotiation dialogues, 2017.

Liang, X., Shu, K., Lee, K., and Abbeel, P. Reward uncertainty for exploration in preference-based reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=OWZVD-l-ZrC.

Lin, B. Y., Zhou, W., Shen, M., Zhou, P., Bhagavatula, C., Choi, Y., and Ren, X. Commongen: A constrained text generation challenge for generative commonsense reasoning, 2020.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Maas, A., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., and Potts, C. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback, 2022.

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

Peters, J. and Schaal, S. Reinforcement learning of motor skills with policy gradients. *Neural networks*, 21(4):682–697, 2008.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2023.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., and Choi, Y. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *International Conference on Learning Representations*, 2023.

Sanh, V., Debut, L., Chaumond, J., and Wolf, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation, 2018.

Snell, C., Kostrikov, I., Su, Y., Yang, M., and Levine, S. Offline rl for natural language generation with implicit language q learning, 2023.

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

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT Press, 1998.

Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., and Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Connections to PPO

Before starting the derivation, we define the following stationary distributions:

$$d^\pi(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s),$$
$$d^\pi(s,a) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s, a_t = a),$$

where $s_0 \sim p_0$ and the actions are chosen according to π. In addition, the policy gradient objective can be formulated as Equation 1:

$$\nabla_\theta J_{\text{PG}}(\theta) = \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}[A^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a|s)].$$

A.1. PPO Objective

From the policy gradient objective, TRPO and PPO use the following gradient of the surrogate objective $\hat{J}_{\text{PG}}(\theta)$:

$$\begin{aligned}\nabla_\theta J_{\text{PG}}(\theta) &\approx \mathbb{E}_{s\sim d^{\pi_{\text{old}}}, a\sim\pi_\theta(\cdot|s)}[A^{\pi_{\text{old}}}(s,a)\nabla_\theta \log \pi_\theta(a|s)] \\ &= \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\left[A^{\pi_{\text{old}}}(s,a)\frac{\pi_\theta(a|s)\nabla_\theta \log \pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}\right] \\ &= \nabla_\theta \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\left[A^{\pi_{\text{old}}}(s,a)w_{\theta,\text{old}}(s,a)\right] =: \nabla_\theta \hat{J}_{\text{PG}}(\theta),\end{aligned}$$

where $w_{\theta,\text{old}}(s,a) := \frac{\pi_\theta(a|s)}{\pi_{\text{old}}(a|s)}$, and $\pi_{\text{old}}(a|s)$ denotes the previous policy before the update.

To stabilize this objective, PPO modifies the surrogate objective to penalize deviations of $\pi_\theta(a|s)$ from $\pi_{\text{old}}(a|s)$ (Schulman et al., 2017):

$$J_{\text{PPO}}(\theta) = \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\left[\min\{A^{\pi_{\text{old}}}(s,a)w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a)\text{clip}(w_{\theta,\text{old}}(s,a), 1-\epsilon, 1+\epsilon)\}\right]$$

for a hyperparameter $\epsilon > 0$. For three arbitrary variables u, A, and ϵ, we define following two functions:

$$\begin{aligned} f(w,A,\epsilon) &:= \min\{Aw, A\cdot\text{clip}(w,1-\epsilon,1+\epsilon)\}, \\ g(w,A,\epsilon) &:= \{\mathbb{I}\big(A>0, w<1+\epsilon\big) + \mathbb{I}\big(A<0, w>1-\epsilon\big)\},\end{aligned}$$

where $\mathbb{I}$ is an indicator function:

$$\mathbb{I}(\mathit{cond}) := \begin{cases} 1 & \text{if } \mathit{cond} = \texttt{True} \\ 0 & \text{otherwise}\end{cases}.$$

Then, the following equations hold:

$$\begin{aligned}\nabla_\theta J_{\text{PPO}}(\theta) &= \nabla_\theta \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\left[f\big(w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a), \epsilon\big)\right] \\ &= \nabla_\theta \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\left[g\big(w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a), \epsilon\big)A^{\pi_{\text{old}}}(s,a)w_{\theta,\text{old}}(s,a)\right],\end{aligned}$$

where the last equation can be obtained based on Figure 6. Consequently, the gradient of $J_{\text{PPO}}(\theta)$ is equivalent to the gradient of the following objective:

$$\hat{J}_{\text{PPO}}(\theta) := \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\left[g\big(w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a), \epsilon\big)A^{\pi_{\text{old}}}(s,a)w_{\theta,\text{old}}(s,a)\right].$$

Compared to the original surrogate objective $\hat{J}_{\text{PG}}(\theta)$, $g\big(w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a), \epsilon\big)$ is added to prevent deviations of $\pi_\theta(s,a)$ too far from $\pi_{\text{old}}(s,a)$.

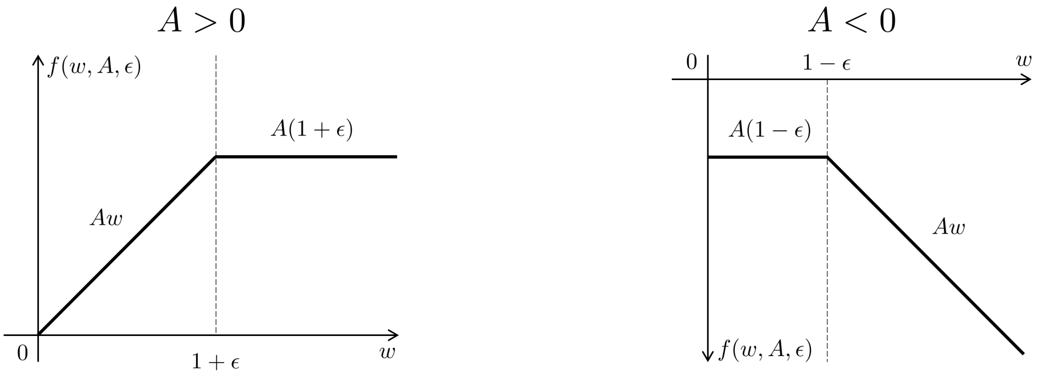


Figure 6. Plots are provided to illustrate the behavior of $f(w, A, \epsilon)$ with respect to w under two conditions: **(Left)** $A > 0$ and **(Right)** $A < 0$ (Schulman et al., 2017). For the case of $A > 0$, $\nabla_w f(w, A, \epsilon) = 0$ when $w \geq 1 + \epsilon$. Conversely, for $A < 0$, $\nabla_w f(w, A, \epsilon) = 0$ when $w \leq 1 - \epsilon$. In all other instances, $\nabla_w f(w, A, \epsilon) = \nabla_w (Aw) = A$. As a result, the relationship $\nabla_\theta f(w_\theta, A, \epsilon) = \nabla_\theta (g(w_\theta, A, \epsilon) A w_\theta)$ holds for arbitrary values of w_θ, A, and ϵ.

A.2. DfPO Objective

In NLP tasks, when we have a reward model, the environment interaction is not too expensive because generating an action sequence from π is equivalent to generating a trajectory from π. Thus, we will use the policy gradient objective instead of the surrogate objective. In addition, we want to penalize deviations of π_θ from π_0, instead of π_{old}. Hence, similar to the original PPO objective, incorporating $g\big(w_{\theta,0}(s,a), A^\pi(s,a), \epsilon\big)$ into the original policy gradient objective $J_{\text{PG}}(\theta)$ can be employed to avoid substantial deviations of $\pi_\theta(s,a)$ from $\pi_0(s,a)$:

$$\begin{aligned} J_{\text{PG}}(\theta) &= \mathbb{E}_{(s,a)\sim d^\pi}[A^\pi(s,a)] \\ &\approx \mathbb{E}_{(s,a)\sim d^\pi}\big[g\big(w_{\theta,0}(s,a), A^\pi(s,a), \epsilon\big) A^\pi(s,a)\big] =: \tilde{J}_{\text{PG}}(\theta) \end{aligned} \tag{8}$$

where $w_{\theta,0}(s,a) := \frac{\pi_\theta(a|s)}{\pi_0(a|s)}$. After setting $\epsilon = 0$, the gradient of $\tilde{J}_{\text{PG}}(\theta)$ is derived as follows:

$$\begin{aligned} \nabla_\theta \tilde{J}_{\text{PG}}(\theta) =& \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\big[\{\mathbb{I}\big(A^\pi(s,a) > 0, w_{\theta,0}(s,a) < 1\big) + \mathbb{I}\big(A^\pi(s,a) < 0, w_{\theta,0}(s,a) > 1\big)\} A^\pi(s,a) \nabla_\theta \log \pi_\theta(a|s)\big] \\ =& \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\big[\mathbb{I}\big(w_{\theta,0}(s,a) < 1\big) A^\pi(s,a)_+ \nabla_\theta \log \pi_\theta(a|s)\big] \\ & + \mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\big[\mathbb{I}\big(w_{\theta,0}(s,a) > 1\big) A^\pi(s,a)_- \nabla_\theta \log \pi_\theta(a|s)\big] \\ =& \underbrace{\mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\Big[M^+_{\text{KL}}(s,a) A^{\pi_\theta}(s,a)_+ \nabla_\theta \log \pi_\theta(a|s)\Big]}_{\substack{\text{Likelihood maximization for actions} \\ \text{with both positive advantage and log ratio}}} + \underbrace{\mathbb{E}_{(s,a)\sim d^{\pi_\theta}}\Big[M^-_{\text{KL}}(s,a) A^{\pi_\theta}(s,a)_- \nabla_\theta \log \pi_\theta(a|s)\Big]}_{\substack{\text{Likelihood minimization for actions} \\ \text{with both negative advantage and log ratio}}}. \end{aligned}$$

The last resulting term is equivalent to the DfPO objective in Eq. (7), which successfully improves the task performance while preserving the naturalness of generated texts.

In summary, we confirm a remarkably high correlation between our DfPO objective and the clipped objective proposed by Schulman et al. (2017). After reformulating PPO objective as:

$$\begin{aligned} \hat{J}_{\text{PPO}}(\theta) &= \mathbb{E}_{(s,a)\sim d^{\pi_{\text{old}}}}\Big[g\big(w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a), \epsilon\big) A^{\pi_{\text{old}}}(s,a) w_{\theta,\text{old}}(s,a)\Big] \\ &= \mathbb{E}_{s\sim d^{\pi_{\text{old}}}, a\sim\pi_\theta(\cdot|s)}\Big[g\big(w_{\theta,\text{old}}(s,a), A^{\pi_{\text{old}}}(s,a), \epsilon\big) A^{\pi_{\text{old}}}(s,a)\Big], \end{aligned} \tag{9}$$

we observe the primary difference between the DfPO objective (8) and the reformulated PPO objective (9) lies in the use of π_0 instead of π_{old}. This distinction enables us to prevent the deviation of π from π_0 without the need for an additional KL regularization penalty, which is commonly employed in PPO-based RLHF algorithms.

B. Experimental Details

B.1. Task Specification and Hyperparameter Configuration

Table 3 summarizes the task specifications and hyperparameter settings that we used in our experiments. For a fair comparison, we use exactly same settings of task, decoding strategy and tokenizer to train and evaluate each algorithms. that used in (Ramamurthy et al., 2023). We also provide the settings of hyperparameters that used in our experiments.

		IMDB (Text Continuation)	REALTOXICITYPROMPTS (Text Detoxification)	CommonGen (Commonsense Generation)
Task Specification	task preference metric	Learned Sentiment Classifier	Learned Toxicity Classifier	METEOR
	naturalness metric	Perplexity	Perplexity	SPICE
Decoding	sampling	top-k ($k = 50$)	top-k ($k = 50$)	beam search ($n = 5$)
	min length	48	24	5
	max new tokens	48	24	20
Tokenizer	padding side	left	left	left
	truncation side	left	left	-
	max length	64	64	20
Hyper-parameters	batch size	16	16	16
	learning rate	0.00001	0.00001	0.00001
	discount factor	0.99	0.99	0.99
	gae lambda	0.95	0.95	0.95

Table 3. Task specification and hyperparameter configuration used in our experimental results on IMDB, REALTOXICITYPROMPTS, and CommonGen domain.

B.2. Experimental Settings of DPO

In order to show the effectiveness of DfPO, we also compare the performance with DPO on IMDB text continuation and text detoxification tasks. In contrast to the online RL setting (i.e. PPO, NLPO, and DfPO) given the reward function considered in this paper, DPO assumes an offline RL setting given a pre-collected pairwise dataset. For the pairwise dataset, we created 20000 preference pairs for each task using the ground-truth reward model, and used it for fine-tuning the LMs with DPO.

B.3. Implementation Details of DfPO

We implement DfPO based on the codebase of RL4LMs (Ramamurthy et al., 2023), which is one of the representative RL library for NLP tasks. For the policy network, we use GPT-2 and GPT-J (6B) for the IMDB text continuation task, GPT-J (6B) for the text detoxification task, and T5 for the commonsense generation task. We provide the pseudocode of our algorithm DfPO in Algorithm 1. For the advantage function estimation, we use Generalized Advantage Estimation (GAE) (Schulman et al., 2018), but any other advantage function estimation method can be used. Since our algorithm generates sentences and then maximizes or minimizes the likelihood of generated sentences, we implemented our algorithm as a sentence-level policy optimization (not a word-level policy optimization).

B.4. Pseudocode of DfPO

Algorithm 1 Degeneration-free Policy Optimization (DfPO)

```
Input: Training dataset $\mathcal{D} = \{(s_t^j, a_t^j, s_{t+1}^j)_{t=0}^T\}_{j=1}^N$, a policy network $\pi_\theta$ with parameter $\theta$, a value network $V_\phi$ with parameter $\phi$, a reference policy $\pi_0$
for each iteration $i$ do
    Define KL-masking with $\pi_0$ as Eq. (5) and Eq. (6):
        $$M_{\text{KL}}^{+}(s,a) := \begin{cases} 1 & \text{if} \quad R^{\pi_\theta}(s,a) = \log \pi_0 - \log \pi_\theta > 0 \\ 0 & \text{otherwise} \end{cases},$$
        $$M_{\text{KL}}^{-}(s,a) := \begin{cases} 1 & \text{if} \quad R^{\pi_\theta}(s,a) = \log \pi_0 - \log \pi_\theta < 0 \\ 0 & \text{otherwise} \end{cases},$$
    Sample mini-batch of initial states $\{s_0^m\}_{m=1}^M$ from $\mathcal{D}$
    Generate trajectories $\mathcal{T} = \{s_t^m, a_t^m\}$ by running policy $\pi_\theta$
    Update policy via likelihood max/minimization with KL-masking and truncated advantage functions as Eq. (7):
        $$\arg\max_\theta \sum_{m=1}^{M}\sum_{t=0}^{T} M_{\text{KL}}^{+}(s_t^m, a_t^m) A_\phi^{\pi_\theta}(s_t^m, a_t^m) \nabla_\theta \log \pi_\theta(a_t^m|s_t^m) + \sum_{m=1}^{M}\sum_{t=0}^{T} M_{\text{KL}}^{-}(s_t^m, a_t^m) A_\phi^{\pi_\theta}(s_t^m, a_t^m) \nabla_\theta \log \pi_\theta(a_t^m|s_t^m)$$
    Update the value function $V_\phi^{\pi_\theta}$:
        $$\arg\min_\phi \sum_{m=1}^{M}\sum_{t=0}^{T} (V_\phi^{\pi_\theta}(s_t^m) - R(s_t^m, a_t^m))^2$$
end for
Output: updated policy $\pi_\theta$
```

C. Additional Experimental Results

C.1. Experimental Results on IMDB Text Continuation Task

C.1.1. Learning Curve of NLPO

We provide the results of NLPO with KL-regularization penalty on the IMDB text continuation task. As shown in Figure 7, NLPO with KL-regularization penalty also shows very sensitive performance on both sentiment score and perplexity to hyperparameter.

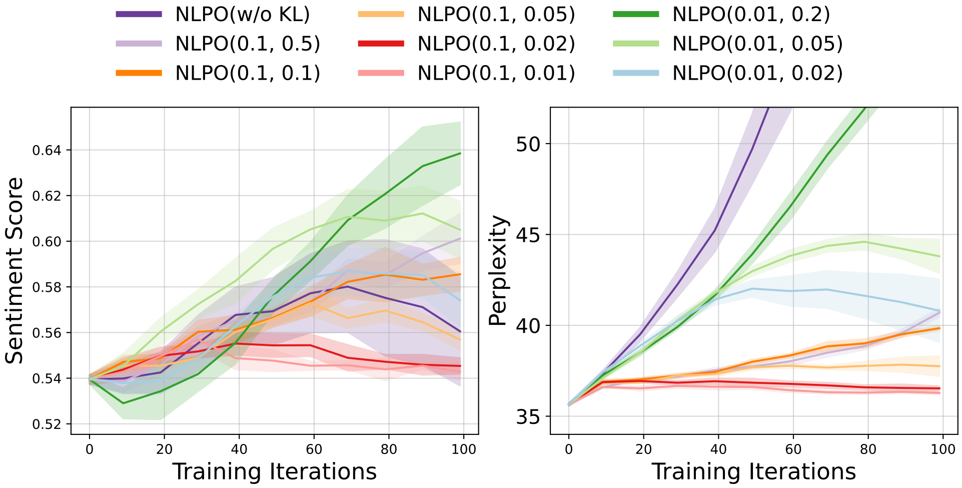


Figure 7. Averaged learning curve over 5 runs of NLPO with KL-regularization penalty on **IMDB text continuation** task for varying KL coefficient and target KL. NLPO (β, $\text{KL}_{\text{target}}$) indicates the NLPO that considers the KL-regularization as a reward penalty with KL coefficient β and target KL. The goal of the IMDB text continuation task is to learn a policy that maximizes the sentiment score (i.e. task reward) while preserving the perplexity (i.e. naturalness) of the initial policy. However, as shown in the results, NLPO with KL-regularization penalty shows very sensitive performance on both sentiment score and perplexity to hyperparameter.

C.1.2. Learning Curve of DPO

We provide the results of DPO with various hyperparameters for the KL-regularization penalty (i.e.β) on the IMDB text continuation task. As shown in Figure 8, DPO with KL-regularization penalty also shows very sensitive performance on both sentiment score and perplexity to hyperparameter.

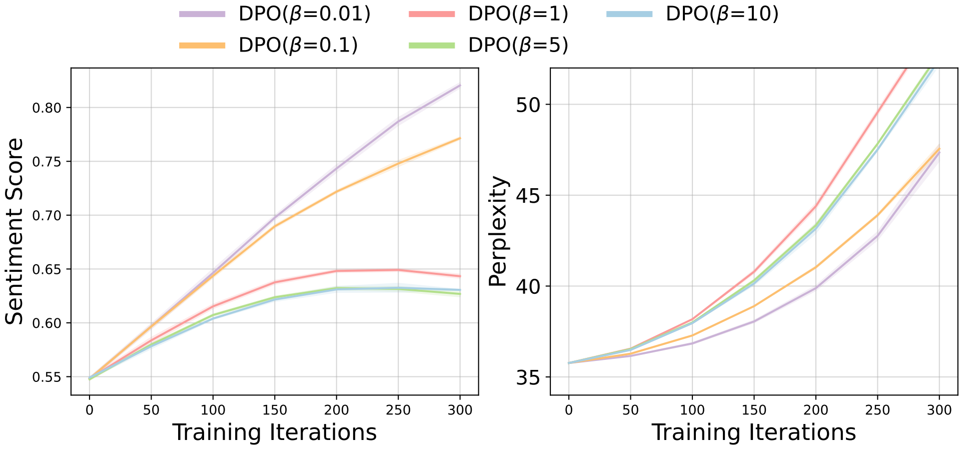


Figure 8. Averaged learning curve over 5 runs of DPO on **IMDB text continuation** task for varying KL coefficient β. DPO(β) indicates the DPO that considers the KL-regularization as a reward penalty with KL coefficient β. As shown in the results, DPO shows very sensitive performance on both sentiment score and perplexity to hyperparameter.

C.1.3. Learning Curve with Large Language Model

We also provide the result of DfPO with a large language model on the IMDB text continuation task. For the large language model, we use GPT-J (6B parameters) as initial and reference policies. As shown in Figure 9, DfPO can improve task scores while preserving the naturalness of the generated texts, even when using a large language model (GPT-J-6B) as a policy.

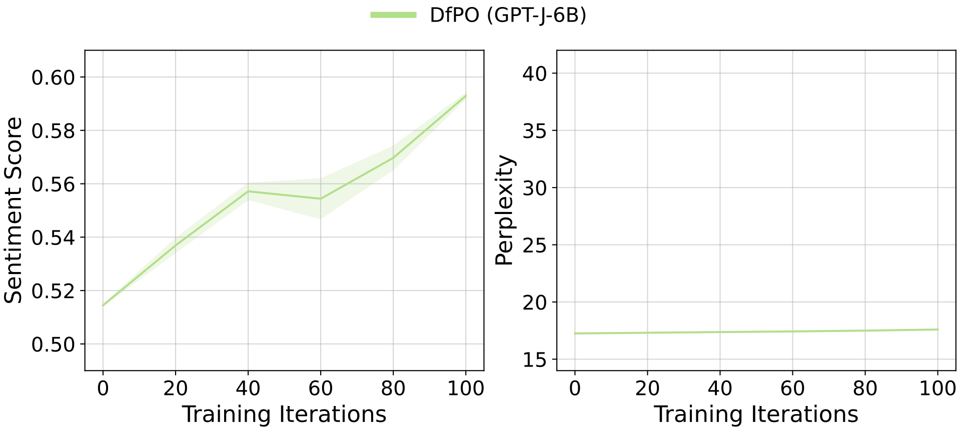


Figure 9. Experimental results of DfPO with large language model on **IMDB text continuation task**. The plots show the results of DfPO that was trained starting with GPT-J (6B parameters) model. All results are averaged over 5 runs, and the shaded area represents the standard error.

C.1.4. Numerical Results on IMDB Text Continuation Task

We also provide the numerical results of each algorithm on the IMDB text continuation task. Since the baseline algorithms are sensitive to hyperparameters, it is difficult to select the best model from the whole results. Therefore, for the results of baseline algorithms (except DPO), we use the the results from the original paper (Ramamurthy et al., 2023) which are recorded by massive hyperparameter tuning. For the result of DPO, we reports DPO (S/P) which are selected by a similar sentiment score and perplexity with the results of DfPO, respectively. Table 4 summarizes the performance of DfPO and baseline algorithms on the IMDB text continuation task. Even though the results of DfPO are obtained without additional hyperparameter search, DfPO outperforms or matches baseline algorithms in optimizing sentiment scores while preserving the perplexity of initial policy (i.e. Zero Shot model for DfPO, and Supervised model for DfPO+Supervised, respectively).

	Algorithms	**Sentiment Score ↑**	**Perplexity ↓**
Cold Starting	Zero Shot*	0.489 ± 0.006	32.171 ± 0.137
	Supervised*	0.539 ± 0.004	35.472 ± 0.074
	PPO*	0.602 ± 0.012	33.816 ± 0.233
	NLPO*	0.611 ± 0.023	33.832 ± 0.283
	DfPO (**ours**)	0.620 ± 0.009	32.563 ± 0.082
Warm Starting (Online)	PPO+Supervised*	0.626 ± 0.014	35.049 ± 0.347
	NLPO+Supervised*	0.620 ± 0.014	34.816 ± 0.340
	DfPO+Supervised (**ours**)	**0.663** ± **0.006**	35.796 ± 0.077
(Offline)	DPO+Supervised (S)	0.661 ± 0.004	37.076 ± 0.091
	DPO+Supervised (P)	0.551 ± 0.003	35.785 ± 0.013

Table 4. Experimental results on **IMDB text continuation task** with GPT-2 as a policy model. Cold starting indicates the results of each algorithm trained starting with the pre-trained language model (i.e. Zero Shot), and Warm starting indicates the results of each algorithm trained starting with the supervised fine-tuning model (i.e. Supervised). The results with * are from the original paper (Ramamurthy et al., 2023), and DPO (S/P) indicate the results selected by a similar sentiment score and perplexity with the results of DfPO, respectively. (All learning curves and frontier results for DPO can be found in Appendix C.1.2.) All results indicate averages and standard errors over 5 independent runs.

C.1.5. Diversity Results on IMDB Text Continuation Task

We also evaluate the diversity of each algorithm, which is one of the most important factors when fine-tuning LMs by using reinforcement learning. Table 5 summarizes the results for diversity metrics on the IMDB text continuation task. The results show that DfPO can generate more diverse sentences than baseline algorithms for all diversity metrics. Unlike existing RL methods that easily suffer from overoptimization issues as the policy deviates far from the reference policy, DfPO maintains diversity by optimizing the policy without deviating from the reference policy. As a result, DfPO effectively optimizes task scores while preserving the ability to generate diverse sentences.

Algorithms	**MSTTR**	**Distinct$_1$**	**Distinct$_2$**	**H$_1$**	**H$_2$**	**Unique$_1$**	**Unique$_2$**
Zero Shot*	0.682 ± 0.001	0.042 ± 0.001	0.294 ± 0.001	8.656 ± 0.004	13.716 ± 0.003	5063 ± 15	47620 ± 238
Supervised*	0.682 ± 0.001	0.047 ± 0.001	0.312 ± 0.002	8.755 ± 0.012	13.806 ± 0.016	5601 ± 57	51151 ± 345
PPO*	0.664 ± 0.007	0.042 ± 0.001	0.278 ± 0.005	8.529 ± 0.037	13.366 ± 0.119	5108 ± 204	45158 ± 961
PPO+Sup*	0.668 ± 0.004	0.048 ± 0.002	0.307 ± 0.008	8.704 ± 0.053	13.656 ± 0.066	5757 ± 324	50522 ± 1514
NLPO*	0.670 ± 0.002	0.043 ± 0.002	0.286 ± 0.006	8.602 ± 0.049	13.530 ± 0.076	5179 ± 196	46294 ± 1072
NLPO+Sup*	0.672 ± 0.006	0.048 ± 0.002	0.310 ± 0.012	8.725 ± 0.090	13.709 ± 0.174	5589 ± 140	50734 ± 1903
DfPO	0.711 ± 0.009	0.059 ± 0.004	0.369 ± 0.019	9.100 ± 0.119	14.386 ± 0.217	7609 ± 612	61217 ± 3415
DfPO+Sup	0.711 ± 0.007	0.061 ± 0.003	0.378 ± 0.014	9.155 ± 0.089	14.450 ± 0.151	7781 ± 435	62868 ± 2502

Table 5. Experimental results for diversity on **IMDB text continuation task** with GPT-2 as a policy model. The results with * are from the original paper (Ramamurthy et al., 2023). All results indicate averages and standard errors over 5 independent runs.

C.2. Experimental Results on Text Detoxification Task

We provide the numerical results of the final model for each algorithm. Table 6 summarizes the performance of DfPO and baseline algorithms on the text detoxification task. The results show that DfPO successfully improves the non-toxicity score while preserving the perplexity of the initial policy. PPO (N/P) and DPO (N/P) indicate the results selected by a similar non-toxicity score and perplexity with the results of DfPO, respectively.

Algorithms	**Non-Toxicity ↑**	**Perplexity ↓**
GPT-J(6B)	4.066 ± 0.036	39.463 ± 0.000
GPT-J(6B)+PPO (N)	4.347 ± 0.194	52.610 ± 2.342
GPT-J(6B)+PPO (P)	4.102 ± 0.019	39.207 ± 0.019
GPT-J(6B)+DPO (N)	4.343 ± 0.026	39.266 ± 0.059
GPT-J(6B)+DPO (P)	4.263 ± 0.042	39.083 ± 0.046
GPT-J(6B)+DfPO(**ours**)	4.343 ± 0.018	39.091 ± 0.071

Table 6. Experimental results on **text detoxification task**. PPO (N) and PPO (P) indicate the results selected by a similar non-toxicity score and perplexity with the results of DfPO, respectively. All results indicate averages and standard errors over 3 independent runs.

C.3. Experimental Results on Commonsense Generation Task

C.3.1. Diversity Results on Commonsense Generation Task

We provide the diversity results of DfPO and baseline algorithms on commonsense generation task. As shown in Table 7, DfPO shows similar diversity in all metrics for diversity only except MSTTR compared to other algorithms.

Algorithms	**MSTTR**	**Distinct$_1$**	**Distinct$_2$**	**H$_1$**	**H$_2$**	**Unique$_1$**	**Unique$_2$**
Zero Shot*	0.430	0.090	0.335	5.998	7.957	345	1964
Supervised*	0.509 ± 0.001	0.101 ± 0.001	0.339 ± 0.001	6.531 ± 0.006	10.079 ± 0.016	304 ± 7	2159 ± 25
PPO+Sup*	0.514 ± 0.004	0.105 ± 0.002	0.378 ± 0.008	6.631 ± 0.053	10.270 ± 0.064	507 ± 17	2425 ± 73
NLPO+Sup*	0.516 ± 0.006	0.106 ± 0.002	0.377 ± 0.008	6.634 ± 0.044	10.260 ± 0.077	506 ± 4	2401 ± 39
DfPO+Sup	0.489 ± 0.007	0.101 ± 0.003	0.369 ± 0.007	6.552 ± 0.017	10.178 ± 0.023	479 ± 11	2383 ± 24

Table 7. Experimental results for diversity on **commonsense generation task**. The results with * are from the original paper (Ramamurthy et al., 2023). All results indicate averages and standard errors over 5 independent runs.

C.3.2. Learning Curve of DfPO on Commonsense Generation Task

We also provide the learning curve of DfPO on the commonsense generation task. As shown in Figure 10, DfPO can improve task scores (i.e. METEOR) while preserving the naturalness (i.e. Perplexity) of the generated texts.

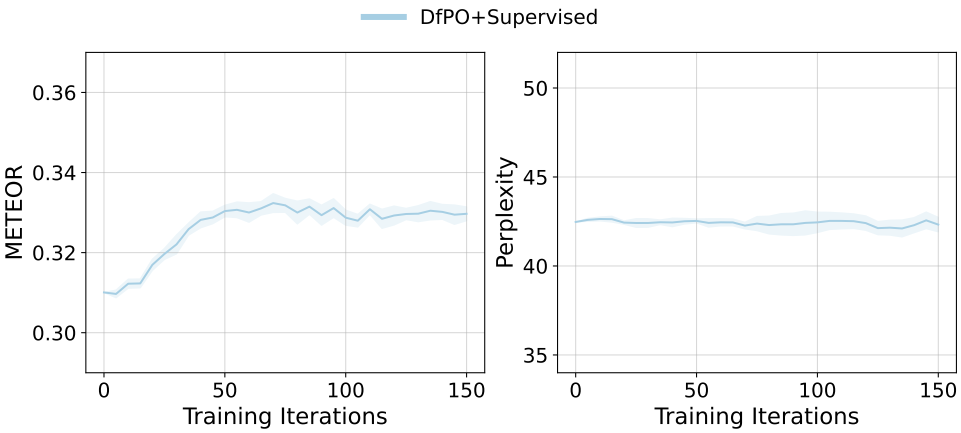


Figure 10. Experimental results of DfPO on **commonsense generation task**. All results are averaged over 5 runs, and the shaded area represents the standard error.

C.4. Ablation Study

To study the role of each part of the objective, we provide ablation study results of DfPO on the IMDB text continuation task. We compare the results of methods trained with the four types of objectives (Policy gradient, Likelihood Maximization, Likelihood Minimization, and DfPO), and the details of each model are provided below. As shown in Figure 11, the naive policy gradient (PG), similar to PPO without the KL regularization penalty in Figure 1, improves task performance but the perplexity diverges (i.e. deteriorates the naturalness of generated texts). In the case of updating the policy with only likelihood minimization, it fails to improve the task performance and also preserve the naturalness of generated texts. In addition, when updating the policy with only likelihood maximization, the perplexity does not diverge, but the task performance is improved relatively low compared to DfPO. Therefore, each part of the objective in DfPO is necessary to successfully improve the task performance while preserving the naturalness of generated texts.

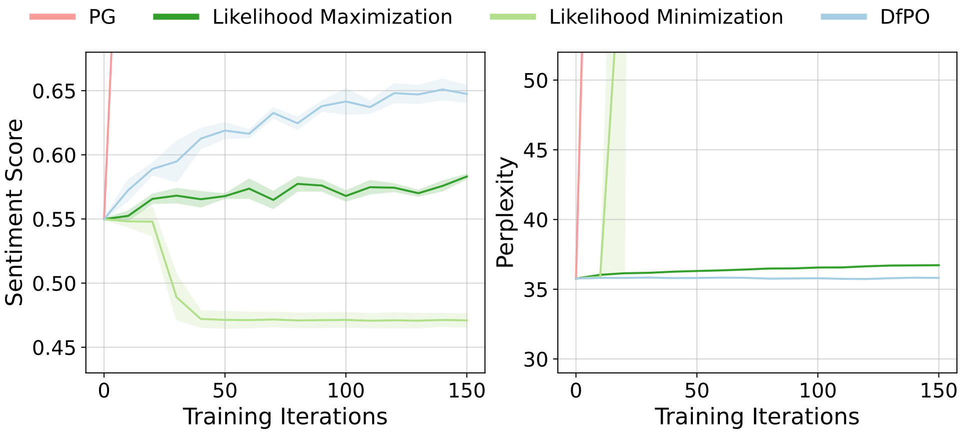


Figure 11. Ablation study results of DfPO on **IMDB text continuation task**. All results are averaged over 5 runs, and the shaded area represents the standard error.

Degeneration-free Policy Optimization (DfPO)

To show the role of each part of the objective, we compare the result of DfPO trained with the following objective, which is exactly the same as the result of DfPO+Supervised in Figure 1:

$$\nabla_\theta J(\theta) = \underbrace{\mathbb{E}_{a\sim\pi_\theta(s)}\left[M^{+}_{\text{KL}}(s,a)A^{\pi_\theta}(s,a)_{+}\nabla_\theta \log\pi_\theta(a|s)\right]}_{\substack{\text{Likelihood maximization for actions}\\ \text{with both positive advantage and log ratio}}} + \underbrace{\mathbb{E}_{a\sim\pi_\theta(s)}\left[M^{-}_{\text{KL}}(s,a)A^{\pi_\theta}(s,a)_{-}\nabla_\theta \log\pi_\theta(a|s)\right]}_{\substack{\text{Likelihood minimization for actions}\\ \text{with both negative advantage and log ratio}}}. \tag{10}$$

Likelihood Maximization

First, we compare the results of updating the policy through the only likelihood maximization, which corresponds to the first term of the main objective of DfPO:

$$\nabla_\theta J(\theta) = \underbrace{\mathbb{E}_{a\sim\pi_\theta(s)}\left[M^{+}_{\text{KL}}(s,a)A^{\pi_\theta}(s,a)_{+}\nabla_\theta \log\pi_\theta(a|s)\right]}_{\substack{\text{Likelihood maximization for actions}\\ \text{with both positive advantage and log ratio}}}. \tag{11}$$

Likelihood Minimization

Second, we compare the results of updating the policy through the only likelihood minimization, which corresponds to the second term of the main objective of DfPO:

$$\nabla_\theta J(\theta) = \underbrace{\mathbb{E}_{a\sim\pi_\theta(s)}\left[M^{-}_{\text{KL}}(s,a)A^{\pi_\theta}(s,a)_{-}\nabla_\theta \log\pi_\theta(a|s)\right]}_{\substack{\text{Likelihood minimization for actions}\\ \text{with both negative advantage and log ratio}}}. \tag{12}$$

Policy Gradient (PG)

We also compare the result of naive policy gradient update without KL-masking and advantage truncating as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{a\sim\pi_\theta(s)}\big[A^{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(a|s)\big]. \tag{13}$$

C.5. Results of PPO with SFT

We conducted additional experiments for PPO+SFT, where the learning objective is aggregated with PPO and supervised fine-tuning objectives. We provide the results of PPO+SFT with KL-regularization penalty on the **IMDB text continuation task**. As shown in Figure 12, PPO+SFT with KL-regularization penalty also shows very sensitive performance on both sentiment score and perplexity to hyperparameter.

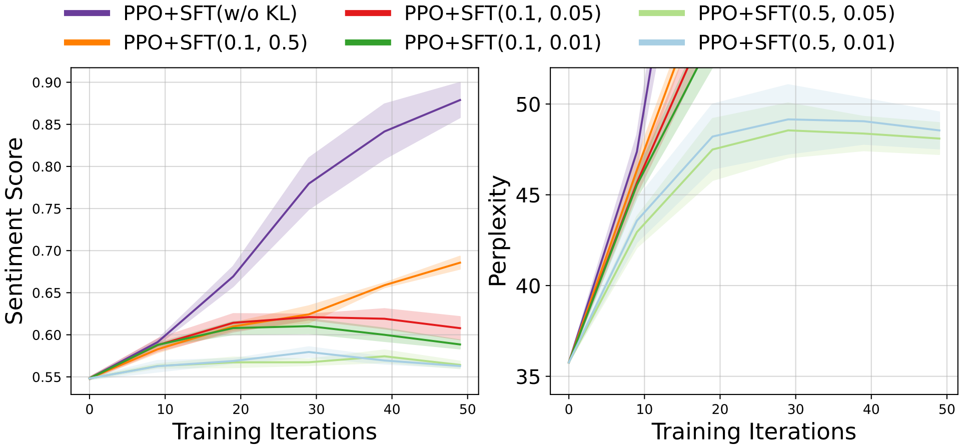


Figure 12. Averaged learning curve over 5 runs of PPO+SFT with KL-regularization penalty on IMDB text continuation task for varying KL coefficient and target KL. PPO+SFT (β, KL$_{\text{target}}$) indicates the PPO+SFT that considers the KL-regularization as a reward penalty with KL coefficient β and target KL. The goal of the IMDB text continuation task is to learn a policy that maximizes the sentiment score (i.e. task reward) while preserving the perplexity (i.e. naturalness) of the initial policy. However, as shown in the results, PPO+SFT with KL-regularization penalty shows very sensitive performance on both sentiment score and perplexity to hyperparameter.

D. Qualitative Results

In this section, we demonstrate through qualitative examples that DfPO improves task performance while maintaining the naturalness of generated texts. We provide generation results from DfPO and baseline algorithms. We selected the same prompts as used in (Ramamurthy et al., 2023), and the generation results of baseline algorithms are from (Ramamurthy et al., 2023). As shown in the examples, the generation results of DfPO show maintaining naturalness, which does not diverge from human language. On the other hand, PPO-no-KL and NLPO-no-KL, which do not use the KL regularization penalty, show results with poor naturalness of generated texts.

Sample 1

Prompt:
A man and his wife are not getting along because of the wife's jealousy and drinking problems. When the wife disappears, her sister begins a vigorous search involving the

References:
local police, state government, and a television crime show. What she finds out stuns the entire community. Good drama with lots of fine performances; based on a true story

Zero-Shot:
husband's sister, which leads several men to a remote cabin and then a boat. The wife and her husband then start to see the same things. Finally, a strange woman is driving out to her cabin and discovers the husband's disappearance but

Supervised:
amily tree. The widower, now widowed, returns, the daughter, daughter is not pleased when his wife is mistaken for one of her sons, and the patriarch, then a wealthy man, falls victim to a jealous husband. The

PPO:
police and the local public. At the end, the woman is killed and the wife is found naked and with a red head tattooed on her back. This is part of a special feature called"On the Road, Hidden" which was

PPO-no-KL:
police and other local governments. Written by G. Scott Fitzgerald, who also stars in this production, "A Very Merry Christmas" is a wonderful family movie with a wonderful soundtrack and wonderful characters. It brings home the true beauty of small-

NLPO:
mother while the brother goes under. After all, his sister, who was apparently kidnapped but who turns out to be a man, is no longer in her place. If a film that was made in the 1950s

NLPO-no-KL:
friends of the wife, and also the friend of the wife of the woman that is missing, and eventually he begins to find that she has gone to Germany and he has made a very good friend. There is a good scene between John and

DfPO:
town of Berenner, California. This is an excellent film, and it was made one year before George Cloorman (Stick With Her). It's as if George was already aware of the historical context

Sample 2

Prompt:
A very credible and unsettling movie portraying the effects of war on the general public, Shame gives us a chilling understanding oh Ingmar Bergman’s beliefs. The absence of color and sound (as in a soundtrack) helps to give the film a more realistic feeling. There is no soft or hard lighting or a variety of camera angles to give the story a charming or dramatic effect. Everything is very simply portrayed as it would be in reality. At times the storyline was confusing, but perhaps in order to give a sense of the characters’ frame of mind how they too do now know what is happening or why. The simplicity of the

References:
film made it somewhat boring, but I could understand how the director is trying not to glamorize the story and points. He depicted the harsh and unsettling events and effects of war. In the beginning the characters were introduced and shown as content with the lighter settings

Zero-Shot:
camera work is at once apparent and fascinating in the manner in which it affects the way things are portrayed. In other films these two have been referred to from time to time in a form and manner more suited to the screen. The film shows

Supervised:
"Dakota Clan" and his sister, the two become friends with Ted Masters who soon finds the chance to fly back to their home land and go back to save his brother’s life and his family. The

PPO:
characters relationships is so engrossing and the great and unique story presented by each pair of brothers and sisters creates a truly gripping and enjoyable experience. To put it simply, I strongly recommend to everyone who plays the role to buy this DVD!!!

PPO-no-KL:
movie helps it to flow along in a believable way. Rating 4 out of 5 Great Movie! I highly recommend these movies! I highly recommend these movies! Rating 5 out of 5 Great Movie! I highly recommend these movies! Rating 10 of"

NLPO:
novel makes it one of the first books I have read by a British actor. It is a fun way for someone who likes their books, like me, to sit through a novel. If you like to read novels, check it out.

NLPO-no-KL:
setting makes the final product very enjoyable to watch. Highly recommended!! Michael Bowerman, SPSA Film Festival Host, SPSA, WA (December 17, 2006), 8/17/2006 Great photography, good acting,

DfPO:
screenplay which follows that, makes for an excellent point, by telling this to the general public at hand with an all-merse narration and sound. The scene with Sgt Columbo fighting for our pride is actually excellent and

Sample 3

Prompt:
Shakespearean language is wild and rambling, saturated in multiple meanings, word play and metaphor. To be understood it must be wrangled and tamed by an actor with the strength and knowledge to do so. When an actor fails, the words pour forth in a torrent of incomprehensible words, but when he succeeds, the English language springs to life with an immediacy and vibrancy that takes your breath away. Al Pacino is one such actor, and here displays an incredible level of clarity and control that, were there any justice,

References:
would sweep every award in the offering. He meets the challenge of presenting Shylock head on, and delivers an extraordinarily subtle and nuanced performance. It would be a crime if we never got the opportunity to see what he does with King Lear.<br /><br />The supporting cast

Zero-Shot:
might still be seen in the world today. This movie, like "Vacation", is an effort at telling the tale of two families from different races. One is African but never grew up the other. One of them, a man

Supervised:
his performance (so far) would seem mere shadow. He is truly in the middle of a movie, and this film is one of those films where he can be convincing in it (and his trademark acting, as you can see in the

PPO:
would have made him an instant-classic. This movie captures the spirit of childhood and has everyone in the world watching it. It’s a love story and it’s really heartbreaking. And at the same time, it’s a classic. It

PPO-no-KL:
will bring victory to the fight against hatred and extremism. Al Pacino is a very beautiful, well executed actor who brings an absolute intensity and sensibility to his portrayal of love and loyalty. A must see! Recommended! Also, an Excellent

NLPO:
I would watch and enjoy. 7/10, Oscar Wilde is always a pleasure to see. A must see. Highly recommend. Highly recommend! One of my top ten films of all time. A must see!!

NLPO-no-KL:
the whole film would have fallen to the same fate, just as it did just a couple hundred years ago. Don’t miss it. It’s a real classic. Highly Recommended. ⋆ outta five stars for it!

DfPO:
hardly could be conceived, couldn´ve only appear to be a dream. His acting speaks for an incredible combination of passion and fear; his portrayal of Shakespeare that comes from passion is mesmerishing. His accent is reminiscent of the softening with