On Reinforcement Learning and Distribution Matching for Fine-Tuning Language Models with no Catastrophic Forgetting

Tomasz Korbak[*]
University of Sussex
tomasz.korbak@gmail.com

Hady Elsahar
Naver Labs Europe
hady.elsahar@gmail.com

Germán Kruszewski
Naver Labs Europe
german.kruszewski@naverlabs.com

Marc Dymetman[†]
marc.dymetman@gmail.com

Abstract

The availability of large pre-trained models is changing the landscape of Machine Learning research and practice, moving from a "training from scratch" to a "fine-tuning" paradigm. While in some applications the goal is to "nudge" the pre-trained distribution towards preferred outputs, in others it is to steer it towards a different distribution over the sample space. Two main paradigms have emerged to tackle this challenge: Reward Maximization (RM) and, more recently, Distribution Matching (DM). RM applies standard Reinforcement Learning (RL) techniques, such as Policy Gradients, to gradually increase the reward signal. DM prescribes to first make explicit the target distribution that the model is fine-tuned to approximate. Here we explore the theoretical connections between the two paradigms, and show that methods such as KL-control developed for RM can also be construed as belonging to DM. We further observe that while DM differs from RM, it can suffer from similar training difficulties, such as high gradient variance. We leverage connections between the two paradigms to import the concept of *baseline* into DM methods. We empirically validate the benefits of adding a baseline on an array of controllable language generation tasks such as constraining topic, sentiment, and gender distributions in texts sampled from a language model. We observe superior performance in terms of constraint satisfaction, stability and sample efficiency.

1 Introduction

Pre-trained language models (Devlin et al., 2019; Radford et al., 2019) are changing the landscape of Machine Learning research and practice. Due to their strong generative capabilities many studies have found it sufficient to "nudge" these models to conform to global preferences defined over the generated sequences instead of training from scratch using annotated data. These preferences could include topic and sentiment (Dathathri et al., 2020), valid musical notes and molecular structures (Jaques et al., 2017a), code compilability (Korbak et al., 2021), reducing distributional biases (Khalifa et al., 2021; Weidinger et al., 2021), evaluation metrics for Machine Translation and Summarization (Ranzato et al., 2016; Bahdanau et al., 2016), or direct human feedback (Ziegler et al., 2019; Stiennon et al., 2020). This large body of studies is driven by two different paradigms: *Reward Maximization* (RM) and *Distribution Matching* (DM).

[*] Work partly done during an internship at Naver Labs Europe.

[†] Independent Researcher. Work done at Naver Labs Europe.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

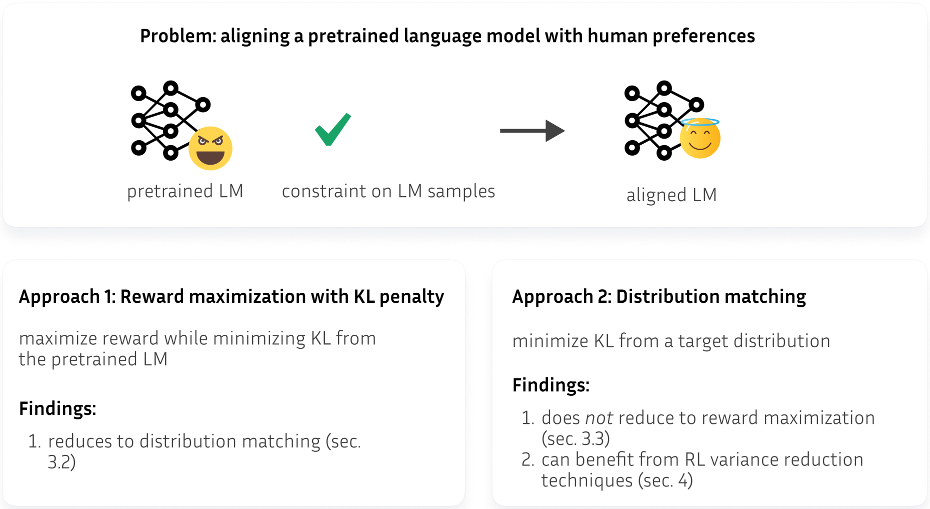

Problem: aligning a pretrained language model with human preferences
pretrained LM
constraint on LM samples
aligned LM
Approach 1: Reward maximization with KL penalty
maximize reward while minimizing KL from the pretrained LM
Findings:
1. reduces to distribution matching (sec. 3.2)
Approach 2: Distribution matching
minimize KL from a target distribution
Findings:
1. does *not* reduce to reward maximization (sec. 3.3)
2. can benefit from RL variance reduction techniques (sec. 4)

Figure 1: In this study we make a connection between two popular paradigms for aligning language models to human preferences: Reward maximization (RM) and Distribution matching (DM).

Reward Maximization RM intuitively nudges pre-trained models towards certain preferences by providing global sequence-level rewards when the model generates outputs that satisfy desired features. For instance, if the model is producing toxic content, we can apply Reinforcement Learning (RL) techniques to discourage it from producing similar content. However, naively applying RL yields a model that can undergo *catastrophic forgetting* of its original distribution. For example, it can degenerate into producing a single nonsensical but at least nontoxic sequence. Although several studies have considered hand-crafting general rewards to ensure desirable features like fluency (Liu et al., 2016a; Tambwekar et al., 2019), coming up with complete or perfect rewards is highly non-trivial (Wu et al., 2016; Vedantam et al., 2015). This has sparked a wide discussion on the overall effectiveness of RM for some tasks such as machine translation (Choshen et al., 2020; Kiegeland & Kreutzer, 2021).

Reward Maximization with KL-Control To tackle the aforementioned issues of "catastrophic forgetting", several studies, still under an RM paradigm, have considered incorporating a distributional term inside the reward to be maximized. In particular Jaques et al. (2017b, 2019) and Ziegler et al. (2019) or more recently Stiennon et al. (2020), Ouyang et al. (2022), Bai et al. (2022) and Perez et al. (2022) have applied variations of KL-control (Todorov, 2007; Kappen et al., 2012) which adds a penalty term to the reward term so that the resulting policy does not deviate too much from the original one in terms of KL-divergence. The overall objective with the KL-penalty is maximized using an RL algorithm of choice including: PPO (Schulman et al., 2017a) as in Ziegler et al. (2019) or Bai et al. (2022) or Q-learning (Mnih et al., 2013) as in Jaques et al. (2017b). Adding this *distributional* KL-penalty to the reward raises some important questions: What effect does it have on the shape of the optimal policy? Does this new objective have any interpretation from a distributional perspective?

Distribution Matching A different recent paradigm for fine-tuning language models to satisfy downstream preferences formulates the problem as Distribution Matching (DM). This paradigm consists of two steps: first a target distribution incorporating the desired preferences is defined as an Energy-Based Model (LeCun et al., 2006). Then the forward KL divergence is minimized between this target distribution and an auto-regressive policy using a family of algorithms referred to as Distributional Policy Gradients (DPG) (Parshakova et al., 2019b; Khalifa et al., 2021; Korbak et al., 2021, 2022a). This approach capitalizes on the flexibility of EBMs in specifying the target distribution. For example, the EBM can be defined so that it conforms to all downstream preferences while its corresponding normalized distribution has a minimal KL divergence from the original, pre-trained language model, therefore tackling the problem of "catastrophic forgetting" (Khalifa et al., 2021). Interestingly, this DM paradigm can also deal with *distributional* preferences, for instance, for de-biasing language models by specifying that the generated sequences should be gender-balanced, i.e. that 50% of generations contain female mentions. Such distributional constraints cannot be defined in the RM paradigm where a reward is calculated for a single sequence.

We can notice the promises and limitations of these two paradigms for fine-tuning language models. RM approaches are equipped with an arsenal of RL algorithms and optimization techniques that can be efficient in reward maximization, however they lack the distributional aspect to avoid catastrophic forgetting and impose distributional preferences over LMs. DM approaches are suited to tackle those limitations, however, the family of DPG algorithms currently used is not as rich as its RL counterpart.

While the connections between these two seemingly distinct paradigms have been noted (Parshakova et al., 2019b; Korbak et al., 2022b), they have not been explored in detail. Clarifying such connections might help import ideas from one approach to the other. This is our goal in this paper, detailing the nuanced connections and applying them to a case-study in variance reduction. Overall, our contributions are the following:

- We clarify relations between the RM and DM paradigms through a detailed comparison between the family of DPG algorithms and Policy Gradients (Table 1), stressing the differences between *parametric* and *non-parametric* rewards that are important in this regard.
- We introduce an interpretation of KL-control techniques from a distribution matching perspective, placing such techniques at an intermediate place between RM and DM (Theorem 1).
- We show how these connections can enable cross-pollination between the two perspectives by applying *baselines* — a variance reduction technique from RL — to DPG and derive a particular choice of a baseline (Facts 1 and 2). On an array of controllable language generation experiments, we show that adding baselines leads to superior performance on constraint satisfaction (Figure 3), stability on small batch sizes, and sample efficiency (Figure 4).

2 Background

Standard Policy Gradients One popular method for adapting the behaviour of language models to certain preferences has been that of assigning a "reward" score $R(x)$ for sequences x sampled from an autoregressive language model (policy) π_θ. Then, the simplest policy gradient algorithm in reinforcement learning, namely, REINFORCE (Williams, 1992a), aims to find the policy $\pi_\theta(x)$ that maximizes the average reward $\mathbb{E}_{x\sim\pi_\theta} R(x)$, and this leads, via the so-called "log derivative trick", to a gradient ascent algorithm that iteratively samples x from π_θ and update parameters by increments proportional to $R(x)\nabla_\theta \log \pi_\theta(x)$ via the following identity:

$$\nabla_\theta \mathbb{E}_{x\sim\pi_\theta} R(x) = \mathbb{E}_{x\sim\pi_\theta} R(x) \nabla_\theta \log \pi_\theta(x). \tag{1}$$

KL-control (Todorov, 2007; Kappen et al., 2012), was leveraged by Jaques et al. (2017b, 2019) and Ziegler et al. (2019) to include a KL penalty term in the reward function to penalize large deviations from the original pretrained model $a(x)$, weighted by a free hyperparameter β to control the trade-off between the two goals. That is, they maximize the expectation $\mathbb{E}_{x\sim\pi_\theta} R^z_\theta(x)$, where:

$$R^z_\theta(x) \doteq r(x) - \beta \log \frac{\pi_\theta(x)}{a(x)}. \tag{2}$$

Distributional Policy Gradients (DPG) (Parshakova et al., 2019b) is a recent approach used to fit an autoregressive policy π_θ to the distribution $p(x) = P(x)/Z$ induced by the EBM $P(x)$, where $Z = \sum_x P(x)$ is the normalization constant (partition function). Given an arbitrary EBM $P(x)$, DPG optimizes the loss function $D_{\mathrm{KL}}(p, \pi_\theta)$ with respect to the parameters θ of an autoregressive model π_θ, a loss which is minimized for $\pi_\theta = p$. The KL-divergence minimization objective leads to a gradient estimate of the form:

$$\begin{aligned}
\nabla_\theta D_{\mathrm{KL}}(p, \pi_\theta) &= -\, \nabla_\theta \mathbb{E}_{x\sim p} \log \pi_\theta(x) && (3)\\
&= -\sum_x p(x) \nabla_\theta \log \pi_\theta(x) = -\frac{1}{Z}\sum_x P(x)\nabla_\theta \log \pi_\theta(x) && (4)\\
&= -\frac{1}{Z}\;\mathbb{E}_{x\sim\pi_\theta} \frac{P(x)}{\pi_\theta(x)} \nabla_\theta \log \pi_\theta(x). && (5)
\end{aligned}$$

3 Reward Maximization vs Distribution Matching

In the previous section, we have summarized three approaches that have been suggested for fine-tuning language models. Two of them can be characterized as "Reward Maximization" (RM): Standard Policy Gradients (PG) and KL-control. On the other hand, DPG clearly belongs to the realm of "Distribution Matching" (DM) as it first defines the target distribution and then optimizes a policy to match it. In the rest of this section, we will explore connections between these two seemingly distinct concepts and, in the following section, we will exploit them to improve DM-based methods.

3.1 Standard vs. Parametric Rewards

Let us start with distinguishing between a "parametric reward" R_θ which depends on θ and a standard reward R, which does not. If we wished to maximize the expected parametric reward, $\mathbb{E}_{\pi_\theta} R_\theta(x)$, we would follow its gradient, leading to the identities:

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{x\sim\pi_\theta} R_\theta(x) &= \nabla_\theta \sum_x \pi_\theta(x) R_\theta(x) = \sum_x \pi_\theta(x) \nabla_\theta R_\theta(x) + \sum_x R_\theta(x) \nabla_\theta \pi_\theta(x) && (6)\\
&= \sum_x \pi_\theta(x)\nabla_\theta R_\theta(x) + \sum_x \pi_\theta(x) R_\theta(x) \nabla_\theta \log \pi_\theta(x) && (7)\\
&= \underbrace{\mathbb{E}_{x\sim\pi_\theta} \nabla_\theta R_\theta(x)}_{\text{RG-term}} + \underbrace{\mathbb{E}_{x\sim\pi_\theta} R_\theta(x) \nabla_\theta \log \pi_\theta(x)}_{\text{PG-term}}. && (8)
\end{aligned}$$

Equation (8) is the sum of two terms: the first one, the "RG-term" (Reward Gradient term), involves the gradient of the reward. The second one, the "PG-term" (Policy Gradient term), was obtained using the "log derivative trick" and involves the gradient of the policy *stricto sensu*. In standard RL, where the reward does *not* depend on θ, the RG-term disappears and the gradient of expected reward consists solely of the PG-term. However, when R_θ depends on θ, the gradients are distinct (apart from specific cases where the RG-term evaluates to 0, as we will see below).

3.2 KL-control as Distribution Matching

Adding a KL-penalty term to the reward (as in the case of KL-control) leads to a parametric reward. However, due to the particular form of its objective, the RG-term actually *vanishes*,[3] leaving only the PG-term $\mathbb{E}_{x\sim\pi_\theta} R^z_\theta(x) \nabla_\theta \log \pi_\theta(x)$ and simplifying the tuning procedure to a standard Policy Gradient. While this algorithm falls under the RM paradigm, here we argue that is its nature is multifaceted, and explore deeper connections with the DM paradigm. More precisely, the maximization of reward with the KL penalty term is equivalent to a distributional matching with an underlying emergent sequential EBM, a remark that already reveals some similarities with DPG.[4]

Theorem 1. *Consider the following EBM:*

$$P_z(x) = a(x) e^{r(x)/\beta} \tag{9}$$

and let p_z *be the normalized distribution* $p_z(x) = \frac{1}{Z} P_z(x)$*, with* $Z = \sum_x P_z(x)$*. Then:*

(i) $\arg\max_{\pi_\theta} \mathbb{E}_{x\sim\pi_\theta} R^z_\theta(x) = \arg\min_{\pi_\theta} D_{\mathrm{KL}}(\pi_\theta, p_z)$*;*

(ii) $\arg\max_{\pi\in\mathcal{D}(X)} \mathbb{E}_{x\sim\pi} R^z_\pi(x) = p_z$*, where* $\mathcal{D}(X)$ *is the family of all distributions over* X*, and* $R^z_\pi(x) \doteq r(x) - \beta \log \frac{\pi(x)}{a(x)}$.

Proof. A simple way to prove this is to notice that the expectation of the reward R^z_θ has a monotonically decreasing relationship with the *reverse* KL divergence between π_θ and p_z:

$$D_{\mathrm{KL}}(\pi_\theta, p_z) = \mathbb{E}_{x\sim\pi_\theta} \log \frac{\pi_\theta(x)}{p_z(x)} = \mathbb{E}_{x\sim\pi_\theta}\Big[\log \pi_\theta(x) - \log \frac{1}{Z} a(x) e^{r(x)/\beta}\Big]$$

[3] This is because $\mathbb{E}_{\pi_\theta} \nabla_\theta R^z_\theta(x) = -\beta\, \mathbb{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(x) = 0$, via the identity $\mathbb{E}_{\pi_\theta} \nabla_\theta \log \pi_\theta(x) = \sum_x \pi_\theta(x) \nabla_\theta \log \pi_\theta(x) = \sum_x \nabla_\theta \pi_\theta(x) = \nabla_\theta \sum_x \pi_\theta(x) = 0$.

[4] The optimal policy p_z is briefly mentioned in (Ziegler et al., 2019) without reference or derivation. The proof, which reveals a connection to the reverse KL divergence from π_θ, is ours.

	Policy Gradients	DPG
Reward	$R(x)$	$R_\theta(x) = \frac{P(x)}{\pi_\theta(x)}$
∇_θ	$\mathbb{E}_{x\sim\pi_\theta} R(x)\nabla_\theta \log \pi_\theta(x)$	$\mathbb{E}_{x\sim\pi_\theta} \frac{P(x)}{\pi_\theta(x)}\nabla_\theta \log \pi_\theta(x)$
Baseline	$\mathbb{E}_{x\sim\pi_\theta} R(x)$	Z
∇_θ **with Baseline**	$\mathbb{E}_{x\sim\pi_\theta}\Big[R(x) - \mathbb{E}_{x\sim\pi_\theta} R(x)\Big]\nabla_\theta \log \pi_\theta(x)$	$\mathbb{E}_{x\sim\pi_\theta}\Big[\frac{P(x)}{\pi_\theta(x)} - Z\Big]\nabla_\theta \log \pi_\theta(x)$

Table 1: A comparison between Policy Gradients (Sutton et al., 1999) and Distributional Policy Gradients (Parshakova et al., 2019b) forms of Reward, Baseline, and Gradient of the loss function (the PG-term) before (∇_θ) and after (∇_θ with Baseline) including a baseline for variance reduction .

$$= \log Z - \frac{1}{\beta}\,\mathbb{E}_{x\sim\pi_\theta}\Big[r(x) - \beta \log \frac{\pi_\theta(x)}{a(x)}\Big] = \log Z - \frac{1}{\beta}\,\mathbb{E}_{x\sim\pi_\theta} R^z_\theta(x), \tag{10}$$

so that the $\arg\min_{\pi_\theta} D_{\mathrm{KL}}(\pi_\theta, p_z)$ coincides with the $\arg\max_{\pi_\theta} \mathbb{E}_{x\sim\pi_\theta} R^z_\theta(x)$, proving (i). On the other hand, $\arg\min_{\pi\in\mathcal{D}(X)} D_{\mathrm{KL}}(\pi, p_z)$, which also corresponds to $\arg\max_{\pi\in\mathcal{D}(X)} \mathbb{E}_{x\sim\pi} R^z_\pi$ because of (i) applied to a family $\pi_{\theta'}$ covering $\mathcal{D}(X)$ in full, is just p_z, concluding the proof. □

Overall, we can conclude that the addition of the distributional term (KL-penalty) to the reward does indeed provide a DM interpretation, namely in terms of minimizing the reverse KL divergence with an emergent underlying distribution $p_z(x)$. We note that $p_z(x)$ does not correspond to a free and explicit choice of EBM (e.g. one that balances the gender and topic distributions of a language model). Instead equation (9) appears in a restrictive format, which is implicitly defined by the reward R^z_θ, along with a β hyperparameter without a clear meaning. By contrast, the DPG algorithms are designed to perform DM on any EBM specification, corresponding to an explicit distributional objective.

3.3 Similarities and Differences between DPG and Policy Gradients

In the previous subsection, we have connected KL-control, a method designed under a RM paradigm, to DM. Now, we turn to the converse question of whether DPG, a DM method, can be connected to RM. We begin by noting that after defining $R_\theta = \frac{P(x)}{\pi_\theta(x)}$, the DPG gradient $\mathbb{E}_{x\sim\pi_\theta} \frac{P(x)}{\pi_\theta(x)}\nabla_\theta \log \pi_\theta(x)$ acquires the format of the PG-term $\mathbb{E}_{\pi_\theta} R_\theta \nabla_\theta \log \pi_\theta(x)$.

However, the DM objective of DPG *cannot* be considered as maximizing the average "reward" $R_\theta(x) = \frac{P(x)}{\pi_\theta(x)}$, as this would require adding also the RG-term $\mathbb{E}_{\pi_\theta} \nabla_\theta \frac{P(x)}{\pi_\theta(x)}$ into the gradient, which in this case does not vanish.

Nonetheless, the analogy behind this gradient term is more fruitful than it first appears. As a matter of fact, DPG gradient estimates suffer from the same high-variance problems as with standard PG. While the objective of DPG (distribution matching) is different from that of Policy Gradients (reward maximization), DPG also needs to estimate the PG-term $\mathbb{E}_{\pi_\theta} R_\theta(x)\nabla_\theta \log \pi_\theta(x)$ at a *given* value of θ, using a batch of samples x. For such a *fixed* θ, we can define provisionally set $R(x) \doteq R_\theta$ and the problem of gradient estimation *for this fixed* θ is identical to the estimation $\mathbb{E}_{x\sim\pi_\theta} R(x)\nabla_\theta \log \pi_\theta(x)$ based on a set of samples x in standard RL. Therefore, techniques that have been developed to reduce the variance of the gradients estimates in RL can be ported to DPG insofar as we are computing the gradient estimates *at a given* θ. In Section 4, we show how one can import one such variance reduction technique to the DPG: baselines.

4 A Case Study on Variance Reduction

Baselines are a standard variance reduction technique in the context of Policy Gradients (Sutton & Barto, 2018). The idea is to subtract from the reward $R(x)$ a value B that does not introduce bias to the gradients but may change variance. After the introduction of baseline, equation (1) then takes the following form:

$$\nabla_\theta \mathbb{E}_{\pi_\theta} R(x) = \mathbb{E}_{\pi_\theta}\left[R(x) - B\right]\nabla_\theta \log \pi_\theta(x). \tag{11}$$

In standard RL, the simplest form of baseline B is just the average of the rewards for the policy:[5]

$$B^{\text{RL}} = \mathbb{E}_{x\sim\pi_\theta} R(x). \tag{12}$$

Following the same methodology of taking the baseline to be the expectation of the reward term, we can obtain a remarkably simple form of a baseline for DPG: [6]

$$B = \mathbb{E}_{x\sim\pi_\theta} \frac{P(x)}{\pi_\theta(x)} = \sum_x \pi_\theta(x) \frac{P(x)}{\pi_\theta(x)} = \sum_x P(x) = Z. \tag{13}$$

Fact 1. *Subtracting B from $R_\theta(x)$ does not introduce bias into DPG gradient estimates.*

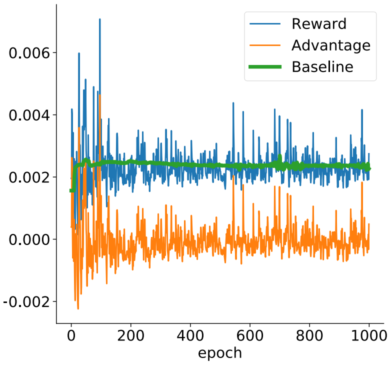


Figure 2: Values of reward, advantage and the baseline for first 1000 epochs of a pointwise constraint experiment.

Proof. Let us rewrite the DPG gradient in (5) with the added baseline $B = Z$:

$$\begin{aligned}\mathbb{E}_{x\sim\pi_\theta}\Big[R_\theta(x) - Z\Big]\nabla_\theta \log \pi_\theta(x) &= \mathbb{E}_{x\sim\pi_\theta} R_\theta(x)\nabla_\theta \log \pi_\theta(x) - Z\,\mathbb{E}_{x\sim\pi_\theta} \nabla_\theta \log \pi_\theta(x)\\ &= \mathbb{E}_{x\sim\pi_\theta} R_\theta(x)\nabla_\theta \log \pi_\theta(x) - Z\Big[\sum_x \nabla_\theta \pi_\theta(x)\Big]\end{aligned} \tag{14}$$

Here, the second term does not introduce bias because $Z\Big[\sum_x \nabla_\theta \pi_\theta(x)\Big] = 0$, leaving us with the exact same form of gradient as in the original DPG algorithm. □

Note that since B^{RL} depends on θ, it has to be be re-estimated after each gradient update. On the other hand, B does *not* depend on θ, which is an advantage because B could be now estimated by averaging over samples from *all* the different θ's without introducing bias, leading to a more accurate estimation. See Table 1 for a comparison of these two forms of baselines.

Algorithm 1 KL-Adaptive DPG with baseline

```
Require: $P$, initial generative model $a$
$\pi_\theta \leftarrow a$, $q \leftarrow a$
for each iteration do
    for each episode do
        sample $x$ from $q(\cdot)$
        $\theta \leftarrow \theta + \alpha^{(\theta)} \Big[\frac{P(x)}{q(x)} - Z\frac{\pi_\theta(x)}{q(x)}\Big] \nabla_\theta \log \pi_\theta(x)$
    if $D_{\text{KL}}(p||\pi_\theta) < D_{\text{KL}}(p||q)$ then
        $q \leftarrow \pi_\theta$
Ensure: $\pi_\theta$
```

The off-policy DPG version introduced in (Parshakova et al., 2019b) and its KL-adaptive variant (Khalifa et al., 2021) sample a proposal distribution q instead of the policy π_θ. Then, the baseline takes the form

$$B^{\text{off}}(x) = Z\frac{\pi_\theta(x)}{q(x)}, \tag{15}$$

where the $\frac{\pi_\theta(x)}{q(x)}$ term is an importance weight correcting for the bias introduced by sampling from q. Similarly to the DPG case, we can prove the following (see Appendix C):

Fact 2. *Subtracting $B^{\text{off}}(x)$ from $R_\theta(x)$ does not bias the off-policy DPG gradient estimates.*

In practice, as shown on Figure 2, adding a baseline to KL-adaptive DPG (Algorithm 1) centers the advantage (defined as $A \doteq \frac{P(x)}{q(x)} - Z\frac{\pi_\theta(x)}{q(x)}$) around 0 leading to better performance on: convergence (section 4.3), stability on small batch sizes (section 4.4), and variance reduction (section 4.5).

[5] While this baseline is not optimal (proof Appendix C.1), it is widely used in practice.

[6] In the scope of this paper, our focus is on importing to DPG simple constant baselines. The advantage is that this is a technique that is not impacted by the fact that R_θ depends on θ: it can be applied "θ-locally" to provide a more accurate estimate of $\mathbb{E}_{x\sim\pi_\theta} R_\theta(x)\nabla_\theta \log \pi_\theta(x)$ for a *fixed* θ, irrespective of the values of $R_{\theta'}$ elsewhere, while variance reduction techniques that involve several $\theta's$ simultaneously raise additional challenges for parametric rewards.

4.1 Generation with Distributional Control

We investigate the benefits of adding a baseline to the DPG algorithm, on the Generation with Distributional Control (GDC) (Khalifa et al., 2021) framework. GDC makes use of DPG to control the properties of pre-trained language models to satisfy certain constraints. In our experiments, follow target distribution form of Parshakova et al. (2019a), Khalifa et al. (2021) and Korbak et al. (2022a), in which the EBM $P(x)$ is defined so that its normalized variant $p(x)$ matches a set of desired moments constraints on given features $\phi_i(x)$, while having a minimal KL divergence $D_{\mathrm{KL}}(p, a)$ from an original pretrained language model a, to avoid catastrophic forgetting.

These constraints are expressed as conditions $\bar{\mu}_i = \mathbb{E}_{x\sim p}\phi_i(x)$, for $i \in \{1, \dots, n\}$, by which the moments (expectations) under the distribution p of each feature $\phi_i(x)$ are required to take certain desired values $\bar{\mu}_i$. For instance, let $\phi_1(x) = 1$ iff the topic of x is science and $\phi_2(x) = 1$ iff x mentions a female person, then imposing moments $\bar{\mu}_1 = 1$ and $\bar{\mu}_2 = 0.5$ constrains the language model p to only generate sequences about science, half of which mention females. $P(x)$ is uniquely determined by the following form:[7]

$$P(x) = a(x)e^{\sum_{i=1}^{n} \lambda_i \phi_i(x)}, \tag{16}$$

where λ_i terms control the moments μ_i of the associated features, which can be estimated through self-normalized importance sampling (Owen, 2013); and then, to make the moments match the desired values, the λ_i terms can be optimized through SGD (Parshakova et al., 2019a).

4.2 Experimental setup

We evaluate our method on an array of 10 controlled text generation tasks. For each, given a pre-trained language model $a(x)$, and a set of constraints, the objective of each fine-tuning method is to obtain a fine-tuned language model π_θ that satisfies the imposed constraints while deviating as minimally as possible from the original language model $a(x)$.

Constraints are defined as a set of binary features $\{\phi_i\}$ and their corresponding desired percentages (moments) $\{\bar{\mu}_i\}$ within the generations of the target language model. Based on the value of the moment constraints these 10 tasks are divided into 6 tasks of pointwise constraints (for which $\bar{\mu}_i = 1$), 2 tasks of distributional constraints ($0 < \bar{\mu}_i < 1$) and 2 tasks of mixed type constraints (hybrid):

(a) Single-word constraints, where $\phi(x) = 1$ iff the a given word appears in the sequence x. We experiment with frequent words (task 1: “amazing”, original frequency: 10^{-4}) and (task 2: “WikiLeaks”, original frequency: 10^{-5}) rare words,

(b) Wordlist constraints, where $\phi(x) = 1$ iff x contains at least one word from a given list. We consider lists of word associated with politics (task 3) and science (task 4) published by Dathathri et al. (2020),

(c) Sentiment classifier constraints, where $\phi(x) = 1$ if x is classified as positive (task 5), or negative (task 6) by a pre-trained classifier published by Dathathri et al. (2020).

(d) A single distributional constraint where $\phi(x) = 1$ iff x contains a female figure mention, and $\bar{\mu} = 0.5$ (task 8),

(e) A set of four distributional constraints: $\phi_i(x) = 1$ iff x contains at least one of the words in the “science", “art", “sports" and “business" wordlists (compiled by Dathathri et al. (2020)), respectively. For each i, $\bar{\mu}_i = 0.25$ (task 8),

(f) Hybrid constraints where $\phi_1(x) = 1$ iff x contains more female than male pronouns, $\bar{\mu}_1 = 0.5$ and $\phi_2(x) = 1$ iff x contains at least one of the words from the “sports" wordlist (task 9) or “politics” wordlist, $\bar{\mu}_2(x) = 1$ (task 10).

Methods We modify the GDC framework Khalifa et al. (2021), namely its KL-DPG algorithm, to include a baseline as shown in Algorithm 1. We refer to this method as **GDC++**. In addition to comparing **GDC++** with **GDC** we compare with two reward maximization baselines: **Reinforce** (Williams, 1992b) and **Ziegler** (Ziegler et al., 2019). Reinforce tries to maximize the expected reward $\mathbb{E}_{x\sim\pi_\theta}R(x)$, where $R(x) = 1$ if and only if the pointwise constraints are met. Ziegler instantiates the KL-control approach: its objective includes a KL penalty term for departures from a. Following (Khalifa et al., 2021), for hybrid and distributional constraints (tasks 8-10) we compare

[7] For a more precise formulation of this EBM, see (Khalifa et al., 2021).

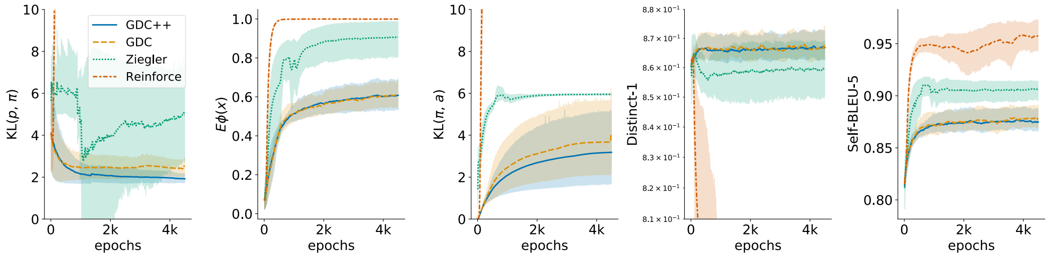


Figure 3: Evaluation metrics: $D_{\mathrm{KL}}(p, \pi_\theta)$ (↓ better), $\mathbb{E}_{\pi_\theta}\phi(x)$ (↑ better), $D_{\mathrm{KL}}(\pi_\theta, a)$ (↓ better), Self-BLEU-5 (↓ better), and Distinct-1 (↑ better) aggregated over 6 pointwise constraints experiments (tasks 1-6) for policies obtained from GDC++, GDC, Ziegler and Reinforce. See Figure 6 for aggregated distributional constraints experiments. In the Appendix Figures 7-10 and Table 4 contain individual view and final results of each run.

only GDC and GDC++ because the RM objective of Ziegler and Reinforce is not equipped to handle them.

Metrics We report the following metrics at each validation step over batches of samples from π_θ:

1. $\mathbb{E}_{x\sim\pi_\theta}\phi_i(x)$, measuring the ability to reach the target moment of the i-th feature.
2. $D_{\mathrm{KL}}(p, \pi_\theta)$, the forward KL divergence from the optimal target distribution p.[8]
3. $D_{\mathrm{KL}}(\pi_\theta, a)$, the reverse KL divergence from the original pretrained language model a.
4. Distinct-n score, a measure of text diversity in terms of the frequency of repetitions within a single sample x, proposed by (Li et al., 2016a).
5. Self-BLEU-n, a measure of text diversity on a distributional level *across* samples proposed by (Zhu et al., 2018), ensuring that policies don't converge into limited number of sequences that satisfy the imposed constraints Caccia et al. (2020).

Training details For tasks 1-6, we use a pre-trained GPT-2 small with 117M parameters (Radford et al., 2019) as the original language model a. For tasks 7-10, a is the same pre-trained model additionally fine-tuned on the WikiBio (Lebret et al., 2016) dataset. See Appendix E for more details. The code for all the experiments presented in the paper will be available at github.com/naver/gdc.

4.3 Results

We present the evolution of our metrics through training epochs in Figure 3 (aggregated over tasks 1-6) and Figure 6 in the Appendix (aggregated over tasks 7-10). Results for each task are presented separately on Figures 7-10 in the Appendix.

Consistent with prior work (Khalifa et al., 2021; Korbak et al., 2022a), we observe that Reinforce is able to quickly achieve high levels of constraint satisfaction, but at the cost of large deviations from a, which translates into significantly decreased diversity of generated samples (in terms of Self-BLEU-5 and Distinct-1). The KL penalty term in Ziegler imposes an upper bound on deviation from a but the deviation is still significant enough to result in a drop in diversity. Moreover, we have observed Ziegler's objective to result in very unstable training.

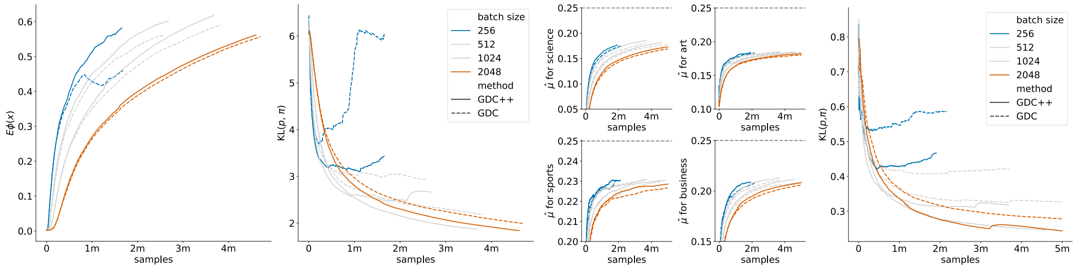
GDC and GDC++ are the only fine-tuning methods that address constraint satisfaction based on a clear formal objective, i.e. reducing the divergence from p. The approach translates into significantly smaller deviations from a and maintaining diversity within and across samples. The addition of a baseline indeed reduces the variance. We analyze that extensively in Appendix 4.5 while here focusing on the downstream effects of variance reduction. One is that π_θ is now able to compound staying closer to p and a *at the same time*, while achieving slightly better constraint satisfaction. We have also observed that baseline stabilizes training, leading to smoother curves.[9]

4.4 The effect of baseline across batch sizes

We expect that reducing gradient estimates variance can allow to train with lower batch sizes, performing gradient updates on estimates based on smaller batch sizes can increase the sample

[8] See Appendix D for a detailed description of how $D_{\mathrm{KL}}(p, \pi_\theta)$ is computed.

[9] The interested reader can compare the large fluctuations of the Ziegler objective to more stable training curves of GDC , and even more of GDC++ , in the disaggregated curves in Figures 7-10 of the Appendix.



(a) Task 1: a pointwise constraint

(b) Task 8: a set of distributional constraints; $\bar{\mu}_i = 0.25$

Figure 4: $\mathbb{E}_{\pi_\theta}\phi(x)$ or $\hat{\mu}$ per constraint (↑ better) and $D_{\mathrm{KL}}(p, \pi_\theta)$ (↓ better) as a function of the number of samples reported for task 1 (a) and task 8 (b). We report the number of samples (i.e. the number of epochs times the batch size) for a fair comparison of convergence speed. *GDC++ is consistently superior across all batch sizes in terms of convergence and constraint satisfaction*. The effect is more conspicuous with small batch sizes. Batch sizes 512 and 2014 are greyed out for clarity.

efficiency. To test this, we rerun tasks 1 (a pointwise constraint on the word "amazing") and 8 (distributional constraints on topics) with four batch sizes (256, 512, 1024, 2048). Figures 4a and 4b show the benefits of adding a baseline — higher constraint satisfaction, lower divergence from p, more stable training — and is especially evident with lower batch sizes. For instance, with batch size 256, GDC++ obtains a significantly higher constraint satisfaction rate and lower divergence from p.

Furthermore, stable training with smaller batch sizes translates into better sample efficiency. For instance, in task 1 (Figure 4a), GDC++ with batch size 256 needs 1M samples to achieve $\mathbb{E}_{x\sim\pi_\theta}\phi(x) = 0.5$ while GDC++ with batch size 2048 needs 4M. In contrast, GDC with batch size 256 does not achieve $\mathbb{E}_{x\sim\pi_\theta}\phi(x) = 0.5$ at all, confirming the importance of adding the baseline.

4.5 Empirical Evaluation of Variance Reduction

Next, we evaluate empirically the effect of the baseline for variance reduction. We select two tasks: task 1 (a pointwise constraint) and task 7 (distributional constraints) described in Section 4.2, each with 3 different seeds, while monitoring the following variance measures:

Gradient Variance The gradient estimate is defined as: $G_\theta(x) \doteq A(x)\nabla_\theta \log \pi_\theta(x)$, where $G_\theta(x) \in \mathbb{R}^{|\theta|}$ is an unbiased estimate of the gradient of the forward KL loss $\nabla_\theta D_{\mathrm{KL}}(p, \pi_\theta)$ with respect to the parameters θ. We then have, with $\mu(G_\theta) \doteq \mathbb{E}_{x\sim q} G_\theta(x)$:

$$\mathrm{Var}(G_\theta) \doteq \mathbb{E}_{x\sim q} \left|\left|G_\theta(x) - \mu(G_\theta)\right|\right|_2^2 \tag{17}$$

$$= \mathbb{E}_{x\sim q}||G_\theta(x)||_2^2 - ||\mu(G_\theta)||_2^2. \tag{18}$$

Variance of the advantage is defined by:

$$\mathrm{Var}\left(\mathrm{A}\right) \doteq \mathbb{E}_{x\sim q} \left\|\mathrm{A}(x) - \mu^{\mathrm{A}}\right\|_2^2 \tag{19}$$

where, $\mu^{\mathrm{A}} \equiv \mathbb{E}_{x\sim q}\ \mathrm{A}(x)$ is the mean of the advantage, which we showed above to be null after the addition of the baseline.

Expected absolute value of the advantage This metric is defined as:

$$\mu^{|\mathrm{A}|} \doteq \mathbb{E}_{x\sim q}\ |\mathrm{A}(x)|. \tag{20}$$

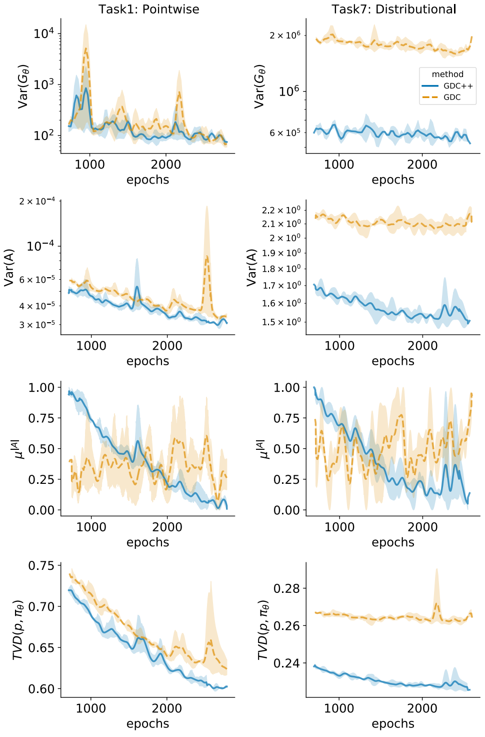


Figure 5: Comparison between GDC and GDC++ using a set of Variance diagnosis metrics on pointwise and distributional constraints experiments.

It directly provides a standard measure of distributional discrepancy between p and π_θ, in terms of TVD (Total Variation Distance). We have:

$$\mathbb{E}_{x\sim q}\left|\frac{p(x)}{q(x)} - \frac{\pi_\theta(x)}{q(x)}\right| = 2\,\mathrm{TVD}(p, \pi_\theta). \tag{21}$$

Results Figure 5 shows that GDC++ obtains lower variance in the gradient estimates $\mathrm{Var}(G_\theta)$ and the variance of the advantage $\mathrm{Var}\,(\mathrm{A})$ in both pointwise and distributional experiments compared to its non-baseline counterpart GDC. We further observe a decreasing trend in the mean absolute value of the advantage $\mu^{|\mathrm{A}|}$ which is correlated with a decreasing trend in the TVD distance between the trained policy π_θ and the optimal distribution p. Overall, these results shows that adding a baseline to DPG reduces the variance during training and yields better convergence towards the optimal distribution p.

5 Related work

The idea of posing control problems as distribution matching has resurfaced numerous times in the RL literature (Kappen et al., 2012; Friston et al., 2010; Levine, 2018; Hafner et al., 2020; Buckley et al., 2017). KL-control can be seen as a generalisation of maximum entropy RL (MaxEnt RL) (Haarnoja et al., 2017, 2018) to informed priors. If in (2) we chose $a(x)$ to be a uniform distribution (assuming right now finiteness of X) instead of a pretrained LM distribution, then the KL penalty $D_{\mathrm{KL}}(\pi_\theta, a)$ would reduce to an entropy bonus. Both KL-control and MaxEnt RL can be derived from a general framework of control-as-inference (Levine, 2018) which poses control as minimising KL from a certain target distribution. However, most practical algorithms in the MaxEnt RL family minimise KL from a target policy which changes throughout training; in contrast, DPG's target distribution p and KL-control implicit target distribution p_z are defined at trajectory level and fixed throughout training.

Perhaps the closest method to KL-control and DPG in the larger family of inference-based RL (Furuta et al., 2021) is AWR (Peng et al., 2019) which minimises the *forward* KL from an off-policy target distribution. Yet another approach with apparent similarity to KL-control and DPG is state marginal matching (SMM) (Hazan et al., 2018; Lee et al., 2019). SMM poses exploration as learning a policy that induces a state marginal distribution that matches a target state distribution. While SMM's target distribution is fixed, it is defined for individual states, while in the controllable language generation tasks we consider, the target distribution is defined over a complete trajectory considered as a unit. See Appendix B for an extended discussion of related work.

6 Conclusion

Fine-tuning large language models has become an active area of research, due to its importance in adapting large language models to satisfy task-level preferences, or in combating their social risks such as "distributional" stereotyping (Weidinger et al., 2021; Welbl et al., 2021). [10] In this paper, we analyzed in depth the nuanced relation between two popular fine-tuning paradigms: RM and DM. We demonstrated that KL-control can be seen as a form of DM and showed that while DPG and PG have different goals, some similarities (similar forms of gradient estimates despite different objectives) can be exploited. We used these insights to inform an extension of DPG, consisting in adding a baseline to reduce the variance of gradient estimates.

The connections we established suggest that despite fundamental differences between DPG and RL, some of the theoretical results and algorithmic techniques from RL can be adapted to a DM framework without losing their formal guarantees. In this paper, we focus on variance reduction using baselines, but the space of possible enhancements is vast. Promising candidates include further reducing the variance using a learned value function (Konda & Tsitsiklis, 2000) and preventing detrimentally large policy updates by maintaining a trust region in the policy space – akin to techniques such as TRPO (Schulman et al., 2015) and PPO (Schulman et al., 2017b). Another future direction could consist in analyzing the relation between explicit EBMs in DPG and implicit EBMs arising in KL-control and characterizing the space of EBMs that could be reached through KL-control.

[10] See Appendix A for a discussion of broader impacts of large language models and controllable language generation.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. Globally Normalized Transition-Based Neural Networks. 2016. doi: 10.18653/v1/P16-1231.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. An Actor-Critic Algorithm for Sequence Prediction. (2015):1–17, 2016. URL http://arxiv.org/abs/1607.07086.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=SJDaqqveg.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

A. Bakhtin, Y. Deng, S. Gross, Myle Ott, Marc'Aurelio Ranzato, and Arthur Szlam. Energy-based models for text. *ArXiv*, abs/2004.10188, 2020.

David Belanger and Andrew McCallum. Structured prediction energy networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 983–992. JMLR.org, 2016. URL http://dl.acm.org/citation.cfm?id=3045390.3045495.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5454–5476, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.485. URL https://www.aclweb.org/anthology/2020.acl-main.485.

Christopher L. Buckley, Chang Sub Kim, Simon McGregor, and Anil K. Seth. The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 81:55–79, 2017. ISSN 0022-2496. doi: https://doi.org/10.1016/j.jmp.2017.09.004. URL https://www.sciencedirect.com/science/article/pii/S0022249617300962.

Massimo Caccia, Lucas Caccia, William Fedus, Hugo Larochelle, Joelle Pineau, and Laurent Charlin. Language gans falling short. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJgza6VtPB.

Leshem Choshen, Lior Fox, Zohar Aizenbud, and Omri Abend. On the weaknesses of reinforcement learning for neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=H1eCw3EKvH.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. Pre-training transformers as energy-based cloze models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 285–294. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.20. URL https://doi.org/10.18653/v1/2020.emnlp-main.20.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=H1edEyBKDS`.

Peter Dayan. Reinforcement comparison. In *Proceedings of the 1990 Connectionist Models Summer School*, pp. 45–51. Morgan Kaufmann, San Mateo, CA, 1990.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL `https://openreview.net/forum?id=B1l4SgHKDH`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `https://www.aclweb.org/anthology/N19-1423`.

Karl J Friston, Jean Daunizeau, James Kilner, and Stefan J Kiebel. Action and behavior: a free-energy formulation. *Biological cybernetics*, 102(3):227–260, 2010.

Hiroki Furuta, Tadashi Kozuno, Tatsuya Matsushima, Yutaka Matsuo, and Shixiang Shane Gu. Co-adaptation of algorithmic and implementational innovations in inference-based deep reinforcement learning, 2021. URL `https://arxiv.org/abs/2103.17258`.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *J. Mach. Learn. Res.*, 5:1471–1530, December 2004. ISSN 1532-4435.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `http://proceedings.mlr.press/v9/gutmann10a.html`.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1352–1361. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/haarnoja17a.html`.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL `https://arxiv.org/abs/1801.01290`.

Danijar Hafner, Pedro A. Ortega, Jimmy Ba, Thomas Parr, Karl Friston, and Nicolas Heess. Action and perception as divergence minimization, 2020.

Elad Hazan, Sham M. Kakade, Karan Singh, and Abby Van Soest. Provably efficient maximum entropy exploration, 2018. URL `https://arxiv.org/abs/1812.02690`.

Tianxing He, Bryan McCann, Caiming Xiong, and Ehsan Hosseini-Asl. Joint energy-based model training for better calibrated natural language understanding models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1754–1761, Online, April 2021. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2021.eacl-main.151`.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Comput.*, 14(8):1771–1800, 2002. doi: 10.1162/089976602760128018. URL `https://doi.org/10.1162/089976602760128018`.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, José Miguel Hernández-Lobato, Richard E. Turner, and Douglas Eck. Sequence tutor: Conservative fine-tuning of sequence generation models with kl-control. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1645–1654. PMLR, 2017a. URL http://proceedings.mlr.press/v70/jaques17a.html.

Natasha Jaques, Shixiang Gu, Dzmitry Bahdanau, Jose Miguel Hernandez Lobato, Richard E. Turner, and Doug Eck. Tuning recurrent neural networks with reinforcement learning. 2017b. URL https://openreview.net/pdf?id=Syyv2e-Kx.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Àgata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind W. Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *CoRR*, abs/1907.00456, 2019. URL http://arxiv.org/abs/1907.00456.

Hilbert J Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *Machine learning*, 87(2):159–182, 2012.

Muhammad Khalifa, Hady Elsahar, and Marc Dymetman. A distributional approach to controlled text generation. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jWkw45-9AbL.

Samuel Kiegeland and Julia Kreutzer. Revisiting the weaknesses of reinforcement learning for neural machine translation. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 1673–1681. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.133. URL https://doi.org/10.18653/v1/2021.naacl-main.133.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

A.S. Klyubin, D. Polani, and C.L. Nehaniv. Empowerment: a universal agent-centric measure of control. In *2005 IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135 Vol.1, 2005. doi: 10.1109/CEC.2005.1554676.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. In S. Solla, T. Leen, and K. Müller (eds.), *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000. URL https://proceedings.neurips.cc/paper/1999/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

Tomasz Korbak, Hady Elsahar, Marc Dymetman, and Germán Kruszewski. Energy-based models for code generation under compilability constraints. *CoRR*, abs/2106.04985, 2021. URL https://arxiv.org/abs/2106.04985.

Tomasz Korbak, Hady Elsahar, German Kruszewski, and Marc Dymetman. Controlling conditional language models without catastrophic forgetting. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11499–11528. PMLR, 17–23 Jul 2022a. URL https://proceedings.mlr.press/v162/korbak22a.html.

Tomasz Korbak, Ethan Perez, and Christopher L Buckley. RL with KL penalties is better viewed as Bayesian inference, 2022b. URL https://arxiv.org/abs/2205.11275.

Rémi Lebret, David Grangier, and Michael Auli. Neural text generation from structured data with application to the biography domain. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 1203–1213. The Association for Computational Linguistics, 2016. doi: 10.18653/v1/d16-1128. URL https://doi.org/10.18653/v1/d16-1128.

Yann LeCun, Sumit Chopra, Raia Hadsell, Marc'Aurelio Ranzato, and Fu Jie Huang. A Tutorial on Energy-Based Learning. In *Predicting Structured Data*. MIT Press, 2006.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. 2019.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review, 2018.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 110–119, San Diego, California, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1014. URL https://www.aclweb.org/anthology/N16-1014.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. Deep reinforcement learning for dialogue generation. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 1192–1202. The Association for Computational Linguistics, 2016b. doi: 10.18653/v1/d16-1127. URL https://doi.org/10.18653/v1/d16-1127.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Towards understanding and mitigating social biases in language models, 2021.

Chia-Wei Liu, Ryan Lowe, Iulian Serban, Michael Noseworthy, Laurent Charlin, and Joelle Pineau. How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation. In Jian Su, Xavier Carreras, and Kevin Duh (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pp. 2122–2132. The Association for Computational Linguistics, 2016a. doi: 10.18653/v1/d16-1230. URL https://doi.org/10.18653/v1/d16-1230.

Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. Optimization of image description metrics using policy gradient methods. *CoRR*, abs/1612.00370, 2016b. URL http://arxiv.org/abs/1612.00370.

Beren Millidge, Alexander Tschantz, Anil Seth, and Christopher Buckley. Understanding the origin of information-seeking exploration in probabilistic objectives for control, 2021.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1791–1799, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/mnih14.html.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin A. Riedmiller. Playing atari with deep reinforcement learning. *CoRR*, abs/1312.5602, 2013. URL http://arxiv.org/abs/1312.5602.

Subhajit Naskar, Pedram Rooshenas, Simeng Sun, Mohit Iyyer, and A. McCallum. Energy-based reranking: Improving neural machine translation using energy-based models. *ArXiv*, abs/2009.13267, 2020.

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

Art B. Owen. Importance Sampling. In *Monte Carlo theory, methods and examples*, chapter 9. 2013. URL https://statweb.stanford.edu/~owen/mc/Ch-var-is.pdf.

Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. Global Autoregressive Models for Data-Efficient Sequence Learning. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pp. 900–909, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/K19-1084. URL https://www.aclweb.org/anthology/K19-1084.

Tetiana Parshakova, Jean-Marc Andreoli, and Marc Dymetman. Distributional Reinforcement Learning For Energy-Based Sequential Models. *CoRR*, 2019b. URL https://arxiv.org/abs/1912.08517.

Ramakanth Pasunuru and Mohit Bansal. Reinforced video captioning with entailment rewards. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 979–985. Association for Computational Linguistics, 2017. doi: 10.18653/v1/d17-1103. URL https://doi.org/10.18653/v1/d17-1103.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Romain Paulus, Caiming Xiong, and Richard Socher. A deep reinforced model for abstractive summarization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=HkAClQgA-.

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

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models, 2022. URL https://arxiv.org/abs/2202.03286.

Jan Peters and Stefan Schaal. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21(4):682–697, 2008. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2008.02.003. URL https://www.sciencedirect.com/science/article/pii/S0893608008000701. Robotics and Neuroscience.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Marc'Aurelio Ranzato, Y-Lan Boureau, Sumit Chopra, and Yann LeCun. A unified energy-based framework for unsupervised learning. In Marina Meila and Xiaotong Shen (eds.), *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS 2007, San Juan, Puerto Rico, March 21-24, 2007*, volume 2 of *JMLR Proceedings*, pp. 371–379. JMLR.org, 2007. URL http://proceedings.mlr.press/v2/ranzato07a.html.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06732.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017a. URL http://arxiv.org/abs/1707.06347.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint: 1707.06347*, 2017b.

Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, and Nanyun Peng. The woman worked as a babysitter: On biases in language generation. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 3405–3410. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1339. URL https://doi.org/10.18653/v1/D19-1339.

David Silver, Satinder Singh, Doina Precup, and Richard S. Sutton. Reward is enough. *Artificial Intelligence*, 299:103535, 2021. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2021.103535. URL https://www.sciencedirect.com/science/article/pii/S0004370221000862.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. Learning to summarize from human feedback. *CoRR*, abs/2009.01325, 2020. URL https://arxiv.org/abs/2009.01325.

Richard S. Sutton. *Temporal credit assignment in reinforcement learning*. PhD thesis, University of Massachusetts, 1984.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, pp. 1057–1063, Cambridge, MA, USA, 1999. MIT Press.

Pradyumna Tambwekar, Murtaza Dhuliawala, Lara J. Martin, Animesh Mehta, Brent Harrison, and Mark O. Riedl. Controllable neural story plot generation via reward shaping. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 5982–5988. ijcai.org, 2019. doi: 10.24963/ijcai.2019/829. URL https://doi.org/10.24963/ijcai.2019/829.

Emanuel Todorov. Linearly-solvable markov decision problems. In B. Schölkopf, J. Platt, and T. Hoffman (eds.), *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2007. URL https://proceedings.neurips.cc/paper/2006/file/d806ca13ca3449af72a1ea5aedbed26a-Paper.pdf.

Lifu Tu, Richard Yuanzhe Pang, Sam Wiseman, and Kevin Gimpel. Engine: Energy-based inference networks for non-autoregressive machine translation. *ArXiv*, abs/2005.00850, 2020.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 4566–4575. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7299087. URL https://doi.org/10.1109/CVPR.2015.7299087.

Lex Weaver and Nigel Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 538–545, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

Laura Weidinger, John Mellor, Maribeth Rauh, Conor Griffin, Jonathan Uesato, Po-Sen Huang, Myra Cheng, Mia Glaese, Borja Balle, Atoosa Kasirzadeh, Zac Kenton, Sasha Brown, Will Hawkins, Tom Stepleton, Courtney Biles, Abeba Birhane, Julia Haas, Laura Rimell, Lisa Anne Hendricks, William S. Isaac, Sean Legassick, Geoffrey Irving, and Iason Gabriel. Ethical and social risks of harm from language models. *CoRR*, abs/2112.04359, 2021. URL https://arxiv.org/abs/2112.04359.

Johannes Welbl, Amelia Glaese, Jonathan Uesato, Sumanth Dathathri, John Mellor, Lisa Anne Hendricks, Kirsty Anderson, Pushmeet Kohli, Ben Coppin, and Po-Sen Huang. Challenges in detoxifying language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pp. 2447–2469. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.findings-emnlp.210. URL https://doi.org/10.18653/v1/2021.findings-emnlp.210.

Ronald J. Williams. Reinforcement-learning connectionist systems. Technical report, Northeastern University, 1987. Technical Report NU-CCS-87-3.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256, 1992a. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*, pp. 229–256, 1992b.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL http://arxiv.org/abs/1609.08144.

Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Texygen: A benchmarking platform for text generation models. In Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (eds.), *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*, pp. 1097–1100. ACM, 2018. doi: 10.1145/3209978.3210080. URL https://doi.org/10.1145/3209978.3210080.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *CoRR*, abs/1909.08593, 2019. URL http://arxiv.org/abs/1909.08593.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Appendix A.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In Appendix C we present proofs of all mathematical facts referred to in the paper.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code is included as supplementary material available to the reviewers and area chairs and will be made publicly available alongside the camera ready version of the paper.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In Appendix E we provide the hyperparameters used throughout our experiments and report our hardware configuration. In Appendix D, we describe in detail how $D_{\mathrm{KL}}(p, \pi_\theta)$ and $\mathrm{TVD}(p, \pi_\theta)$ were estimated and provide an extended pseudocode for our training loop in Algorithm 2.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] However, we found the variance across random seeds to be negligible and not comparing across random seeds is a standard practice when working with large language models where the cost of a single run is significant.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In Appendix E we report our hardware configuration.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] In Appendix E.
   (b) Did you mention the license of the assets? [Yes] In Appendix E.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]