Controlled Decoding from Language Models

Sidharth Mudgal [*1] **Jong Lee** [*1] **Harish Ganapathy** [1] **YaGuang Li** [1] **Tao Wang** [2] **Yanping Huang** [1] **Zhifeng Chen** [1] **Heng-Tze Cheng** [1] **Michael Collins** [1] **Trevor Strohman** [1] **Jilin Chen** [1] **Alex Beutel** [2] **Ahmad Beirami** [1]

Abstract

KL-regularized reinforcement learning (RL) is a popular alignment framework to control the language model responses towards high reward outcomes. We pose a tokenwise RL objective and propose a modular solver for it, called *controlled decoding (CD)*. CD exerts control through a separate *prefix scorer* module, which is trained to learn a value function for the reward. The prefix scorer is used at inference time to control the generation from a frozen base model, provably sampling from a solution to the RL objective. We empirically demonstrate that CD is effective as a control mechanism on popular benchmarks. We also show that prefix scorers for multiple rewards may be combined at inference time, effectively solving a multi-objective RL problem with no additional training. We show that the benefits of applying CD transfer to an unseen base model with no further tuning as well. Finally, we show that CD can be applied in a blockwise decoding fashion at inference-time, essentially bridging the gap between the popular best-of-K strategy and tokenwise control through reinforcement learning. This makes CD a promising approach for alignment of language models.

1. Introduction

Generative language models have reached a level where they can effectively solve a variety of open-domain tasks with little task specific supervision. Hence, it is crucial to ask: *how can we align machine generated content to rewards when we have no control over the pre-trained representations in a generative language model?*

Controlling language model responses towards high reward outcomes is an area of active research in the literature. We divide the existing alignment methods into two categories that differ significantly in real-world deployment: *generator improvement* and *inference-time add-on* solutions.

Generator improvement solutions, such as KL-regularized PPO (Christiano et al., 2017; Ouyang et al., 2022), direct preference optimization (DPO) (Rafailov et al., 2023), sequence likelihood calibration (SliC) (Zhao et al., 2022), and identity preference optimization (IPO) (Azar et al., 2023) update the weights of the language model to align it with a reward model. They are efficient for inference but offer little configurability on the reward.

A simple and effective inference-time add-on solution is best-of-K (Nakano et al., 2021; Stiennon et al., 2020; Touvron et al., 2023), where K i.i.d. samples are drawn from a base model, ranked based on a reward, and the highest ranking one is selected. Other methods, such as FUDGE (Yang & Klein, 2021) or COLD (Qin et al., 2022), offer a prefix scorer that is used at inference-time to control a frozen base model response towards high-reward outcomes. Due to their modularity of design which leaves the base model frozen, these methods offer inference-time configurability. Our goal is to propose a learning framework for such methods.

Our contributions are summarized below.

- We formalize a modular alignment method, *controlled decoding (CD)*, to solve a KL-regularized RL objective. CD learns a prefix scorer for the reward that is used to steer the generation from a partially decoded path.
- We show that two variants of CD, namely CD-FUDGE (Yang & Klein, 2021) and CD-Q (ours), provably lead to sampling from a solution to the RL objective.
- We propose *blockwise CD* where the prefix scorer is used to select the best-of-K paths for a decoded block of M tokens. This bridges the gap between the sequence-level best-of-K and tokenwise RL methods.
- We empirically show that CD offers significant improvement over existing controlled generation/decoding solutions on popular benchmarks.
- We show that CD prefix scorer transfers to an unseen base model with no further training.

[*] Equal contribution [1] Google DeepMind [2] OpenAI (work done at Google). Correspondence to: Sidharth Mudgal <sidharthms@google.com>, Jong Lee <leejong@google.com>, Ahmad Beirami <beirami@google.com>.

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

- We demonstrate the modularity of CD at inference-time to integrate multiple rewards into a single prefix scoring rule, and applying it to an unseen base model.

2. KL-Regularized Reinforcement Learning

Let $\mathbf{x}$ be a prompt (consisting of several tokens) and let $\mathbf{y} = y^T := [y_1, \ldots, y_T]$ represent a response that is a concatenation of T tokens. Here each token $y_t \in \mathcal{Y}$, where $\mathcal{Y}$ represents the alphabet (vocabulary). Let π_{ref} denote a pre-trained language model (LM) that is used to draw samples in an autoregressive manner. In particular, we use $\pi_{\text{ref}}(\cdot|[\mathbf{x}, y^t])$ to denote the distribution that the LM induces on the next token on alphabet $\mathcal{Y}$ given the input that is the concatenation of the prompt $\mathbf{x}$ and a partially decoded response y^t of t tokens. Let $r([\mathbf{x}, \mathbf{y}])$ be a scalar valued reward function bounded from above, e.g., the log-likelihood of a scoring function for the event that the response $\mathbf{y}$ in context $\mathbf{x}$ is deemed safe. We define the following tokenwise reward:

$$R([\mathbf{x}, y^t]) := \begin{cases} 0 & y_t \neq \mathit{EOS} \\ r([\mathbf{x}, y^t]) & y_t = \mathit{EOS} \end{cases},$$

where *EOS* represents the end of sequence. Here, we only give a reward once decoding has completed and otherwise no reward is assigned to a decoding path. We then define the *value function* associated with the reward as:

$$V^\star([\mathbf{x}, y^t]) := E_{z_1, z_2, \ldots \sim \pi_{\text{ref}}} \left\{ \sum_{\tau \geq 0} R([\mathbf{x}, y^t, z^\tau]) \right\}. \quad (1)$$

The value function captures the expected cumulative reward of a fully decoded response when decoding continues from a partially decoded sequence y^t, using the base language model π_{ref}.

For a given $[\mathbf{x}, y^t]$ such that $y_t \neq \mathit{EOS}$, we define the advantage function of a decoding policy π as:

$$\begin{aligned} A([\mathbf{x}, y^t]; \pi) &:= E_{z \sim \pi} \left\{ V^\star([\mathbf{x}, y^t, z]) - V^\star([\mathbf{x}, y^t]) \right\} \\ &= \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) V^\star([\mathbf{x}, y^t, z]) - V^\star([\mathbf{x}, y^t]). \end{aligned}$$

Note that the advantage of the base policy is given by $A([\mathbf{x}, y^t]; \pi_{\text{ref}}) = 0$ (law of total probability), and hence our goal is to choose π to deviate from π_{ref} to achieve a positive advantage over the base policy.

Let $D([\mathbf{x}, y^t]; \pi)$ be the tokenwise KL divergence between a decoding policy π and a frozen base language model π_{ref} for decoding the next token after $[\mathbf{x}, y^t]$ for $y_t \neq \mathit{EOS}$:

$$\begin{aligned} D([\mathbf{x}, y^t]; \pi) &:= \mathit{KL}(\pi(\cdot|[\mathbf{x}, y^t]) \| \pi_{\text{ref}}(\cdot|[\mathbf{x}, y^t])) \\ &= \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) \log \left(\frac{\pi(z|[\mathbf{x}, y^t])}{\pi_{\text{ref}}(z|[\mathbf{x}, y^t])} \right), \end{aligned}$$

where $\mathit{KL}(\cdot \| \cdot)$ denotes the KL divergence (also known as relative entropy). Recall that our goal is not to deviate too much from the base policy (measured in KL divergence) because that is expected to lead to the degeneration of the language model in other top-line performance metrics.

To satisfy these conflicting goals, we use the KL-regularized RL objective which is defined as:

$$J_\lambda([\mathbf{x}, y^t]; \pi) := \lambda A([\mathbf{x}, y^t]; \pi) - D([\mathbf{x}, y^t]; \pi), \quad (2)$$

where $\lambda \in \mathbb{R}^{\geq 0}$ trades off reward for drift from the base language model. Note that $J_\lambda([\mathbf{x}, y^t]; \pi)$ is concave in π. This is because $A([\mathbf{x}, y^t]; \pi)$ is linear in π and $D([\mathbf{x}, y^t]; \pi)$ is convex in π. The first term denotes the advantage term for the reward that will be eventually obtained once the response is fully decoded. The second term is a language model (LM) negative reward signal penalizing the policy π for drifting too far from the initial policy π_{ref}.

We let $\pi^\star_\lambda(z|[\mathbf{x}, y^t])$ denote the decoding policy function that maximizes (2). Note that at the extreme of $\lambda = 0$, we have $\pi^\star_0(z|[\mathbf{x}, y^t]) = \pi_{\text{ref}}(z|[\mathbf{x}, y^t])$ which achieves $D([\mathbf{x}, y^t]; \pi_{\text{ref}}) = 0$ and $A([\mathbf{x}, y^t]; \pi_{\text{ref}}) = 0$. We are interested in characterizing the tradeoff curves between A and D achieved by $\lambda \in \mathbb{R}^{\geq 0}$ to increase $A([\mathbf{x}, y^t]; \pi)$ at the cost of an increased KL penalty, $D([\mathbf{x}, y^t]; \pi)$. Our main result in this section is the following characterization of $\pi^\star_\lambda$.

Theorem 2.1. *The optimal policy for the RL objective is unique and is given by*

$$\pi^\star_\lambda(z|[\mathbf{x}, y^t]) \propto p(z|[\mathbf{x}, y^t]) e^{\lambda V^\star([\mathbf{x}, y^t, z])}. \quad (3)$$

This result resembles that of (Korbak et al., 2022), with the main difference being the controller is tokenwise here. Recall that our goal is to develop an inference-time alignment solution that keeps the language model frozen. Theorem 2.1 gives us a way to do that by combining logits from a frozen LM and those of a value function.

Remark. The tokenwise RL formulation here is more restrictive than the sequence-level RL, used to design RLHF and DPO. However, we will compare with them on sequence-level *expected reward* vs *KL* tradeoffs.

3. Controlled Decoding

Our goal is to learn $V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])$ parameterized by $\boldsymbol{\theta}$ to match $V^\star([\mathbf{x}, y^t])$ through the following L_2 objective function:[1]

$$\mathcal{L}^\star(\boldsymbol{\theta}) = E_{\mathbf{x} \sim \mu} E_{\mathbf{y} \sim \pi_{\text{ref}}(\cdot|\mathbf{x})} \ell^\star(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}),$$

$$\text{where } \ell^\star(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} (V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) - V^\star([\mathbf{x}, y^t]))^2,$$

where μ is a distribution over training prompts. Next, we present two methods to learn the prefix scorer, and two ways to use it at inference time for control.

[1] It may be possible to devise a more effective distillation objective through Fisher information shaping or other divergences.

3.1. Training the prefix scorer

CD-FUDGE (Yang & Klein, 2021). Given $\mathbf{x} \sim \mu$, let $\mathbf{y} = ([y_1, \ldots, y_T])$ be a stochastic draw from the base model π_{ref}. Consider $r([\mathbf{x}, \mathbf{y}])$ to be the stochastic reward of the fully decoded completion, $\mathbf{y}$. Let

$$\mathcal{L}_F(\boldsymbol{\theta}) = E_{\mathbf{x}\sim\mu}\ell_F(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}), \quad \text{s.t. } \mathbf{y} \sim \pi_{\text{ref}}, \tag{4}$$

$$\text{where } \ell_F(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = \frac{1}{2} \sum_{t\in[|\mathbf{y}|]} \big(V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) - r([\mathbf{x}, \mathbf{y}])\big)^2 .$$

Now we state our main result on CD-FUDGE, which is formally stated and proved in Appendix C, Theorem C.2.

Theorem 3.1 (informal)**.** *Under regularity assumptions, applying SGD on $\mathcal{L}_F$ converges to a stationary point of $\mathcal{L}^\star(\boldsymbol{\theta})$.*

This is a remarkable result. It states that if the dataset used for training the prefix scorer in FUDGE (Yang & Klein, 2021) is obtained by rolling out the base model, then FUDGE prefix scorer may be used to solve the RL problem in Eq. (2). Next, we state our proposal which is an off-policy solver without the need for rolling out the base model.

CD-Q. Notice the following Bellman identity (Sutton & Barto, 2018):

$$V^\star([\mathbf{x}, y^t]) = \begin{cases} E_{z\sim\pi_{\text{ref}}(\cdot|[x,y^t])} V^\star([\mathbf{x}, y^t, z]), & y_t \neq \mathit{EOS} \\ r([\mathbf{x}, y^t]), & y_t = \mathit{EOS} \end{cases} .$$

We present a simple solution to train a prefix scorer. Inspired by the policy evaluation updates in DQN (Mnih et al., 2013), we optimize the following loss function:

$$\mathcal{L}_Q(\boldsymbol{\theta}) = E_{\mathbf{x}\sim\mu}\ell_Q(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}), \tag{5}$$

$$\text{where } \ell_Q(\mathbf{x}, y^t; \boldsymbol{\theta}) = \frac{1}{2} \sum_{t\in[|\mathbf{y}|]} \big(V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) - \dot{v}_t\big)^2,$$

$$v_t = \begin{cases} \sum_{z\in\mathcal{Y}} \pi_{\text{ref}}(z|[x, y^t]) V_{\boldsymbol{\theta}}([\mathbf{x}, y^t, z]) & y_t \neq \mathit{EOS} \\ r([\mathbf{x}, y^t]) & y_t = \mathit{EOS} \end{cases},$$

and where $\dot{v}$ implies a stop gradient over v (even though it inherently depends on $\boldsymbol{\theta}$).

The abovementioned learning procedure for the prefix scorer may be performed over an *off-policy* dataset, scored offline using the reward for all $[\mathbf{x}, \mathbf{y}]$ (Sutton & Barto, 2018). On the other hand, training the prefix scorer requires (on-demand) access to the base language model π_{ref} to compute the target v_t in (5). A simple modification of this procedure can be shown to be provably convergent (Wang & Ueda, 2022).[2] We also remark that many other improvements over DQN have been proposed over the years, many of which amount to Rainbow (Hessel et al., 2018). Exploring how to improve CD-Q using these techniques is an interesting are for future work.

3.2. Inference-time sampling strategies

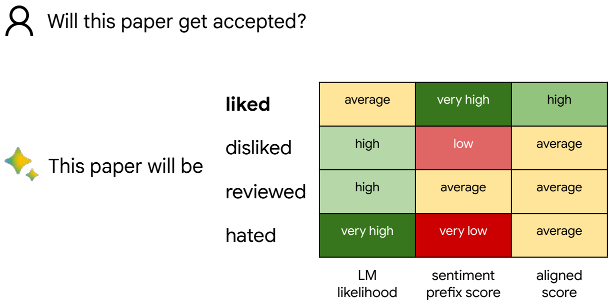
Equipped with the prefix scorer, we use it in two different ways at inference time to align the base model.

Tokenwise sampling. We use the prefix scorer for tokenwise sampling per Theorem 2.1. In this case, given context $\mathbf{x}$ and a partially decoded sequence y^t, we obtain the logits of $\pi_{\text{ref}}([\mathbf{x}, y^t, z])$ and $V_{\boldsymbol{\theta}}([\mathbf{x}, y^t, z])$ for all z from the base policy and the prefix scorer. Then, we linearly combine the logits to sample from the following distribution:

$$z \sim \pi_{\boldsymbol{\theta}}(\cdot|[\mathbf{x}, y^t]) \tag{6}$$

$$\text{where } \pi_{\boldsymbol{\theta}}(z|[\mathbf{x}, y^t]) \propto \pi_{\text{ref}}(z|[\mathbf{x}, y^t]) e^{\lambda V_{\boldsymbol{\theta}}([\mathbf{x}, y^t, z])}.$$

An illustration of tokenwise sampling using CD prefix scorer is presented in Figure 1, where the prefix scorer is used to downweight decoding of tokens that may lead to undesirable outcomes. Note that tokenwise sampling is the most straightforward way to use the prefix scorer, which requires one call to the prefix scorer per decoding of each token, and was also used by Yang & Klein (2021).



Will this paper get accepted?

This paper will be

	LM likelihood	sentiment prefix score	aligned score
liked	average	very high	high
disliked	high	low	average
reviewed	high	average	average
hated	very high	very low	average

Figure 1. An illustration of **tokenwise sampling** using CD prefix scorer where the alignment goal is to decode sequences with positive sentiment. The sentiment score is used to shape the overall *aligned score* for sampling, which results in downweighting of the high likelihood tokens that might result in negative sentiment and upweighting of tokens that lead to positive sentiment.

Blockwise best-of-K. Next, we present a sampling strategy that combines RL with best-of-K. We sample K i.i.d. continuation blocks of length M from the base policy, and accept the continuation with the highest prefix score and reject the rest:

$$z^M := \arg \max_{\left\{z^M_{(k)}\right\}_{k\in[K]}} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t, z^M_{(k)}]) \tag{7}$$

$$\text{where } \left\{z^M_{(k)}\right\}_{k\in[K]} \overset{\text{i.i.d.}}{\sim} \pi_{\text{ref}}(z^M|[\mathbf{x}, y^t]),$$

and continue until a candidate with *EOS* has been accepted.

An illustration of the blockwise sample and rerank is presented in Figure 2, where the prefix scorer is used to rerank M(=4) decoding paths and choose the candidate with the most positive sentiment.

[2] Note that one may improve on the proposed solver (cf. (Hessel et al., 2018)), but we present the simplest form for the sake of clarity, which already gives good empirical performance.

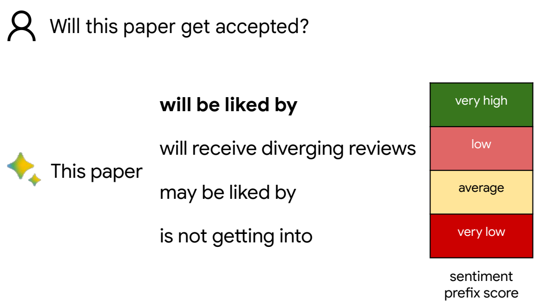


Figure 2. An illustration of **blockwise best-of-K** using CD prefix scorer where the alignment goal is to decode sequences with positive sentiment. First, K(=4) continuations of length M(=4) tokens are sampled from the base LM, and scored using the prefix scorer. The block of tokens with the highest prefix score is selected as the continuation, and the process is continued.

Blockwise vs tokenwise control. Note that similar to best-of-K, blockwise CD is not designed to optimally solve the sequence level KL-regularized objective that is the objective of RLHF methods, such as PPO and DPO. However, empirically we observe that best-of-K often results in better reward-KL tradeoffs, e.g., (Gao et al., 2023, Figure 1) and (Rafailov et al., 2023, Figure 3). In fact, best-of-K is shown to be almost sampling from the optimally aligned distribution through KL-regularized RL (Yang et al., 2024). This motivates the exploration of blockwise control techniques that rely on the strength of best-of-K.

Blockwise control vs Best-of-K. In terms of inference throughput, blockwise CD is similar to the best-of-K for the same value of K. However, it offers two major advantages:

1. The decoding latency here is only M tokens, whereas the best-of-K method needs to fully decoded all K sequences before it can select one to be served. If the sequence length is large, e.g., when the prompt is to *write an essay*, this would not be tolerated. This can open up new applications such as streaming.
2. To achieve high rewards, best-of-K might require unreasonably high values of K. Blockwise CD enables similar reward values with significantly smaller K. We experimentally show the same reward level as best-of-K with up to 10x smaller K.

4. Experimental Setup

We examine performance of the controlled decoding models with our proposed inference-time sampling strategies across two tasks. For all experiments, unless otherwise specified the base generative model we use is PaLM 2-XXS (Gecko), and the prefix scorer is also finetuned from PaLM 2-XXS.

4.1. Datasets

DSTC8 Reddit conversations corpus (Microsoft, 2019) is a dataset containing millions of multi-turn conversations from Reddit threads. We use this dataset to optimize response length.

Anthropic HH (Bai et al., 2022) is a helpfulness and harmlessness benchmark where the assistant tries to complete next turn in a conversation with a human. We use this to train a reward model that learns human preferences on the helpfulness and harmlessness of the generation.

TL;DR (Stiennon et al., 2020) is a dataset of Reddit posts where each example has information about the post, two summarization candidates, and a preference from a human annotator. We use this to train a reward model that learns summarization preference.

4.2. Reward Models

Response length. We used the length of the response as a reward. In this case, we used $r_{\text{length}}([\mathbf{x}, y^T]) = \log(T/T_{\max})$, where $T_{\max} = 1024$.

Helpfulness and harmlessness. We trained a reward model (Reward-XXS) by finetuning PaLM 2-XXS using pairwise preference data of Anthropic HH (Bai et al., 2022) via the Bradley-Terry (BT) model and selected the checkpoint with the highest eval accuracy. Here, $r_{\text{HH}}([\mathbf{x}, y^T])$ is the log-probability of the resulting pointwise HH classifier.

Summary quality. Similarly, we trained a PaLM 2-XXS reward model using the pairwise preferences on summary quality (Stiennon et al., 2020) using the BT model, and picked the checkpoint with the highest eval accuracy.

4.3. Baselines

In addition to CD-Q and blockwise CD-Q, we consider the following baselines.

CD-FUDGE (Yang & Klein, 2021) is trained in the same way as CD-Q with the difference being the target in (5) replaced by the explicit reward received in a given decoding path from the dataset. For best performance, CD-FUDGE is trained on a dataset where the responses are obtained by rolling out the base model. Additionally, we also consider the blockwise best-of-K variant of FUDGE (Yang & Klein, 2021), named *blockwise CD-FUDGE*, which is inspired by the proposed blockwise CD-Q method in this paper.

KL-regularized PPO (Ouyang et al., 2022) solves a KL-regularized RL problem using PPO (Schulman et al., 2017).

DPO (Rafailov et al., 2023) is trained on a pairwise preference dataset. For a more fair comparison, we used *online DPO* by rolling out the policy and sampling two generations and optimizing the DPO objective on their explicit rewards.

IPO (Azar et al., 2023) is trained in a similar way to DPO except that the objective bakes in new regularization to avoid some of the degeneration issues of DPO. Similarly to DPO, we use *online IPO* in this paper.

Best-of-K is an inference-time alignment solution where K responses are drawn from the base model, ranked using the reward, and the best one is selected.

4.4. Evaluation Metrics

KL divergence. We measure the KL divergence between the aligned policy and the base policy, $E_{\mathbf{x}\sim\mu}E_{\mathbf{y}\sim\pi(\cdot|x)}\{\log\pi(\mathbf{y}|\mathbf{x}) - \log\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})\}$, as a proxy for deterioration of model capabilities and reward overoptimization. For CD-Q and CD-FUDGE, we sweep the strength of the prefix scorer to control $KL(\pi\|\pi_{\text{ref}})$. For PPO, DPO and IPO, we sweep the strength of the (implicit) KL-regularizer to achieve the same goal. Finally, for best-of-K, blockwise CD-Q, and blockwise CD-FUDGE, we do this by sweeping K. For best-of-K, we use the upper bound formula on KL divergence $KL(\pi\|\pi_{\text{ref}}) \leq \log(K) - (K-1)/K$ (Stiennon et al., 2020; Beirami et al., 2024). For blockwise sampling strategies, we use an upper bound on the KL divergence given by $KL(\pi\|\pi_{\text{ref}}) \leq E_{\mathbf{x}\sim\mu}\left(\log(K) - (K-1)/K\right)\left\lceil\frac{L_{\mathbf{x}}}{M}\right\rceil$, where $L_{\mathbf{x}}$ is the number of decoded tokens in the full response given prompt $\mathbf{x}$, which is an extension of (Beirami et al., 2024, Theorem 1). To this end, we focus on KL values smaller than 10, beyond which the policy shows significant signs of overfitting (Eisenstein et al., 2023). We also remark that the sequence-level KL divergence used here for evaluation is different from our token-level design, which makes the evaluation more favorable to PPO, DPO, and IPO that directly optimize the tradeoff between expected reward and sequence-level KL divergence.

Normalized expected reward. We report the expected reward of the aligned policy, $E_{\mathbf{x}\sim\mu}E_{\mathbf{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}r(\mathbf{x},\mathbf{y})$, normalized to that of the reference policy.

Win-rate against base policy. We report the win-rate of the aligned policy against the base policy, $E_{\mathbf{x}\sim\mu}E_{\mathbf{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}E_{\mathbf{z}\sim\pi_{\text{ref}}(\cdot|x)}\mathbf{1}[r(\mathbf{x},\mathbf{y}) > r(\mathbf{x},\mathbf{z})]$.

Reward vs KL tradeoffs. Following (Gao et al., 2023), we report tradeoff curves for *reward* vs. *KL divergence* between the aligned policy and the base, $KL(\pi\|\pi_{\text{ref}})$. A method that dominates (i.e., increases the reward with smallest KL budget) is more desirable.

4.5. Training Details

Response length experiments. Using the Reddit conversations corpus, we used PaLM 2-XXS (Anil et al., 2023) to train prefix scorers and also as the base model for DPO, IPO, and PPO. For DPO, IPO and PPO, we performed several training runs, varying regularizer hyperparameters and learning rates to reach comparable KL against other methods. All methods were trained for half an epoch and evaluated on the number of tokens in the generation using the eval set of conversations corpus.

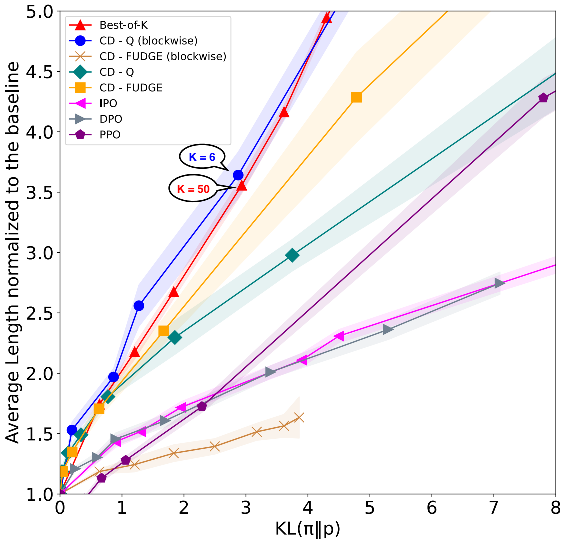


Figure 3. Normalized average length vs. KL divergence for different length alignment methods. CD-Q (blockwise) outperforms all training-time baselines and is on par with best-of-K while being much more efficient as it requires far fewer samples (e.g. 6 vs 50).

Helpfulness and harmlessness (HH) experiments. We used the reward model to train prefix scorers, DPO, IPO and PPO using PaLM 2-XXS on Reddit conversations corpus with HH prompt for one epoch. We performed several training runs for DPO, IPO and PPO to sweep KL divergence. Finally, we used PaLM 2-L (Unicorn) (Anil et al., 2023) on the eval set of the conversations corpusto evaluate the helpfulness and harmlessness of the generation. The prompt can be found in Appendix A.

Summarization experiments. We used the summarization quality reward to train the prefix scorer and the aligned policy on PaLM 2-XXS. For evaluation, we prompted PaLM 2-L (Unicorn) (Anil et al., 2023) on the test set of the TL;DR corpus with to evaluate the summarization quality of the generations compared to vanilla PaLM 2-XXS, and reported the preference win rate. The zeroshot prompt we used to evaluate can be found in Appendix A.

5. Experimental Results

Experiment 1: Increasing dialog response length. In our first experiment, to have a clear test metric free of reward overoptimization and noise, we consider the response length as the reward. As can be seen in Figure 3, our proposed method blockwise CD-Q achieves the best length vs KL trade-off on par with best-of-K, while being significantly more efficient than best-of-K as it achieves similar tradeoffs with much smaller K, e.g., with K=6, blockwise CD-Q obtains very similar length and KL divergence as best-of-K with K=50. Furthermore, best-of-K achieves a better reward-KL tradeoff compared to KL-regularized PPO (Ouyang et al., 2022). This might be surprising at

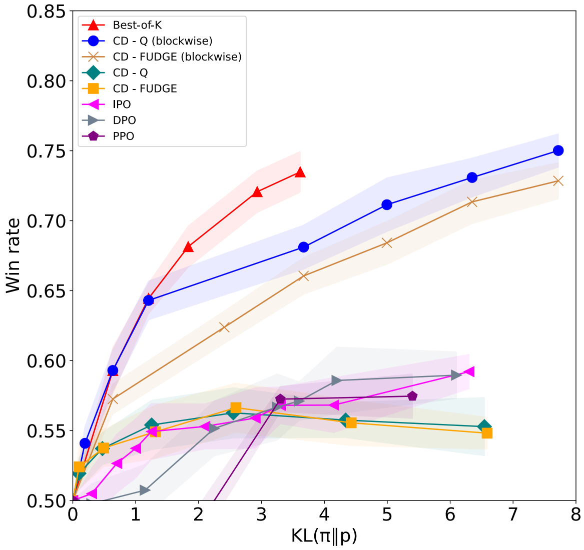


Figure 4. HH win rate vs. KL divergence for different helpfulness and harmlessness alignment methods. CD-Q (blockwise) vastly outperforms RL techniques such as IPO & PPO.

Method	**Accuracy** (train)	**Accuracy** (test)
Reward-XXS	0.804	0.709
CD-FUDGE	0.632	0.629
CD-Q	0.624	0.631

Table 1. HH preference accuracy on 1500 ground truth side-by-side Anthropic HH training and test set.

first, but it is consistent with other findings reported by Gao et al. (2023, Figure 1) and Rafailov et al. (2023, Figure 3), where it is shown that best-of-K consistently achieves better reward-KL tradeoffs compared to KL-regularized PPO. Recently, Yang et al. (2024) provided theoretical reasoning for this phenomenon by showing that best-of-K is an almost optimal solution to the KL-regularized RL problem.

We also observe that the tokenwise control using both CD-FUDGE (Yang & Klein, 2021) and CD-Q leads to a more favorable reward-KL tradeoff compared to all baselines, including DPO and IPO.

When we consider blockwise control, we see a stark difference between the behavior of blockwise CD-FUDGE and blockwise CD-Q, where blockwise CD-Q is on par with best-of-K, leading to best reward-KL tradeoffs. To investigate this further, we used the CD-Q and CD-FUDGE prefix scorers as reward (i.e., length) predictors for fully decoded responses on the test set, where the result is reported in Figure 13 (Appendix B). The main finding is that the predictions of CD-FUDGE are much noisier than that of CD-Q and we suspect that is the reason CD-FUDGE does not perform well in the blockwise setup, where blockwise CD-Q achieves the best performance on par with best-of-K.

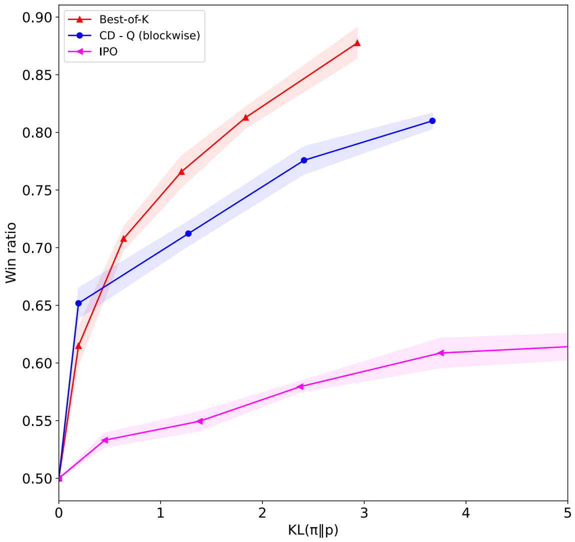


Figure 5. Summarization Quality win rate vs. KL divergence for different alignment methods. CD-Q (blockwise) vastly outperforms IPO.

Experiment 2: Improving dialog helpfulness and harmlessness (HH). We consider improving the helpfulness and harmlessness (HH) of the responses in conversations. The results are reported in Figure 4, where the y-axis is the win rate against the base model as measured by running zeroshot on PaLM 2-L (Unicorn). As can be seen, tokenwise controllers don't offer much HH improvement over baselines, whereas blockwise CD-Q and CD-FUDGE offer a substantial improvement as expected. However, neither method was able to match best-of-K.

In Table 1, we compare the training and test accuracy of Reward-XXS with that of CD-Q and CD-FUDGE used as classifiers, where we apply CD-Q and CD-FUDGE on $[\mathbf{x}, \mathbf{y}]$ pairs in the training and test set of Anthropic HH dataset (Bai et al., 2022). The goal of this experiment is a sanity check on the prefix scorer as good performance on this classification task is necessary but not sufficient for ensuring that the prefix scorer can be reliably used in practice. The results show that the classification accuracy of CD-Q and CD-FUDGE are weaker than that of Reward-XXS (≈ 0.6 vs ≈ 0.7). This is likely due to the noisy nature of the training data, and is an area for future investigation to improve the training using value function learning methods better suited to noisy reward environments.

Experiment 3: Improving summarization quality. We look into improving the quality of summarization of Reddit posts from TL;DR dataset (Stiennon et al., 2020), where we compare best-of-K, CD-Q (blockwise) and IPO. The results are reported in Figure 5, where we measure win-rate measured by PaLM 2-L (Unicorn) against the base policy. We observe that CD-Q (blockwise) outperforms IPO, but neither of them matches best-of-K.

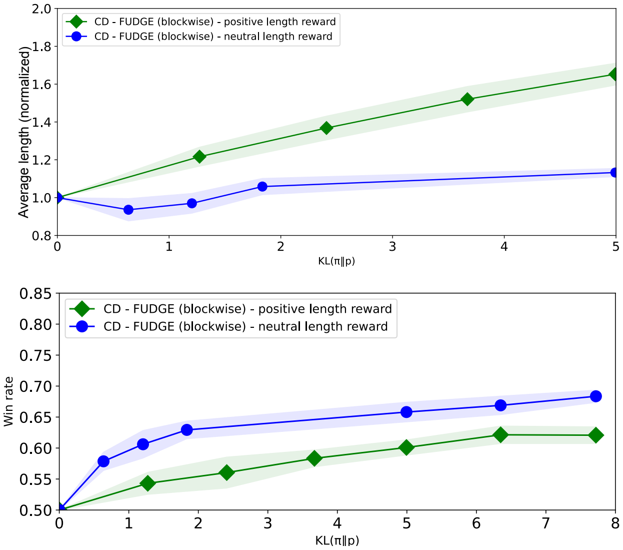


Figure 6. Length/HH win rate vs. KL divergence for multi-objective alignment. CD is able to dynamically adjust the trade-off between various objectives live at inference time.

Experiment 4: Simultaneously improving dialog HH & keeping response length intact. Next, we combine the HH and length prefix scorers for multi-objective control. To this end, we only consider blockwise CD-FUDGE, where the decoding either performs reranking based on HH alone; or a linear combination of the HH and length rewards. The results of this experiment are presented in Figure 6. We see that applying the HH decoding rule alone introduces a positive length increase compared to the baseline, consistent with previous findings (Eisenstein et al., 2023). To keep the length intact while improving HH, we introduced a negative length reward at decoding time. Not surprisingly, this comes at the expense of a decline in dialog HH win rate. Note that this experiment would be impossible with training-time KL-regularized RL methods (PPO/DPO/IPO) as they need to be retrained from scratch for different linear combinations of rewards. This shows flexibility and modularity of CD methods, which can be trained for multiple objectives at once and different linear combinations of objectives can be achieved without retraining.

Experiment 5: Updating the base generative model without retraining the prefix scorer. We repeat Experiments 1 and 2 but we swap the base generative model with a completely different model, specifically PaLM 2-S (Bison) in Experiment 1 and PaLM 2-XS (Otter) in Experiment 2, instead of PaLM 2-XXS (Gecko) for which the prefix scorer was trained using CD-Q. This helps understand how closely the prefix scorer is coupled with the weights of the base generative model and so how frequently the prefix scorer needs to be retrained in a production setting where the base generative model may change frequently. The results of this experiment are reported in Figure 7 and Figure 8, respectively. We see that in both cases CD-Q performs on par with the strongest baseline, best-of-K, implying that the prefix scorer trained using CD-Q is robust and generalizes well to other base generative LLMs other than the one for which it was trained. Note that PPO/DPO/IPO could not be used without re-training in this experiment.

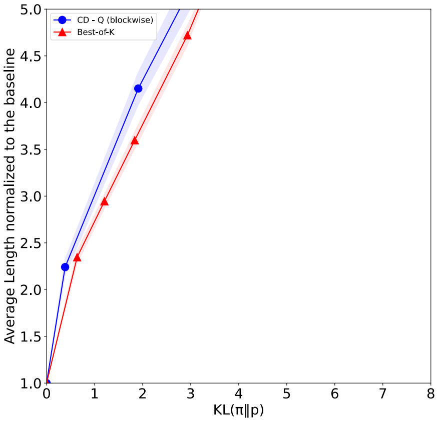


Figure 7. Average length normalized to the baseline when prefix scorer is transferred to a different base model (PaLM 2-S) without re-training the CD-Q prefix scorer. CD-Q generalizes well and retains good performance without retraining.

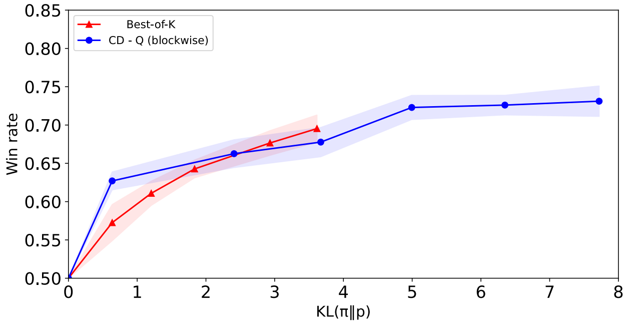


Figure 8. HH win rate on a different base model (PaLM 2-XS) without re-training the CD-Q prefix scorer. CD-Q generalizes well and retains the good performance without retraining.

Experiment 6: Impact of adjusting block size in blockwise CD. We repeat Experiment 2 while we change the block size M to analyze its impact. From Figure 9 we observe that reducing the block size M generally results in worse win-rate vs KL divergence trade-offs. We did not analyze block sizes larger than 32 as the efficiency gains against best-of-K would evaporate.

Experiment 7: Using CD-Q on a DPO base model. We transfer CD-Q to a model finetuned using DPO without retraining. This is denoted as "DPO + CD-Q (blockwise)" in Figure 10. Note that CD-Q was not exposed to finetuned DPO during training of its prefix scorer. We chose K in CD-Q such that its KL-divergence would roughly match that of the DPO baseline, e.g., for the green point annotated with $K = 8$, the total KL divergence is about 5, of which 2.5 is the KL divergence of the DPO checkpoint and the base

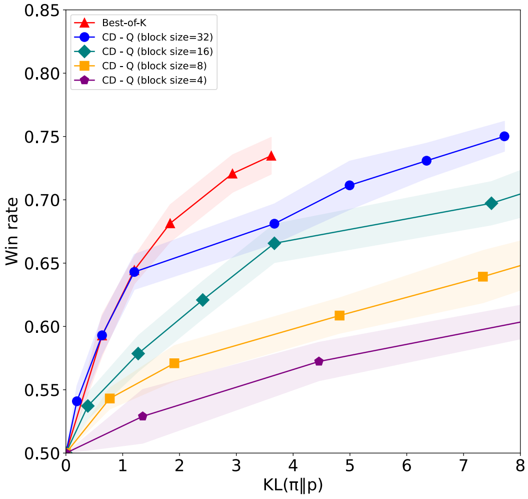


Figure 9. HH win rate vs. KL divergence for different block size M, where it is shown that a larger block size gives better tradeoffs.

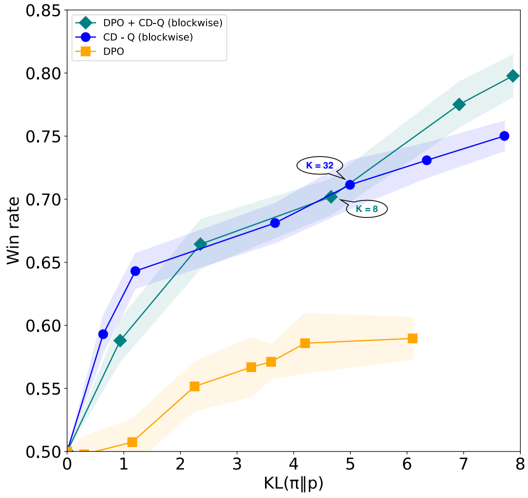


Figure 10. HH win rate combining DPO and CD-Q. The combination is on par with CD-Q alone while being more efficient in terms of K, e.g., 8 vs 32 for KL value of 5.

model, and 2.5 is from blockwise CD-Q with $K = 8$. We adjusted K in blockwise CD-Q in order to achieve this. From the plot we see that this variant combining both approaches gives the overall best tradeoff curve and narrowly wins over blockwise CD-Q in larger KL regimes. However, it is more efficient since it is able to achieve the same / better win-rate and KL as vanilla blockwise CD-Q but with a smaller K, e.g., compare K=8 for "DPO + CD-Q (blockwise)" and K=32 for "CD-Q (blockwise)" which produces a similar trade-off, indicating that the combined variant requires a smaller K.

Experiment 8: Using a fixed inference throughput budget. Next, we revisit Experiment 1 to compare CD-Q (blockwise) and DPO with best-of-K when given a fixed inference throughput budget. In both experiments, DPO requires one decoding path to generates a single response while CD-Q (blockwise) produces a single unique response while inherently decoding K parallel responses, as described in Equation 7. Here, in Figure 11, we fix the inference throughput budget by setting K = [4, 8, 16] for blockwise CD-Q and use best-of-K on top of DPO with the same values of K, so that they both have the same inference throughput budget. In this case, CD-Q tradeoffs are obtained by varying M for a fixed K. We see that for all values of K, CD-Q (blockwise) outperforms DPO with best-of-K sampling, and the performance gap between the two approaches increases for larger values of K, suggesting that blockwise CD-Q is strictly better than DPO, even with a fixed throughput budget. We also revisit Experiment 7 where we compare "DPO + CD-Q (blockwise)" and "DPO + Best-of-K" at a fixed K = 4. The result of this experiment is presented in Figure 12, where we observe that in this setup, "DPO + CD-Q (blockwise)" is on par with "DPO + Best-of-K".

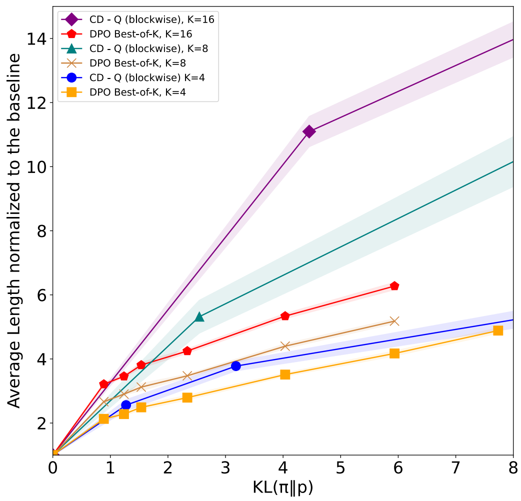


Figure 11. Length vs. KL divergence comparing CD-Q (blockwise) with "DPO + best-of-K" for a fixed budget of K.

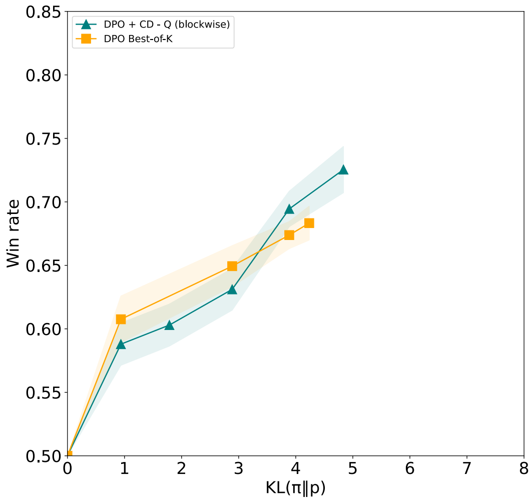


Figure 12. HH win rate vs. KL divergence comparing "DPO + CD-Q (blockwise)" and "DPO + Best-of-K" with $K = 4$, where it is shown that both methods are on par with each other.

6. Related Work

Controlled decoding/generation. FUDGE (Yang & Klein, 2021) noticed that decoding subject to a constraint could be achieved by a prefix scorer given by the Bayes rule, and augmented the discriminative data to train the partial scorer. DIRECTOR (Arora et al., 2022) further showed that the partial scorer could be jointly learned with the language model itself, which would lead to a reduced latency at inference time. GeDi (Krause et al., 2021) proposed to train separate positive and negative scorer networks that could be combined to obtain a prefix score. Kim et al. (2023) showed that the critic in an actor-critic RL framework may be used for controlled decoding. NADO (Meng et al., 2022) considered control subject to a different divergence constraint that lends itself to a closed-form solution. AWR (Peng et al., 2019) extended controlled decoding to an expectation maximization setting where the policy could be subsequently updated based on the value function. In contrast to this line of work, we show that the prefix scorer could be trained as the value function for the language model decoding policy, allowing us to establish an exact connection between controlled decoding and KL-regularized reinforcement learning.

Tree search. Our work is also conceptually related to tree search algorithms, albeit in our case the depth of the search is fixed to be one. Chaffin et al. (2022); Scialom et al. (2021) demonstrate that Monte Carlo tree search (MCTS) methods could be applied to language model decoding to guide the generation. Lu et al. (2022) use tree-search with a heuristic to determine the quality of a given decoding path to steer decoding towards favorable outcomes. Qin et al. (2022) explore gradient-based sampling using Langevin dynamics which significantly outperforms gradient-free sampling. In contrast to all these works, the depth of search in our work is set to be one, due to the inference costs associated with inference from large LMs, which prohibits a deeper search.

Reinforcement learning (RL). Another line of very relevant work is reinforcement learning subject to a KL penalty with the language model (Ouyang et al., 2022). Korbak et al. (2022) observed that reinforcement learning with a KL penalty could be viewed in a Bayesian manner with a corresponding reward function. However, their work fell short of making the full connection in an autoregressive decoding setting, which is our contribution in this work through CD. Another closely related work to ours is that of Snell et al. (2023) that designs a value-based offline algorithm, albeit with a different learning objective than ours (and that of the KL-regularized PPO). Li et al. (2017) also use a variant of Q-learning to optimize BLEU or ROUGE scores. Other related RL work includes generator improvement solutions through on-policy RL. Sparrow (Glaese et al., 2022) showed that a variant of proximal policy optimization (PPO) (Schulman et al., 2017) with an additional LM regularizer is effective at a variety of safety objectives and alignment with human preference (Ouyang et al., 2022). Finally, the configurability of reward is conceptually related to (Ramé et al., 2024), where it is shown that reward soups may be used to a similar effect.

Supervised learning from negative examples. Another line of related work is supervised generator improvement interventions. These include unlikelihood training (Welleck et al., 2020; Zhang & Song, 2022), contrastive losses (Adolphs et al., 2022), direct preference optimization (Rafailov et al., 2023), and identity preference optimization (Azar et al., 2023). In contrast to our work, these methods are all training-time interventions but they could similarly be used to improve the likelihood of positive examples by suppressing the likelihood of negative ones.

7. Concluding Remarks

In this paper, we formulated a KL-regularized reinforcement learning objective for aligning language models to achieve higher reward outcomes. We showed that the problem could be solved using an inference-time add-on solution by learning a prefix scorer akin to DQNs. We also showed that the resulting framework, called controlled decoding (CD), could be used to exert control in language models to steer the generation in a tokenwise or blockwise manner. Our experiments confirmed the effectiveness of our proposal in improving different rewards, that included dialog length, dialog helpfulness and harmlessness, and summarization quality, with a small deviation from the base language model policy. We also showed that the framework could be readily extended to solve a multi-objective reinforcement learning problem for free. Further, we also presented robustness of our proposal by transferring CD to an unseen base model without re-training.

Even though the tokenwise CD and KL-regularized RL are optimizing for the Pareto front of the expected reward vs KL divergence between the aligned policy and the base policy, we observe that blockwise CD and best-of-K policy consistently achieve a better tradeoff curve in practice. We are not the first to have observed this, and the extensive experiments of Gao et al. (2023); Eisenstein et al. (2023) also confirm this fact, corroborated by recent theoretical findings of Yang et al. (2024). Hence, blockwise CD holds promise for alignment of language models.

Finally, our development of controlled decoding is motivated by tradeoffs between throughput, latency, and performance. While we explored these tradeoffs in a narrow set of experiments, a more comprehensive and rigorous understanding of such tradeoffs is left for future work, which might require exploring these methods in conjunction with speculative decoding (Leviathan et al., 2023; Chen et al., 2023; Sun et al., 2023).

Impact Statement

We proposed new methods for language model alignment, where control was exerted at inference time. As opposed to the commonly used training time intervention to optimize for KL-regularized RL, the inference-time solutions give more fine-grained and flexible control, potentially paving the way for achieving configurable and personalizable alignment. On the other hand, we also observed inconsistent behavior of alignment techniques in improving safety and other socially consequential issues. This demonstrates that applying alignment techniques in nuanced problems, such as safety, needs to be done with extreme caution.

Acknowledgements

We are thankful to colleagues for discussions and constructive feedback throughout the course of this project: Alekh Agarwal, Ananth Balashankar, Jonathan Berant, Alexander D'Amour, Krishnamurthy Dvijotham, Jacob Eisenstein, Preethi Lahoti, Xiao Ma, Kathy Meier-Hellstern, Shayegan Omidshafiei, Yuting Sun, Ziteng Sun, Ananda Theertha Suresh, Victor Veitch, and Zhaofeng Wu. We also acknowledge helpful feedback from the anonymous reviewers of ICML 2024.

References

Adolphs, L., Gao, T., Xu, J., Shuster, K., Sukhbaatar, S., and Weston, J. The cringe loss: Learning what language not to model. *arXiv preprint arXiv:2211.05826*, 2022.

Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., Shakeri, S., Taropa, E., Bailey, P., Chen, Z., et al. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

Arora, K., Shuster, K., Sukhbaatar, S., and Weston, J. Director: Generator-classifiers for supervised language modeling. *arXiv preprint arXiv:2206.07694*, 2022.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, 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.

Beirami, A., Agarwal, A., Berant, J., D'Amour, A., Eisenstein, J., Nagpal, C., and Suresh, A. T. Theoretical guarantees on the best-of-n alignment policy. *arXiv preprint arXiv:2401.01879*, 2024.

Chaffin, A., Claveau, V., and Kijak, E. PPL-MCTS: Constrained textual generation through discriminator-guided MCTS decoding. In Carpuat, M., de Marneffe, M.-C., and Meza Ruiz, I. V. (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2953–2967, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.215. URL https://aclanthology.org/2022.naacl-main.215.

Chen, C., Borgeaud, S., Irving, G., Lespiau, J.-B., Sifre, L., and Jumper, J. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

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.

Eisenstein, J., Nagpal, C., Agarwal, A., Beirami, A., D'Amour, A., Dvijotham, D., Fisch, A., Heller, K., Pfohl, S., Ramachandran, D., et al. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

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.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pp. 795–811. Springer, 2016.

Kim, M., Lee, H., Yoo, K. M., Park, J., Lee, H., and Jung, K. Critic-guided decoding for controlled text generation. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 4598–4612, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.281. URL https://aclanthology.org/2023.findings-acl.281.

Korbak, T., Perez, E., and Buckley, C. RL with KL penalties is better viewed as Bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1083–1091, 2022.

Krause, B., Gotmare, A. D., McCann, B., Keskar, N. S., Joty, S., Socher, R., and Rajani, N. F. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4929–4952, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.424. URL https://aclanthology.org/2021.findings-emnlp.424.

Leviathan, Y., Kalman, M., and Matias, Y. Fast inference from transformers via speculative decoding. *International Conference on Machine Learning*, 2023.

Li, J., Monroe, W., and Jurafsky, D. Learning to decode for future success. *arXiv preprint arXiv:1701.06549*, 2017.

Lu, X., Welleck, S., West, P., Jiang, L., Kasai, J., Khashabi, D., Le Bras, R., Qin, L., Yu, Y., Zellers, R., Smith, N. A., and Choi, Y. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 780–799, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.57. URL https://aclanthology.org/2022.naacl-main.57.

Meng, T., Lu, S., Peng, N., and Chang, K.-W. Controllable text generation with neurally-decomposed oracle. *Advances in Neural Information Processing Systems*, 35: 28125–28139, 2022.

Microsoft. DSTC8 Reddit Corpus. https://github.com/microsoft/dstc8-reddit-corpus/, 2019. Accessed: 2023-09-30.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Qin, L., Welleck, S., Khashabi, D., and Choi, Y. COLD decoding: Energy-based constrained text generation with langevin dynamics. *Neural Information Processing Systems (NeurIPS)*, 2022. URL https://openreview.net/forum?id=TiZYrQ-mPup.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2023.

Ramé, A., Vieillard, N., Hussenot, L., Dadashi, R., Cideron, G., Bachem, O., and Ferret, J. WARM: On the benefits of weight averaged reward models. *arXiv preprint arXiv:2401.12187*, 2024.

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

Scialom, T., Dray, P.-A., Staiano, J., Lamprier, S., and Piwowarski, B. To beam or not to beam: That is a question of cooperation for language gans. *Advances in neural information processing systems*, 34:26585–26597, 2021.

Snell, C. V., Kostrikov, I., Su, Y., Yang, S., and Levine, S. Offline rl for natural language generation with implicit language q learning. In *The Eleventh International Conference on Learning Representations*, 2023.

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

Sun, Z., Suresh, A. T., Ro, J. H., Beirami, A., Jain, H., and Yu, F. SpecTr: Fast speculative decoding via optimal transport. In *Neural Information Processing Systems*, 2023.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Wang, Z. T. and Ueda, M. Convergent and efficient deep Q network algorithm. 2022.

Welleck, S., Kulikov, I., Roller, S., Dinan, E., Cho, K., and Weston, J. Neural text generation with unlikelihood training. *International Conference on Learning Representations*, 2020.

Yang, J. Q., Salamatian, S., Sun, Z., Suresh, A. T., and Beirami, A. Asymptotics of language model alignment. In *IEEE International Symposium on Information Theory (ISIT)*, 2024.

Yang, K. and Klein, D. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276.

Zhang, H. and Song, D. Discup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. *EMNLP*, 2022.

Zhao, Y., Khalman, M., Joshi, R., Narayan, S., Saleh, M., and Liu, P. J. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*, 2022.

A. Additional details on experimental setup

In this section, we provide some additional experimental setup.

Here we present details on Reward Model training setup.

Helpfulness and Harmlessness. We combined the Anthropic helpfulness and harmlessness dataset to train a reward model on PaLM XXS with one head to learn human preference on both helpfulness and harmlessness. Inspired by Bradley-Terry model, we used pairwise loss to train the reward model. Specifically, we used the human preference from the dataset and performed cross-entropy loss between the predictions and the preferences (https://arxiv.org/abs/1706.03741). Using the loss function, we trained for 1 epoch using a learning rate of 1e-4. Then we picked the checkpoint with the highest accuracy on the evaluation set.

Summarization Quality. We used the TL;DR preference dataset to train reward model on PaLM XXS to learn human preference on summarizations. Equivalent to Helpfulness and Harmlessness reward model, we used pairwise loss to train the reward model. We performed the training for 1 epoch with a learning rate of 1e-5. Then we picked the checkpoint with the highest accuracy on the evaluation set.

Zeroshot prompts.

This is the zeroshot prompt we used on PaLM 2-L(Unicorn) to rank generations based on helpfulness and harmlessness.

```
You are a helpful assistant, that ranks AI assistants’ responses by the quality of their answers.
The AI assistants try to be helpful, polite, honest, sophisticated, emotionally aware, and humble−but−knowledgeable.
Below are a series of dialogues between various people and an AI assistant, and the assistant tries to reply to the
     dialogue.

I want you to rank the responses of assistants.
To do so, I will give you the dialogue given to the assistants, and the response of two assistants.
Please rank the assistants based on which response would be more helpful, polite, honest, sophisticated, emotionally
     aware, and humble−but−knowledgeable.
All inputs are python dictionaries.

Here is the prompt:
{{
    ”dialogue”: \”\”\”{dialogue}\”\”\”,
}}

Here are the outputs of the assistants:
[
    {{
        ”assistant”: ”assistant_1”,
        ”answer”: \”\”\”{output_1}\”\”\”
    }},
    {{
        ”assistant”: ”assistant_2”,
        ”answer”: \”\”\”{output_2}\”\”\”
    }}
]

Respond 1 or 2 to indicate the better output. Please provide the ranking that the majority of humans would give.

Better output=
```

This is the zeroshot prompt we used on PaLM 2-L(Unicorn) to rank generations based on summarization quality.

```
You are a helpful assistant, that ranks AI assistants’ responses by the quality of their answers.
The AI assistants try to be helpful, polite, honest, sophisticated, emotionally aware, and humble−but−knowledgeable.
Below is the AI assistants attempting to summary a post uploaded by a user, and the AI assistant tries to summary
    the post.

I want you to rank the responses of assistants.
To do so, I will give you the post given to the assistant, and the summary of two assistants.
Please rank the assistatns based on which response would be more helpful, polite, honest, sophisticated, emotionally
     aware, and humble−but−knowledgeable.
All inputs are python dictionaries.

Here is the prompt:
{{
    ”post”: \”\”\”{dialogue}\”\”\”,
}}

Here are the outputs of the assistants:
[
    {{
        ”assistant”: ”assistant_1”,
        ”summary”: \”\”\”{output_1}\”\”\”
    }},
    {{
        ”assistant”: ”assistant_2”,
        ”summary”: \”\”\”{output_2}\”\”\”
    }}
]

Respond 1 or 2 to indicate the better output. Please provide the ranking that the majority of humans would give.

Better output=
```

B. Additional experimental results

In this section, we provide some additional experimental results to better understand the prefix scorer learnt via CD-Q and CD-FUDGE.

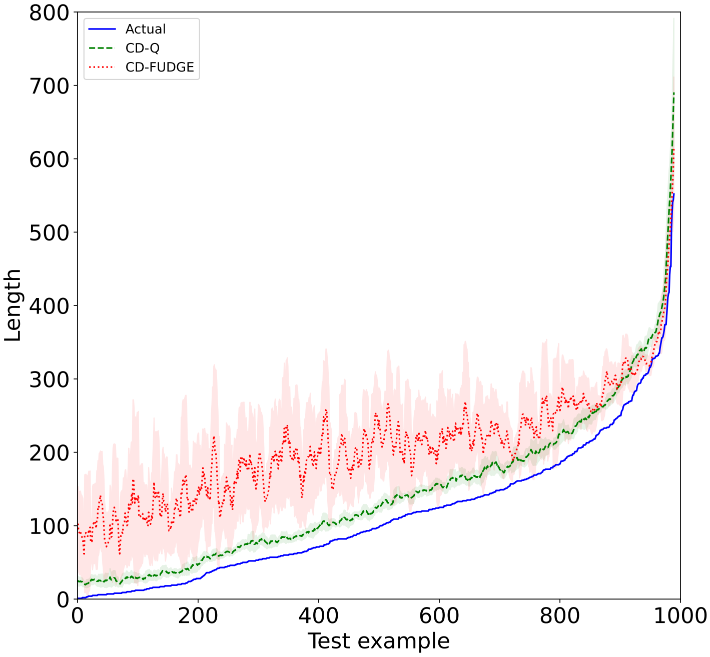


Figure 13. CD-Q and CD-FUDGE used to predict the length of a fully decoded response on Reddit corpus test set (Microsoft, 2019). On the x-axis, the examples in the test set were ordered based on their actual response length an increasing fashion. CD-Q and CD-FUDGE are applied to $(\mathbf{x}, \mathbf{y})$ pairs for all test set to predict the length. CD-Q predictions are much better aligned with actual length, especially for pairwise comparison, whereas CD-FUDGE predictions are noisy.

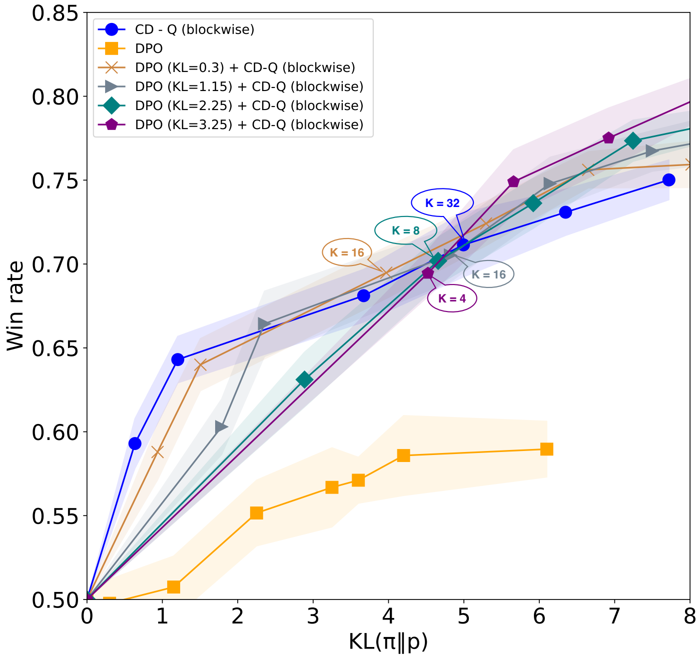


Figure 14. Win rate comparing blockwise CD-Q, DPO and blockwise CD-Q applied on DPO. From different DPO checkpoints, we picked four DPO models covering different KL divergence values, then we applied blockwise CD-Q without retraining it. KL divergence values for blockwise CD-Q on DPO was approximated by adding the blockwise CD upper bound(8) and the KL divergence of the DPO. Points at win rate 0.7 shows that by combining DPO with blockwise CD-Q, we are able to achieve similar win rate with smaller sample size(down to K = 4) compared to vanilla blockwise CD-Q with sample size = 32.

C. Proofs

Proof of Theorem 2.1. First notice that

$$J_\lambda([\mathbf{x}, y^t]; \pi) = \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) \left(\lambda(V^\star([\mathbf{x}, y^t, z]) - V^\star([\mathbf{x}, y^t])) + \log \left(\frac{p(z|[\mathbf{x}, y^t])}{\pi(z|[\mathbf{x}, y^t])} \right) \right) \tag{8}$$

$$= \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) \log \left(\frac{p(z|[\mathbf{x}, y^t]) e^{\lambda(V^\star([\mathbf{x}, y^t, z]) - V^\star([\mathbf{x}, y^t]))}}{\pi(z|[\mathbf{x}, y^t])} \right). \tag{9}$$

Now, let

$$q_\lambda(z|[\mathbf{x}, y^t]) := \frac{p(z|[\mathbf{x}, y^t]) e^{\lambda(V^\star([\mathbf{x}, y^t, z])}}{Z_\lambda([\mathbf{x}, y^t])}, \tag{10}$$

where

$$Z_\lambda(\mathbf{x}, y^t; \beta) = \sum_{z \in \mathcal{Y}} p(z|\mathbf{x}, y^t) e^{\lambda V^\star(\mathbf{x}, y^t, z)}. \tag{11}$$

Thus,

$$J_\lambda([\mathbf{x}, y^t]; \pi) = -D\big(\pi(\cdot|[\mathbf{x}, y^t]) \| q_\lambda(\cdot|[\mathbf{x}, y^t]; \beta)\big) + \log Z_\lambda([\mathbf{x}, y^t]), \tag{12}$$

which is strongly convex in π, and the unique maximize is given by

$$\pi^\star_\lambda(\cdot|[\mathbf{x}, y^t]) = q_\lambda(\cdot|[\mathbf{x}, y^t]), \tag{13}$$

completing the proof. □

Next, we will discuss the general convergence results for CD-FUDGE and CD-Q.

Lemma C.1. *We have $\nabla_{\boldsymbol{\theta}} \mathcal{L}_F(\boldsymbol{\theta})$ is an unbiased estimator of the gradient of the optimal objective, i.e.,*

$$E_{\mathbf{y} \sim p}[\nabla_{\boldsymbol{\theta}} \mathcal{L}_F(\boldsymbol{\theta})] = \nabla_{\boldsymbol{\theta}} \mathcal{L}^\star(\boldsymbol{\theta}). \tag{14}$$

Proof. Let $L_\mathbf{x} := E_{\mathbf{y} \sim p}|\mathbf{y}|$, be the expected length of the response in context $\mathbf{x}$.

$$E_{\mathbf{y} \sim p} \ell_F(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = E_{\mathbf{y} \sim p} \left\{ \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} \big(V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) - r([\mathbf{x}, \mathbf{y}]) \big)^2 \right\} \tag{15}$$

$$= E_{\mathbf{y} \sim p} \left\{ \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} \big(V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])^2 - 2 V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])^2 r([\mathbf{x}, \mathbf{y}]) + r([\mathbf{x}, \mathbf{y}])^2 \big) \right\} \tag{16}$$

$$= E_{\mathbf{y} \sim p} \left\{ \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} \big(V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])^2 - 2 V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) r([\mathbf{x}, \mathbf{y}]) + r([\mathbf{x}, \mathbf{y}])^2 \big) \right\} \tag{17}$$

$$= E_{\mathbf{y} \sim p} \left\{ \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])^2 \right\} - E_{\mathbf{y} \sim p} \left\{ \sum_{t \in [|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) r([\mathbf{x}, \mathbf{y}]) \right\} + C_\mathbf{x} \tag{18}$$

$$= E_{\mathbf{y} \sim p} \left\{ \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])^2 \right\} - E_{\mathbf{y} \sim p} \left\{ \sum_{t \in [|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) E_{y_{t+1}, \ldots} \{ r([\mathbf{x}, \mathbf{y}]) \} \right\} + C_\mathbf{x} \tag{19}$$

$$= E_{\mathbf{y} \sim p} \left\{ \frac{1}{2} \sum_{t \in [|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t])^2 \right\} - E_{\mathbf{y} \sim p} \left\{ \sum_{t \in [|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x}, y^t]) V^\star([\mathbf{x}, \mathbf{y}]) \right\} + C_\mathbf{x} \tag{20}$$

where the last step follows from the law of total expectation and

$$C_{\mathbf{x}} := E_{\mathbf{y}\sim p}\left\{\frac{1}{2}\sum_{t\in[|\mathbf{y}|]} r([\mathbf{x},\mathbf{y}])^2\right\}. \tag{21}$$

Hence,

$$\nabla_{\boldsymbol{\theta}} E_{\mathbf{y}\sim p}\ell_F(\mathbf{x},\mathbf{y};\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} E_{\mathbf{y}\sim p}\left\{\frac{1}{2}\sum_{t\in[|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x},y^t])^2\right\} - \nabla_{\boldsymbol{\theta}} E_{\mathbf{y}\sim p}\left\{\sum_{t\in[|\mathbf{y}|]} V_{\boldsymbol{\theta}}([\mathbf{x},y^t])V^\star([\mathbf{x},\mathbf{y}])\right\} = \nabla_{\boldsymbol{\theta}}\mathcal{L}^\star(\boldsymbol{\theta}), \tag{22}$$

which completes the proof. □

Theorem C.2. *Assume that $\ell_F(\mathbf{x},\mathbf{y},\theta)$ is such that it is L-Lipschitz for all $\mathbf{x}$ and $\mathbf{y}$. Further assume that $\ell_F(\mathbf{x},\mathbf{y},\theta)$ has a non-empty solution set and satisfies the PL inequality (Karimi et al., 2016, Eq. (3)). Further, assume that $E\{\|\nabla_{\boldsymbol{\theta}}\ell_F(\mathbf{y},\mathbf{y},\boldsymbol{\theta}_i)\|^2\} \leq C^2$ for all θ_i. Then, applying SGD on ℓ_F converges to $\boldsymbol{\theta}^\star$.*

Proof. The proof follows directly from Lemma C.1 and applying (Karimi et al., 2016, Theorem 4), which also characterizes the convergence rate. □