Controlled Decoding from Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose controlled decoding (CD), a novel off-policy reinforcement learning method to control the autoregressive generation from language models towards high reward outcomes. CD solves an off-policy reinforcement learning problem through a value function for the reward, which we call a prefix scorer. The prefix scorer is used at inference time to steer the generation towards higher reward outcomes. We show that the prefix scorer may be trained on (possibly) off-policy data to predict the expected reward when decoding is continued from a partially decoded response. We empirically demonstrate that CD is effective as a control mechanism on Reddit conversations corpus. We also show that the modularity of the design of CD makes it possible to control for multiple rewards, effectively solving a multi-objective reinforcement learning problem with no additional complexity. Finally, we show that CD can be applied in a novel blockwise fashion at inference-time, again without the need for any training-time changes, essentially bridging the gap between the popular best-of-K strategy and token-level 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 guide machine generated content to adhere to responsible AI principles, such as safety and factuality, 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 reinforcement learning (RL) (Christiano et al., 2017; Ouyang et al., 2022), direct preference optimization (DPO) (Rafailov et al., 2023), and sequence likelihood calibration (SliC) (Zhao et al., 2022) update the weights of the language model to align it with a reward model. On the other hand, inference-time add-on solutions, such as FUDGE (Yang & Klein, 2021) or COLD (Qin et al., 2022), devise techniques that are used at inference-time to control a frozen based model output towards high-reward outcomes. Due to their modularity of design which leaves the base model frozen, we are interested in the inference-time add-on solutions for responsible AI alignment.

Controlling a language model boils down to learning a *value function* that recognizes the eventual *reward* of a given decoding path (Yang & Klein, 2021; Korbak et al., 2022). In some applications, such a value function might be readily available in a rule-based manner, such as lexicographic constraints (Qin et al., 2022). On the other hand, responsible AI considerations, such as safety and factuality, are generally nuanced and require a data-driven approach to learning the value function, with a model that might be comparable to the base language model. Hence, the inference cost from such a value model is usually a non-negligible fraction of that of the base model, limiting the number of times the value model may be invoked as autoregressive decoding progresses. This renders

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

tree-search algorithms intractable due to latency considerations (Lu et al., 2022). In this paper, we focus on methods that resort to at most one call to a controller model per decoding of each token. See Appendix A.3 for a more complete discussion around the related control strategies.

A common inference-time guardrail to control generation from a language model is to sample k candidates and posthoc rerank them using a reward model to choose the *best-of-K*. This procedure effectively biases the generation to align it to the reward (Stiennon et al., 2020; Gao et al., 2023). While effective at alignment, best-of-K is computationally expensive, and is not applicable to situations that need streaming response generation. Additionally, the desired tradeoff point may require a prohibitively large k making it impractical to deploy in real world (Gao et al., 2023). On the other hand, best-of-K is less prone to reward hacking as the alignment is still happening on responses that are highly likely under the base model. Our contributions are summarized below.

- We propose *controlled decoding (CD)*, an alignment mechanism to increase reward in autoregressive language models subject to a KL constraint. The main ingredient in CD is a prefix scorer for the reward that is used to steer the generation from a partially decoded path.
- We demonstrate that the prefix scorer can be learnt in an *off-policy* manner using the Bellman update, which is significantly different from the on-policy RL alignment methods (such as PPO) that require model rollouts to update the model.
- We propose *blockwise CD* where control is exerted at every block of M tokens, with no additional training requirements. We decode K blocks of length M, and greedily keep the best according to the prefix scorer, and continue decoding from there. Note that $M \to \infty$ is effectively the best-of-K strategy. For intermediate M, this bridges the gap between best-of-K and token-wise control.
- We empirically show that the inference-time add-on control via CD (and its blockwise variant) offer significant improvement over existing controlled generation/decoding solutions on the tasks of improving dialog safety and increasing dialog length.
- We showcase the modularity of CD at inference-time to integrate multiple rewards into a single prefix scoring rule. Additionally, we demonstrate that it is possible to change the importance weight of different rewards via tuning a simple knob at inference time.

2 KL-Regularized Reinforcement Learning Setup

Let $\mathbf{x}$ be the 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 p denote a pre-trained language model (LM) from which we would like to draw samples in an autoregressive manner. In particular, we use $p(\cdot|[\mathbf{x}, y^t])$ 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 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 token-wise reward:

$$R([\mathbf{x}, y^t]) := \begin{cases} 0 & y_t \neq \textit{EOS} \\ r([\mathbf{x}, y^t]) & y_t = \textit{EOS} \end{cases}, \tag{1}$$

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:

$$V([\mathbf{x}, y^t]) := E_{z_1, z_2, \ldots \sim p} \left\{ \sum_{\tau \geq 0} \gamma^\tau R([\mathbf{x}, y^t, z^\tau]) \right\}, \tag{2}$$

where $\gamma \leq 1$ is a discount factor. This captures the expected cumulative reward of a fully decoded response when decoding continues from y^t using the base language model p.

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

$$A([\mathbf{x}, y^t]; \pi) := E_{z \sim \pi} \left\{ V([\mathbf{x}, y^t, z]) - V([\mathbf{x}, y^t]) \right\} = \gamma \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) V([\mathbf{x}, y^t, z]) - V([\mathbf{x}, y^t]). \tag{3}$$

Note that for $\pi = p$, we have $A([\mathbf{x}, y^t]; p) = 0$ (law of total probability), and hence our goal is to choose π to deviate from p to achieve a positive advantage over the base policy.

Let $D([\mathbf{x}, y^t]; \pi)$ be the token-wise KL divergence between a decoding policy π and a frozen base language model p for decoding the next token after $[\mathbf{x}, y^t]$:

$$D([\mathbf{x}, y^t]; \pi) := KL(\pi(\cdot|[\mathbf{x}, y^t]) \| p(\cdot|[\mathbf{x}, y^t])) = \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) \log \left(\frac{\pi(z|[\mathbf{x}, y^t])}{p(z|[\mathbf{x}, y^t])} \right), \tag{4}$$

where $KL(\cdot\|\cdot)$ 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:

$$J([\mathbf{x}, y^t]; \pi, \beta) := (1 - \beta) A([\mathbf{x}, y^t]; \pi) - \beta D([\mathbf{x}, y^t]; \pi), \tag{5}$$

where $\beta \in \mathbb{R}^+$ trades off reward for drift from the base language model. Note that $J([\mathbf{x}, y^t]; \pi, \beta)$ 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.

We let $\pi^\star(z|[\mathbf{x}, y^t]; \beta)$ denote the decoding policy function that maximizes Eq. (5). Note that at the extreme of $\beta = 1$, we have $\pi^\star(z|[\mathbf{x}, y^t]; 1) = p(z|[\mathbf{x}, y^t])$ which achieves $D([\mathbf{x}, y^t]; p) = 0$ and $A([\mathbf{x}, y^t]; p) = 0$. We are interested in characterizing the tradeoff curves achieved by $\beta \in (0, 1)$ 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 $\pi^\star$, with proof relegated to Appendix A.2.

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

$$\pi^\star(z|[\mathbf{x}, y^t]; \beta) \propto p(z|[\mathbf{x}, y^t]) e^{\frac{(1-\beta)\gamma}{\beta} V([\mathbf{x}, y^t, z])}. \tag{6}$$

This result resembles that of Korbak et al. (2022), with the main difference being the controller is token-wise here. Next, we develop our solution to the KL-regularized RL objective.

3 Proposed Method: Controlled Decoding (CD)

While Theorem 2.1 gives a recipe to solve the KL-regularized RL, it requires having access to the value function $V([\mathbf{x}, y^t])$, which we refer to as a *prefix scorer* since we use it at inference time to score the different decoding paths. Notice the following Bellman identity (Sutton & Barto, 2018):

$$V([\mathbf{x}, y^t]) = \begin{cases} \gamma \sum_{z \in \mathcal{Y}} p(z|[x, y^t]) V([\mathbf{x}, y^t, z]) & y_t \neq EOS \\ r([\mathbf{x}, y^t]) & y_t = EOS \end{cases}. \tag{7}$$

Let $V_w([\mathbf{x}, y^t])$ be called a prefix scorer, which is a transformer network parameterized by weights w to approximate $V([\mathbf{x}, y^t])$. Inspired by the policy evaluation updates in DQN (Mnih et al., 2013), we optimize the following loss function:

$$\ell(\mathbf{x}, y^t; w) = \left(V_w([\mathbf{x}, y^t]) - \dot{v}\right)^2, \text{ where } v = \begin{cases} \gamma \sum_{z \in \mathcal{Y}} p(z|[x, y^t]) V_w([\mathbf{x}, y^t, z]) & y_t \neq EOS \\ r([\mathbf{x}, y^t]) & y_t = EOS \end{cases} \tag{8}$$

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

The abovementioned learning procedure for the prefix scorer could be performed over an *off-policy* dataset, scored using the reward for all $[\mathbf{x}, \mathbf{y}]$ (Sutton & Barto, 2018). Training the prefix scorer requires (on-demand) access to the base language model p to compute the targets in Eq. (8).

Token-wise sampling. We use the prefix scorer for token-wise sampling per Theorem 2.1. In this case, given context $\mathbf{x}$ and a partially decoded sequence y^t, we obtain the logits of $p([\mathbf{x}, y^t, z])$ and $V_w([\mathbf{x}, y^t, z])$ for all z from the base policy and the prefix scorer. Then, we linearly combine the logits to sample from the following distribution:

$$z \sim \pi_w(z|[\mathbf{x}, y^t]) \quad \text{where} \quad \pi_w(z|[\mathbf{x}, y^t]) \propto p(z|[\mathbf{x}, y^t]) e^{\frac{1-\beta}{\beta} V_w([\mathbf{x}, y^t, z])}. \tag{9}$$

Block-wise sample and rerank. We also can use the prefix scorer as a reward for blockwise scoring. We sample K independent continuation blocks of length M from the base policy:

$$\left\{z^M_{(k)}\right\}_{k \in [K]} \overset{\text{i.i.d.}}{\sim} p(z^M|[\mathbf{x}, y^t]). \tag{10}$$

Then we accept the continuation with the highest prefix score and reject the rest:

$$z^M := \arg \max_{\left\{z^M_{(k)}\right\}_{k \in [K]}} V_w([\mathbf{x}, y^t, z^M_{(k)}]), \tag{11}$$

and continue until a candidate with EOS has been accepted.

4 Experimental Results

Dataset & model. Our experiments are performed on the DSTC8 Reddit conversations corpus (Microsoft, 2019), where we use PaLM 2 Gecko (Google, 2023) as the base model.

Baselines. We consider FUDGE (Yang & Klein, 2021), KL-regularized PPO (Ouyang et al., 2022), and best-of-K as baselines. Additionally, we also consider the blockwise decoding variant of FUDGE, that is inspired by the proposed blockwise CD method in this paper.

Evaluation. Following Gao et al. (2023), we report tradeoff curves for expected reward or win-rate over base policy vs. KL divergence between the aligned policy and the base, $KL(\pi\|p)$. A method that dominates (i.e., increases the expected reward with smallest sacrifice in KL divergence) is desirable.

Experiment 1: Increasing dialog response length. To decouple the effect of reward overoptimization, in our first task, we consider the response length as the reward. In particular, $r_{\text{length}}([\mathbf{x}, y^T]) = \log(T/T_{\max})$, where $T_{\max} = 1024$. As can be seen in Figure 1, best-of-K achieves a better reward-KL tradeoff compared to KL-regularized PPO (Ouyang et al., 2022). This might be surprising at first, but it is consistent with other findings reported by Gao et al. (2023); Rafailov et al. (2023), where it is shown that best-of-K consistently achieves better reward-KL tradeoffs compared to KL-regularized PPO. We also observe that the token-wise control using both FUDGE (Yang & Klein, 2021) and CD leads to a more favorable reward-KL tradeoff compared to KL-regularized RL. When we consider blockwise control, we see a stark difference between the behavior of blockwise FUDGE and blockwise CD, where blockwise CD in on par with best-of-K, leading to best reward-KL tradeoffs. To investigate this further, we used the CD and FUDGE prefix scorers as reward (i.e., length) predictors for fully decoded responses on the test set, where the result is reported in Figure 4 (Appendix A.3). The main finding is that the predictions of FUDGE are noisier than that of CD and we suspect that is the reason FUDGE does not perform well in the blockwise setup, where blockwise CD achieves the best performance on par with best-of-K.

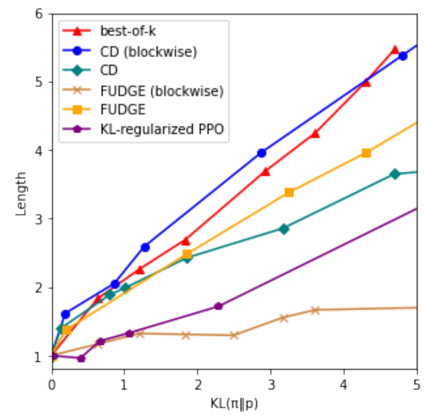

Figure 1: Length vs. KL divergence for different length alignment methods.

Experiment 2: Improving dialog safety. In this experiment, we consider improving the safety of the responses in conversations. We train two independent reward models on the side-by-side safety signal following the Anthropic HH dataset (Bai et al., 2022) using PaLM 2 Gecko (Reward-XXS) and PaLM 2 Otter (Reward-XS). The goal here is to generate safe responses in a dialog setting, where $r_{\text{safety}}([\mathbf{x}, y^T])$ could be roughly interpreted as the log-probability of a safety classifier. For all methods, we used Reward-XXS for training/control and kept Reward-XS solely for evaluations. The results are reported in Figure 2, where the y-axis is the win rate against the base model as measured by Reward-XS. As can be seen, token-wise controllers don't offer much safety improvement over baselines, whereas blockwise CD and FUDGE offer a substantial improvement as expected. However, neither method was able to match best-of-K.

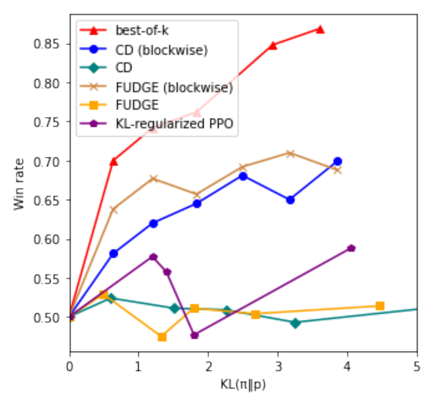

Figure 2: Win rate vs. KL divergence for different safety alignment methods.

In Table 1, we compare the training and test accuracy of Reward-XXS with that of CD and FUDGE used as classifiers, where we apply CD and FUDGE on $[\mathbf{x}, \mathbf{y}]$ pairs in the training and test set of Anthropic HH dataset (Bai et al., 2022). The results show that the predictive power of CD and FUDGE are much weaker than that of Reward-XXS, which is likely due to the noisy nature of the training data. This is an area for future investigation.

Method	**Accuracy** (train)	**Accuracy** (test)
Reward-XXS	0.710	0.696
FUDGE	0.616	0.626
CD	0.598	0.588

Table 1: Safety accuracy on 500 ground truth side-by-side Anthropic HH test set.

We also compare the average safety score of different variants token-wise FUDGE and CD (with varying β) to that of the base model for both Reward-XXS and Reward-XS. The results for this experiment are reported in Tables 2 and 3 (Appendix A.3). The main finding here is that the poor performance of token-wise CD and FUDGE may be partly attributed to overoptimization as well given that we observed more reasonable safety improvement when we used Reward-XXS as the judge; however, these gains didn't translate uniformly when we used the independent Reward-XS as the judge. A more clear understanding of these phenomena is left open for future work.

Experiment 3: Simultaneously improving dialog safety & increasing dialog length. Next, we combine the safety and length prefix scorers (rewards) to simultaneously improve safety and increase dialog length. To this end, we only consider blockwise CD and best-of-K, where the decoding either performs reranking based on safety alone; or a linear combination of the safety and length rewards (prefix scores). Note that this experiment does not need new models and only combine the scores from the two existing prefix scorers suffices to achieve this goal. Further, notice that this would be impossible using KL-regularized PPO as it needs to be retrained from scratch with this new combined reward.

The results of this experiment are presented in Figure 3. As can be seen, with a neutral length reward, the dialog length of blockwise CD remains mostly constant. On the other hand, it is interesting to note that best-of-K with no dialog length reward increases the dialog length around 3x. This might be attributable to potential spurious correlations between safety reward and length but it is left for further investigation. As expected, introducing a positive length reward (or prefix score) results in increasing dialog length both for blockwise CD and best-of-K. Not surprisingly, this comes at the expense of a decline in dialog safety win rate. Finally, similarly to the previous experiment, we observe a gap between best-of-K and blockwise CD in terms of the tradeoffs between performance metrics and KL divergence, which we hope future work can tackle to address.

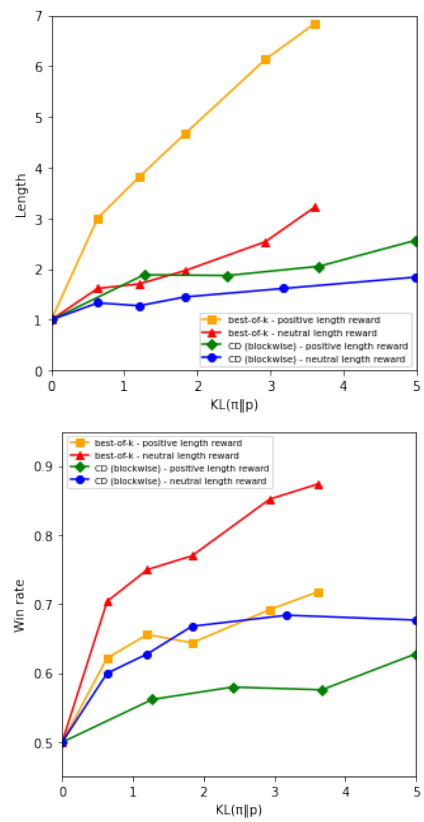

Figure 3: Length/Win rate vs. KL divergence for multi-objective alignment.

5 Conclusion

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 in an off-policy manner 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 token-wise or blockwise manner. Our experiments confirmed the effectiveness of our proposal in improving different rewards, that included dialog length and dialog safety, with a small deviation from the base language model policy. We also showed that the framework could be readily extended to solve a novel multi-objective reinforcement learning problem for free.

Social Impact Statement

We proposed new methods for language model alignment, where control was exerted at inference time. As opposed to the commonly used KL-regularized PPO, which is a training time intervention, the inference-time solutions give more fine-grained and flexible control, potentially paving the way for achieving personalized alignment, which is important when the reward functions encode socially consequential values. On the other hand, we also observed through experiments that alignment techniques may even lead to degradation of safety in responses whereas the goal of the experiment was to improve safety. This demonstrates that applying alignment techniques in nuanced issues, such as safety, needs to be done with extreme caution.

References

Leonard Adolphs, Tianyu Gao, Jing Xu, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. The cringe loss: Learning what language not to model. *arXiv preprint arXiv:2211.05826*, 2022.

Kushal Arora, Kurt Shuster, Sainbayar Sukhbaatar, and Jason Weston. Director: Generator-classifiers for supervised language modeling. *arXiv preprint arXiv:2206.07694*, 2022.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Google. PaLM 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

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

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 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.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 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.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

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

Charlie Victor Snell, Ilya Kostrikov, Yi Su, Sherry Yang, and Sergey Levine. Offline rl for natural language generation with implicit language q learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

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

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. *International Conference on Learning Representations*, 2020.

Kevin Yang and Dan Klein. 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.

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

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*, 2022.

A Appendix

A.1 Related Work

Controlled decoding. 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. In contrast to this line of work, we rigorously show that the prefix scorer should be trained as the value function for the language model decoding policy, which allows us to achieve significant improvements over this existing line of work.

Our work is also conceptually related to rule-based control. 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.

Reinforcement learning (RL). Another line of very relevant work is reinforcement learning subject to a KL penalty with the language model. 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 via a variant of the Bellman update akin to deep Q-learning (DQN) (Mnih et al., 2013). 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).

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).

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), and direct preference optimization (Rafailov 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 drawing positive examples by suppressing the likelihood of negative ones.

A.2 Proof of Theorem 2.1

Proof of Theorem 2.1. First notice that

$$J([\mathbf{x}, y^t]; \pi, \beta) = \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) \left((1-\beta)(\gamma V([\mathbf{x}, y^t, z]) - V([\mathbf{x}, y^t])) + \beta \log \left(\frac{p(z|[\mathbf{x}, y^t])}{\pi(z|[\mathbf{x}, y^t])} \right) \right) \tag{12}$$

$$= \beta \sum_{z \in \mathcal{Y}} \pi(z|[\mathbf{x}, y^t]) \log \left(\frac{p(z|[\mathbf{x}, y^t]) e^{\frac{1-\beta}{\beta}(\gamma V([\mathbf{x}, y^t, z]) - V([\mathbf{x}, y^t]))}}{\pi(z|[\mathbf{x}, y^t])} \right). \tag{13}$$

Now, let

$$q(z|[\mathbf{x}, y^t]; \beta) := \frac{p(z|[\mathbf{x}, y^t]) e^{\frac{(1-\beta)\gamma}{\beta} V([\mathbf{x}, y^t, z])}}{Z([\mathbf{x}, y^t]; \beta)}, \tag{14}$$

where

$$Z(\mathbf{x}, y^t; \beta) = \sum_{z \in \mathcal{Y}} p(z|\mathbf{x}, y^t) e^{\frac{(1-\beta)\gamma}{\beta} V(\mathbf{x}, y^t, z)}. \tag{15}$$

Thus,

$$J([\mathbf{x}, y^t]; \pi, \beta) = -\beta D\big(\pi(\cdot|[\mathbf{x}, y^t]) \| q(\cdot|[\mathbf{x}, y^t]; \beta)\big) + \beta \log Z([\mathbf{x}, y^t]; \beta), \tag{16}$$

which is maximized by

$$\pi(\cdot|[\mathbf{x}, y^t]) = q(\cdot|[\mathbf{x}, y^t]; \beta), \tag{17}$$

completing the proof. □

A.3 Additional experimental results

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

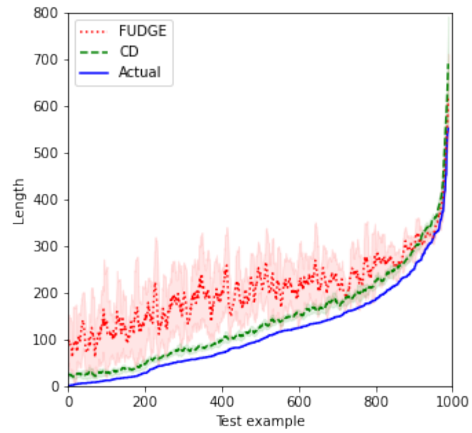

Figure 4: CD and FUDGE used to predict the length of a fully decoded response on Reddit corpus test set (Microsoft, 2019). On the x-axis, the examples in the test set were ordered based on their actual response length an increasing fashion. CD and FUDGE are applied to $(\mathbf{x}, \mathbf{y})$ pairs for all test set to predict the length. CD predictions are much better aligned with actual length, especially for pairwise comparison, whereas FUDGE predictions are noisy.

	FUDGE	**CD**
$\beta = 0$ (base)	1.0	1.0
$\beta = 0.10$	0.981	0.948
$\beta = 0.15$	0.959	0.961
$\beta = 0.20$	0.964	1.023
$\beta = 0.23$	0.926	0.990
$\beta = 2.00$	0.836	0.731

Table 2: Normalized average safety scores where the Reward-XS model (not used for alignment) is the judge, with 1000 generations from each model. The results are normalized to the average safety score of the base model. As can be seen, both prefix scorers generalize poorly which might be partly attributed to overoptimization.

	FUDGE	**CD**
$\beta = -0.50$	0.683	0.661
$\beta = -0.30$	0.792	0.745
$\beta = -0.20$	0.848	0.881
$\beta = 0$ (base)	1.0	1.0
$\beta = 0.100$	1.034	1.066
$\beta = 0.125$	1.009	1.007
$\beta = 0.150$	0.965	1.002
$\beta = 0.175$	0.984	1.021
$\beta = 0.200$	1.034	0.997
$\beta = 0.250$	1.011	1.04

Table 3: Normalized average safety scores where the Reward-XXS model (used for alignment) is the judge, with 1000 generations from each model. The results are normalized to the average safety score of the base model.