Relative Trajectory Balance is equivalent to Trust-PCL

Anonymous Author(s)
Affiliation
Address
email

Abstract

Recent progress in generative modeling has highlighted the importance of Reinforcement Learning (RL) for fine-tuning, with KL-regularized methods in particular proving to be highly effective for both autoregressive and diffusion models. Complementing this line of work, the Relative Trajectory Balance (RTB) objective was recently introduced in the context of Generative Flow Networks (GFlowNets) to serve the same role of improving fine-tuning in sequential generative models. Building on prior work linking GFlowNets and maximum-entropy RL, we establish in this paper an equivalence between RTB and Trust-PCL, an off-policy RL method with KL regularization. This equivalence situates RTB within the broader theoretical landscape of KL-regularized RL, and clarifies its relationship to earlier methods. Leveraging this insight, we revisit an illustrative example from the RTB paper and show that KL-regularized RL methods achieve comparable performance, offering an alternative perspective to what was previously reported.

1 Introduction

Generative models are playing an increasingly prominent role in artificial intelligence, ranging from language and reasoning [Brown et al., 2020, Wei et al., 2022], to vision [Ho et al., 2020, Rombach et al., 2022] and scientific discovery [Abramson et al., 2024]. At the heart of this rapid progress, fine-tuning is central to aligning models with downstream applications while preserving diversity and prior behavior. To complement supervised fine-tuning [Ouyang et al., 2022], Reinforcement Learning (RL) with KL regularization has proven effective and is widely used for this task [Jaques et al., 2017, Uehara et al., 2024], where reward maximization is traded off against maintaining consistency with a pretrained model. In parallel, Generative Flow Networks [GFlowNets; Bengio et al., 2021, 2023] have been proposed as an alternative paradigm to train generative models with techniques inspired by RL. They have since gained popularity in areas such as combinatorial optimization [Zhang et al., 2023], causal discovery [Deleu et al., 2022], and scientific discovery [Jain et al., 2023].

Extending ideas from GFlowNets, the *Relative Trajectory Balance* objective [RTB; Venkatraman et al., 2024] was recently proposed as an alternative to KL-regularized RL for fine-tuning generative models, framing the problem in terms of Trajectory Balance [Malkin et al., 2022]. In this paper, we show that RTB is exactly equivalent to an existing RL algorithm called *Trust-PCL* [Nachum et al., 2018], reinforcing the strong ties existing between GFlowNets and (entropy-regularized) RL [Tiapkin et al., 2024, Mohammadpour et al., 2024, Deleu et al., 2024, Jiralerspong et al., 2025]. In particular, while Venkatraman et al. [2024] suggested that KL-regularized RL methods may perform poorly even on simple tasks, we argue and show on their illustrative example that these issues stem from algorithmic and reward choices rather than from any fundamental limitation of RL with KL regularization itself. This new perspective places RTB within the well-established framework of KL-regularized RL and underscores its actual source of effectiveness.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

2 Background

We study the problem of training a *sequential generative model* capable of sampling objects as a sequence of multiple steps. This covers, for example, the case of diffusion samplers [Sendera et al., 2024] that generate images $\boldsymbol{x}_T$ as a sequence of denoising steps $\boldsymbol{x}_1 \to \boldsymbol{x}_2 \to \ldots \to \boldsymbol{x}_T$, starting from some Gaussian noise $\boldsymbol{x}_1 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and sequentially sampling from a denoising model $\boldsymbol{x}_{t+1} \sim P_\phi(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)$. By following this iterative process, we eventually obtain samples from the *marginal distribution* defined by

$$P_\phi^\top(\boldsymbol{x}_T) = \int P_\phi(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)\, d\boldsymbol{x}_{1:T-1} = \int \left[P(\boldsymbol{x}_1) \prod_{t=1}^{T-1} P_\phi(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t) \right] d\boldsymbol{x}_{1:T-1}. \tag{1}$$

We assume that we have access to a base sequential generative model, with transition probabilities $\pi_{\text{prior}}(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)$, such as a model pre-trained with a large dataset of observations, and that we are also given a fixed *energy function* $\mathcal{E}(\boldsymbol{x}_T)$ that we can query for any object generated by the model. Our objective in this paper is to learn a sequential generative model $P_\phi(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t)$ such that its marginal distribution is the *tilted distribution*, modulated by $\mathcal{E}(\boldsymbol{x}_T)$:

$$P_\phi^\top(\boldsymbol{x}_T) \propto \pi_{\text{prior}}^\top(\boldsymbol{x}_T) \exp(-\mathcal{E}(\boldsymbol{x}_T)/\alpha), \tag{2}$$

for some temperature parameter $\alpha > 0$. The role of the energy function is to steer generation towards samples with desired properties (*e.g.*, generating images based on a description).

2.1 Entropy-regularized Reinforcement Learning

One way to approach this problem is to view it from the perspective of Reinforcement Learning [Korbak et al., 2022]. We consider a finite-horizon Markov Decision Process (MDP) $\mathcal{M} = (\overline{\mathcal{S}}, \mathcal{A}, r)$, where $\overline{\mathcal{S}} = \mathcal{S} \cup \{\perp\}$ is a state space augmented by a special state $\perp \notin \mathcal{S}$ indicating the end of a trajectory, $\mathcal{A}$ is the action space, and r is a reward function that we will detail below. We identify a state $s_0 \in \mathcal{S}$ called the *initial state* from which all trajectories start; these trajectories are guaranteed to end in $\perp$ since the MDP is finite-horizon. To give a concrete example, we may see the multiple steps of denoising in a diffusion sampler as an MDP with a specific structure [Fan et al., 2023, Black et al., 2024]: its states are of the form $s_t = (\boldsymbol{x}_t, t)$, transitions (actions) correspond to applying one step of denoising $(\boldsymbol{x}_t, t) \to (\boldsymbol{x}_{t+1}, t+1)$, and trajectories must terminate when a state of the form $(\boldsymbol{x}_T, T)$ is reached after T steps (termination being indicated by the transition $(\boldsymbol{x}_T, T) \to \perp$).

Following Deleu et al. [2024], the reward function $r(s, s')$ obtained when transitioning from $s \to s'$ is defined such that the sum of rewards along a trajectory only depends on the energy of the final state it reaches right before terminating; in other words, for a trajectory $\tau = (s_0, s_1, \ldots, s_T, \perp)$, we have

$$\sum_{t=0}^{T} r(s_t, s_{t+1}) = -\mathcal{E}(s_T), \tag{3}$$

where we will use the convention $s_{T+1} = \perp$ throughout this paper. This includes in particular the case where the reward is only obtained at the end of the trajectory (*i.e.*, $r(s_T, \perp) = -\mathcal{E}(s_T)$, and zero everywhere else, also called *outcome-based reward*; Uesato et al., 2022).

Contrary to standard RL, where the objective is to find a policy $\pi(s_{t+1} \mid s_t)$ that maximizes the expected sum of rewards, *KL-regularized RL* includes an additional KL regularization term between the current policy and an anchor policy π_{prior} we don't want to deviate too much from:

$$\pi^\star_{\text{RelEnt}} = \arg\max_\pi \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{T} r(s_t, s_{t+1}) - \alpha \mathrm{KL}\big(\pi(\cdot \mid s_t) \,\|\, \pi_{\text{prior}}(\cdot \mid s_t)\big) \right]. \tag{4}$$

It can be shown that with the particular choice of reward function made in (3), the marginal distribution associated with this optimal policy is the tilted distribution [Nachum et al., 2018, Korbak et al., 2022]

$$\pi^{\star\top}_{\text{RelEnt}}(s_T) \propto \pi^\top_{\text{prior}}(s_T) \exp(-\mathcal{E}(s_T)/\alpha), \tag{5}$$

where the temperature parameter α naturally emerges from the regularization constant in (4); a proof of this is given in Appendix A for completeness. This constitutes the foundation of the recent line of work using RL for fine-tuning language models [Jaques et al., 2017, Ouyang et al., 2022] and diffusion models [Fan et al., 2023, Black et al., 2024, Uehara et al., 2024].

2.2 Relative trajectory balance

Taking another perspective, this time rooted in the literature on GFlowNets [Bengio et al., 2023], Venkatraman et al. [2024] introduced an objective called the *Relative Trajectory Balance* (RTB) loss to sample from the tilted distribution (2). Given a transition probability $P_\phi(s_{t+1} \mid s_t)$ and a scalar $Z_\psi > 0$, the RTB loss is a non-linear least-square objective $\mathcal{L}_{\text{RTB}}(\phi, \psi) = \frac{1}{2}\mathbb{E}_{\pi_b}[\Delta^2_{\text{RTB}}(\tau; \phi, \psi)]$, where π_b is an arbitrary distribution over trajectories (this is an *off-policy* objective), and the residual is defined as

$$\Delta_{\text{RTB}}(\tau; \phi, \psi) = \log \frac{\prod_{t=0}^{T} \pi_{\text{prior}}(s_{t+1} \mid s_t)}{Z_\psi \prod_{t=0}^{T} P_\phi(s_{t+1} \mid s_t)} - \frac{\mathcal{E}(s_T)}{\alpha}. \tag{6}$$

Venkatraman et al. [2024] showed that if this RTB loss is minimized, then the marginal distribution associated with the optimal $P_\phi(s_{t+1} \mid s_t)$ satisfies (2), and the optimal Z_ψ is the normalization constant of this tilted distribution.

3 Equivalence between RTB and Trust-PCL

In line with the growing body of work recently connecting GFlowNets with Maximum Entropy Reinforcement Learning (MaxEnt-RL) [Tiapkin et al., 2024, Mohammadpour et al., 2024, Deleu et al., 2024], where all the losses introduced in the GFlowNet literature have been shown to have a counterpart in MaxEnt-RL, one may wonder whether the RTB loss can also be viewed from the point of view of RL. In fact, we saw that solving (4) already provides a way to sample from the tilted distribution (5), and there exists an off-policy algorithm called *Trust-PCL* [Nachum et al., 2018] that does exactly this. For a policy $\pi_\phi(s_{t+1} \mid s_t)$ and a soft state-value function $V^\psi_{\text{soft}}(s)$, Trust-PCL is also defined as a non-linear least-square objective $\mathcal{L}_{\text{T-PCL}}(\phi, \psi) = \frac{1}{2}\mathbb{E}_{\pi_b}[\Delta^2_{\text{T-PCL}}(\tau; \phi, \psi)]$, where

$$\Delta_{\text{T-PCL}}(\tau; \phi, \psi) = -V^\psi_{\text{soft}}(s_0) + \sum_{t=0}^{T} r(s_t, s_{t+1}) + \alpha \log \frac{\pi_{\text{prior}}(s_{t+1} \mid s_t)}{\pi_\phi(s_{t+1} \mid s_t)}. \tag{7}$$

This form suggests that Relative Trajectory Balance is exactly equivalent to Trust-PCL, in the sense that both losses are equal up to a constant factor that only depends on the temperature α.

Proposition 1 (Equivalence RTB – Trust-PCL)**.** *The Relative Trajectory Balance loss [RTB; Venkatraman et al., 2024] defined with the residual in (6) is proportional to the Trust-PCL objective [Nachum et al., 2018] defined with the residual in (7) on the MDP in Section 2.1 (in particular, whose reward function satisfies (3)):* $\mathcal{L}_{\text{T-PCL}}(\phi, \psi) = \alpha^2 \mathcal{L}_{\text{RTB}}(\phi, \psi)$*, with*

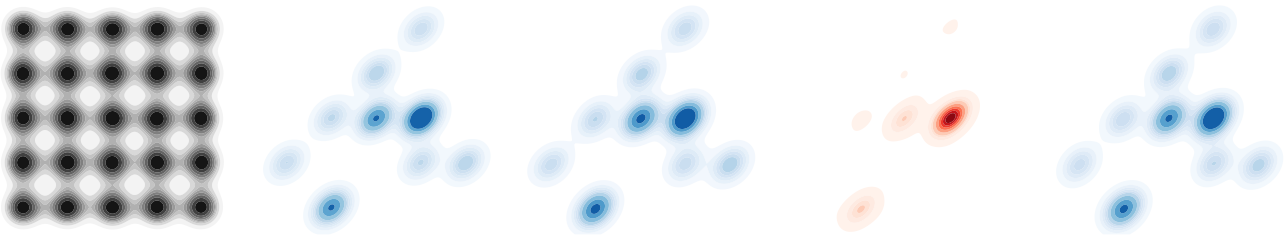
$$V^\psi_{\text{soft}}(s_0) = \alpha \log Z_\psi \qquad\qquad \pi_\phi(s' \mid s) = P_\phi(s' \mid s) \tag{8}$$

This result differs from similar ones relating GFlowNet objectives with MaxEnt-RL [Tiapkin et al., 2024, Deleu et al., 2024] in that unlike those existing connections, the equivalence in Proposition 1 does not require any correction of the reward function (with a backward transition probability). The proof of this proposition is immediate with the correspondence in (8).

3.1 Reinterpreting the empirical success of RTB

Validating Proposition 1 empirically by comparing the performance of Trust-PCL against RTB would be of limited interest, since they both optimize the exact same loss (up to a constant factor). Therefore, we instead choose to revisit the results presented in the RTB paper in light of this equivalence. While Venkatraman et al. [2024] argued that "*RL methods with KL regularization yield inaccurate inference*", our proposition suggests that this is not a fundamental issue of KL-regularized RL as a whole, and the success of RTB should be attributed to the choice of a superior RL algorithm, namely Trust-PCL.

To illustrate the apparent failure of RL with KL regularization for sequential generative modeling, Venkatraman et al. [2024] considered a 2D generation task where the prior is a uniform mixture of 25 Gaussians, and the target marginal distribution is a weighted mixture (Figures 1a & 1b). While RTB is capable of perfectly recovering the target distribution (Figure 1c), they observed that RL with KL regularization seems to not be capturing all the modes of the target distribution (Figure 1d).



(a) Prior $\pi_{\text{prior}}^{\top}$ (b) Target (c) RTB (d) RL with KL reg. (RTB paper) (e) REINFORCE with KL reg.

Figure 1: Comparison of RTB and RL with KL regularization on the illustrative example of [Venkatraman et al., 2024]. (a) Prior distribution π_{prior}. (b) Target tilted distribution $\propto \pi_{\text{prior}}^{\top}(x)\exp(-\mathcal{E}(x))$. (c) Model trained with RTB. (d) Model trained with REINFORCE with KL regularization, as reported by Venkatraman et al. [2024]. (e) Corrected model trained with off-policy REINFORCE with KL regularization.

At first glance, this could be explained by the difference in performance between their RL algorithm of choice REINFORCE [Williams, 1992] and Trust-PCL/RTB. However, we show in Appendix B that this happens to be caused by their reward function not satisfying (3). This, in turns, changes the target distribution (5), which does not match the one in Figure 1b anymore. We show in Figure 1e that if we apply the same method REINFORCE on the MDP of Section 2.1 with minimal changes to ensure a fair comparison (*e.g.*, using off-policy data with self-normalized importance sampling correction [Precup et al., 2000]; see Algorithm 2 for details), then the expected target distribution can also be recovered as accurately as with RTB.

Going beyond this illustrative example, Venkatraman et al. [2024] also compared RTB *on-policy* (*i.e.*, using $\pi_b \equiv P_\phi$ in Section 2.2) to existing methods based on RL for conditional image generation [Black et al., 2024, Fan et al., 2023]. Correcting for the discrepancy in the reward function mentioned above, we show in Appendix C that in this setting, RTB (on-policy) is equivalent to REINFORCE with KL regularization, up to a different control variate. This mirrors existing results in the GFlowNet literature [Malkin et al., 2023].

4 Conclusion

Reinforcement Learning with KL regularization has proven to be a versatile tool for fine-tuning sequential generative models. Its flexibility is such that even seemingly new methods for addressing the same problem can often be framed within this paradigm. In this work, we investigated the recently proposed Relative Trajectory Balance objective [Venkatraman et al., 2024] through the lens of entropy-regularized Reinforcement Learning. We demonstrated that it is exactly equivalent to the Trust-PCL algorithm [Nachum et al., 2018], indicating that RTB effectively rephrases an existing KL-regularized RL approach within the GFlowNet framework. This is further evidence of the strong theoretical connections between these fields.

Leveraging this theoretical correspondence, we revisited the empirical claims by Venkatraman et al. [2024] regarding the shortcomings of KL-regularized RL. By carefully examining the illustrative experiment from the RTB paper, we found that simple methods such as REINFORCE with KL regularization can match RTB in modeling the target tilted distribution on that task. The reported failures stemmed from algorithmic and reward design choices rather than any fundamental limitation of Reinforcement Learning. However, on more challenging tasks, advanced methods like RTB/Trust-PCL are likely to offer clearer benefits, as suggested by recent work incorporating techniques such as update clipping [Fan et al., 2023, Black et al., 2024, Shao et al., 2024] (inspired by PPO [Schulman et al., 2017]). These observations emphasize the need for fair comparisons and well-posed problem formulations, while offering promising directions for capitalizing on the connections between GFlowNets and RL to explore new algorithms for generative model fine-tuning.

References

J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 2024.

E. Bengio, M. Jain, M. Korablyov, D. Precup, and Y. Bengio. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Y. Bengio, S. Lahlou, T. Deleu, E. J. Hu, M. Tiwari, and E. Bengio. GFlowNet Foundations. *Journal of Machine Learning Research (JMLR)*, 2023.

K. Black, M. Janner, Y. Du, I. Kostrikov, and S. Levine. Training Diffusion Models with Reinforcement Learning. *International Conference on Learning Representations (ICLR)*, 2024.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

T. Deleu, A. Góis, C. Emezue, M. Rankawat, S. Lacoste-Julien, S. Bauer, and Y. Bengio. Bayesian Structure Learning with Generative Flow Networks. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2022.

T. Deleu, P. Nouri, N. Malkin, D. Precup, and Y. Bengio. Discrete Probabilistic Inference as Control in Multi-path Environments. *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2024.

Y. Fan, O. Watkins, Y. Du, H. Liu, M. Ryu, C. Boutilier, P. Abbeel, M. Ghavamzadeh, K. Lee, and K. Lee. DPOK: Reinforcement Learning for Fine-tuning Text-to-Image Diffusion Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

J. Ho, A. Jain, and P. Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

M. Jain, T. Deleu, J. Hartford, C.-H. Liu, A. Hernandez-Garcia, and Y. Bengio. GFlowNets for AI-Driven Scientific Discovery. *Digital Discovery*, 2023.

N. Jaques, S. Gu, D. Bahdanau, J. M. Hernández-Lobato, R. E. Turner, and D. Eck. Sequence Tutor: Conservative Fine-Tuning of Sequence Generation Models with KL-control. *International Conference on Machine Learning (ICML)*, 2017.

M. Jiralerspong, E. Derman, D. Vucetic, N. Malkin, B. Sun, T. Zhang, P.-L. Bacon, and G. Gidel. Robust Reinforcement Learning for Discrete Compositional Generation via General Soft Operators. *arXiv Preprint*, 2025.

T. Korbak, E. Perez, and C. L. Buckley. RL with KL penalties is better viewed as Bayesian inference. *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

N. Malkin, M. Jain, E. Bengio, C. Sun, and Y. Bengio. Trajectory Balance: Improved Credit Assignment in GFlowNets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

N. Malkin, S. Lahlou, T. Deleu, X. Ji, E. Hu, K. Everett, D. Zhang, and Y. Bengio. GFlowNets and variational inference. *International Conference on Learning Representations (ICLR)*, 2023.

S. Mohammadpour, E. Bengio, E. Frejinger, and P.-L. Bacon. Maximum entropy GFlowNets with soft Q-learning. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans. Trust-PCL: An Off-Policy Trust Region Method for Continuous Control. *International Conference on Learning Representations (ICLR)*, 2018.

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

D. Precup, R. S. Sutton, and S. Singh. Eligibility traces for off-policy policy evaluation. *International Conference on Machine Learning (ICML)*, 2000.

R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-Resolution Image Synthesis with Latent Diffusion Models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. *arXiv Preprint*, 2017.

M. Sendera, M. Kim, S. Mittal, P. Lemos, L. Scimeca, J. Rector-Brooks, A. Adam, Y. Bengio, and N. Malkin. On diffusion models for amortized inference: Benchmarking and improving stochastic control and sampling. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv Preprint*, 2024.

D. Tiapkin, N. Morozov, A. Naumov, and D. Vetrov. Generative Flow Networks as Entropy-Regularized RL. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.

M. Uehara, Y. Zhao, T. Biancalani, and S. Levine. Understanding reinforcement learning-based fine-tuning of diffusion models: A tutorial and review. *arXiv Preprint*, 2024.

J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins. Solving Math Word Problems with Process-based and Outcome-based Feedback. *arXiv Preprint*, 2022.

S. Venkatraman, M. Jain, L. Scimeca, M. Kim, M. Sendera, M. Hasan, L. Rowe, S. Mittal, P. Lemos, E. Bengio, A. Adam, J. Rector-Brooks, Y. Bengio, G. Berseth, and N. Malkin. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

R. J. Williams. Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine learning*, 1992.

D. Zhang, H. Dai, N. Malkin, A. Courville, Y. Bengio, and L. Pan. Let the Flows Tell: Solving Graph Combinatorial Optimization Problems with GFlowNets. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Appendix

A Marginal distribution of the optimal KL-regularized policy

In this section, we prove that the marginal distribution associated with the optimal policy maximizing (4) is the tilted distribution (5) for completeness, although this result can be found in various contexts in the literature [Nachum et al., 2018, Korbak et al., 2022]. First, recall that the KL-regularized RL objective can be written as

$$\pi^\star_{\mathrm{RelEnt}} = \arg\max_\pi \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} r(s_t, s_{t+1}) - \alpha \mathrm{KL}\big(\pi(\cdot \mid s_t) \,\|\, \pi_{\mathrm{prior}}(\cdot \mid s_t)\big)\right]. \tag{9}$$

It can be shown that the optimal policy $\pi^\star_{\mathrm{RelEnt}}$ can be written in terms of a soft state-value function $V^\star_{\mathrm{soft}}(s)$ and a soft state-action value function $Q^\star_{\mathrm{soft}}(s, s')$ as

$$\pi^\star_{\mathrm{RelEnt}}(s' \mid s) = \pi_{\mathrm{prior}}(s' \mid s) \exp\left(\frac{1}{\alpha}\big(Q^\star_{\mathrm{soft}}(s, s') - V^\star_{\mathrm{soft}}(s)\big)\right), \tag{10}$$

where the soft value functions, adapted to our setting where the MDP $\mathcal{M}$ is finite horizon and deterministic, satisfy

$$Q^\star_{\mathrm{soft}}(s, s') = r(s, s') + V^\star_{\mathrm{soft}}(s') \tag{11}$$

$$V^\star_{\mathrm{soft}}(s) = \alpha \log \sum_{s'\in \mathrm{Ch}(s)} \pi_{\mathrm{prior}}(s' \mid s) \exp\left(\frac{1}{\alpha} Q^\star_{\mathrm{soft}}(s, s')\right), \tag{12}$$

where $\mathrm{Ch}(s)$ are the children or the state s in the MDP (*e.g.*, all the states of the form $(\boldsymbol{x}_{t+1}, t+1)$ for a state $s_t = (\boldsymbol{x}_t, t)$ in the example MDP given in Section 2.1 [Fan et al., 2023, Black et al., 2024]). We defined the marginal distribution (1) in the main text for the particular example of diffusion samplers. However, this naturally generalizes in the context of KL-regularized RL, by viewing it as a marginal over trajectories in the MDP that terminate in s_T (*i.e.*, the trajectory terminates with a transition $s_T \to \perp$):

$$\pi^\top(s_T) = \int \pi(\tau \mid s_0) \mathbf{1}(s_T \to \perp \in \tau)\, d\tau \tag{13}$$

For a trajectory $\tau = (s_0, s_1, \ldots, s_T, \perp)$, we have

$$\pi^\star_{\mathrm{RelEnt}}(\tau \mid s_0) = \prod_{t=0}^{T} \pi^\star_{\mathrm{RelEnt}}(s_{t+1} \mid s_t) \tag{14}$$

$$= \left[\prod_{t=0}^{T} \pi_{\mathrm{prior}}(s_{t+1} \mid s_t)\right] \exp\left(\frac{1}{\alpha}\sum_{t=0}^{T} Q^\star_{\mathrm{soft}}(s_t, s_{t+1}) - V^\star_{\mathrm{soft}}(s_t)\right) \tag{15}$$

$$= \pi_{\mathrm{prior}}(\tau \mid s_0) \exp\left(\frac{1}{\alpha}\sum_{t=0}^{T} r(s_t, s_{t+1}) + V^\star_{\mathrm{soft}}(s_{t+1}) - V^\star_{\mathrm{soft}}(s_t)\right) \tag{16}$$

$$= \pi_{\mathrm{prior}}(\tau \mid s_0) \exp\left(\frac{1}{\alpha}(-\mathcal{E}(s_T) + \cancel{V^\star_{\mathrm{soft}}(\perp)} - V^\star_{\mathrm{soft}}(s_0))\right) \tag{17}$$

$$\propto \pi_{\mathrm{prior}}(\tau \mid s_0) \exp(-\mathcal{E}(s_T)/\alpha), \tag{18}$$

where we used the definition of $\pi^\star_{\mathrm{RelEnt}}$ in (15), the definition of $Q^\star_{\mathrm{soft}}(s_t, s_{t+1})$ in (16), the property that the sum of reward functions only depends on the energy at the end of the trajectory (3) and a telescoping sum in (17), and the fact that $V^\star_{\mathrm{soft}}(s_0)$ is a constant independent of τ (and $V^\star_{\mathrm{soft}}(\perp) = 0$) in (18). Plugging this in the definition of the marginal distribution (13), we get

$$\pi^{\star\top}_{\mathrm{RelEnt}}(s_T) = \int \pi^\star_{\mathrm{RelEnt}}(\tau \mid s_0) \mathbf{1}(s_T \to \perp \in \tau)\, d\tau \tag{19}$$

$$\propto \exp(-\mathcal{E}(s_T)/\alpha) \int \pi_{\mathrm{prior}}(\tau \mid s_0) \mathbf{1}(s_T \to \perp \in \tau)\, d\tau \tag{20}$$

$$\propto \pi^\top_{\mathrm{prior}}(s_T) \exp(-\mathcal{E}(s_T)/\alpha). \tag{21}$$

Algorithm 1 RL with KL reg. [Venkatraman et al., 2024]

1: Sample a batch of trajectories $\{\tau_1, \dots, \tau_N\}$ using π_ϕ

2: Compute the cumulative reward for τ_n:

$$r_n \leftarrow \exp(-\mathcal{E}(s_T^{(n)}))$$

3: Compute the advantage: $\bar{r}_n = r_n - \frac{1}{N}\sum_{n=1}^N r_n$

5: Compute the REINFORCE loss with KL reg.

$$\mathcal{L}(\phi) = \frac{1}{N}\sum_{n=1}^N -\mathrm{sg}(\bar{r}_n)\log \pi_\phi(\tau_n) + \frac{\lambda}{2}\left(\log \frac{\pi_\phi(\tau_n)}{\pi_{\mathrm{prior}}(\tau_n)}\right)^2$$

Algorithm 2 Off-policy REINFORCE with KL reg.

1: Sample a batch of trajectories $\{\tau_1, \dots, \tau_N\}$ using π_b

2: Compute the cumulative reward for τ_n:

$$r_n \leftarrow -\mathcal{E}(s_T^{(n)}) - \alpha \log \frac{\pi_\phi(\tau_n)}{\pi_{\mathrm{prior}}(\tau_n)}$$

3: Compute the advantage: $\bar{r}_n = r_n - \frac{1}{N}\sum_{n=1}^N r_n$

4: Get the SNIS weights: $w_n \propto \left(\pi_\phi(\tau_n)/\pi_b(\tau_n)\right)^{1/T}$

5: Compute the REINFORCE loss

$$\mathcal{L}(\phi) = -\frac{1}{N}\sum_{n=1}^N \mathrm{sg}(w_n \bar{r}_n)\log \pi_\phi(\tau_n)$$

B Off-policy REINFORCE with KL regularization

We saw in Section 2.1 that the marginal distribution associated with the optimal policy $\pi^\star_{\mathrm{RelEnt}}$ matches the tilted distribution (5) in the case where the reward function of the MDP satisfies

$$\sum_{t=0}^T r(s_t, s_{t+1}) = -\mathcal{E}(s_T). \tag{22}$$

In their comparison with KL-regularized RL, Venkatraman et al. [2024] instead used a sparse reward function that equals $\tilde{r}(s_T, \perp) = \exp(-\mathcal{E}(s_T))$ at the terminating transition, and zero everywhere else, meaning that their optimum would be

$$\pi^{\star\top}(s_T) \propto \pi^\top_{\mathrm{prior}}(s_T)\exp(\exp(-\mathcal{E}(s_T))/\alpha), \tag{23}$$

where we emphasize the extra inner "exp". This error likely came from the naming conflict between the "reward function" in RL and the "reward" in the GFlowNet literature, where the latter should indeed equal the exponential of the corresponding reward in RL [Deleu et al., 2024]. This can be seen in the residual for RTB (6), that can be re-written as

$$\Delta_{\mathrm{RTB}}(\tau; \phi, \psi) = \log \frac{\overbrace{\exp(-\mathcal{E}(s_T)/\alpha)}^{\text{GFlowNet reward } "R(s_T)"}\prod_{t=0}^T \pi_{\mathrm{prior}}(s_{t+1} \mid s_t)}{Z_\psi \prod_{t=0}^T P_\phi(s_{t+1} \mid s_t)}. \tag{24}$$

Besides this mismatch in reward functions, we made some minor updates to the REINFORCE [Williams, 1992] algorithm from Algorithm 1 considered in the RTB paper to Algorithm 2 ("sg" is the stop-gradient operator), to ensure a fair comparison with RTB:

- The trajectories are collected using the same exploratory behavior policy π_b as RTB, to address the well known limitations of on-policy methods in terms of exploration (this was also noted by Venkatraman et al. [2024]). To that end, we use importance sampling to address this shift in distributions (between π_b and the policy π_ϕ being learned), and more particularly self-normalized importance sampling [SNIS; Precup et al., 2000] to control the variance of the gradient estimate.
- The log-ratio $\log \pi_\phi/\pi_{\mathrm{prior}}$ is added to the reward itself, as opposed to being a squared regularization, so that it can be incorporated into the average control variate. Note that while Venkatraman et al. [2024] treated the regularization constant λ as a separate hyperparameter they had to tune carefully (with $\lambda = \alpha = 1$ being the best value they found), we saw in Section 2.1 that this exactly corresponds to the temperature parameter α of the tilted distribution.

Going beyond the illustrative example of Figure 1, we note that Venkatraman et al. [2024] only considered baselines based on vanilla REINFORCE in their comparisons with KL-regularized RL methods, and they did not use the advanced techniques such as update clipping that were used in practice by DDPO [Black et al., 2024] & DPOK [Fan et al., 2023].

C On-policy RTB is equivalent to REINFORCE with KL regularization

In this section, we will show that the gradient of on-policy RTB is equal to the gradient of REINFORCE with KL regularization, up to a constant α. First, recall that the gradient of the REINFORCE loss with KL regularization can be written as

$$\nabla_\phi \mathcal{L}_{\mathrm{RL}}(\phi) = -\mathbb{E}_{\pi_\phi}\left[\left(-\mathcal{E}(s_T) - \alpha \log \frac{\pi_\phi(\tau)}{\pi_{\mathrm{prior}}(\tau)} - b\right) \nabla_\phi \log \pi_\phi(\tau)\right], \tag{25}$$

where b is a control variate (baseline), which is often estimated using an average over the batch of trajectories (see Algorithm 2). Similarly, recall that the (on-policy, *i.e.*, where $\pi_b \equiv \pi_\phi$) RTB loss in (6) can be written as

$$\mathcal{L}_{\mathrm{RTB}}(\phi, \psi) = \frac{1}{2}\mathbb{E}_{\pi_\phi}\left[\left(\log \frac{\pi_{\mathrm{prior}}(\tau)}{Z_\psi \pi_\phi(\tau)} - \frac{\mathcal{E}(s_T)}{\alpha}\right)^2\right] \tag{26}$$

where we use the notation π_ϕ instead of P_ϕ to match the policy in KL-regularized RL. Taking the gradient of $\mathcal{L}_{\mathrm{RTB}}$ wrt. ϕ, and ignoring differentiation through the policy over which we take the expectation (which is standard in the GFlowNet literature [Malkin et al., 2023]), we get

$$\begin{aligned}
\nabla_\phi \mathcal{L}_{\mathrm{RTB}}(\phi, \psi) &= \frac{1}{2}\mathbb{E}_{\pi_\phi}[\nabla_\phi \Delta^2_{\mathrm{RTB}}(\tau; \phi, \psi)] && (27)\\
&= \mathbb{E}_{\pi_\phi}\left[\left(\frac{\mathcal{E}(s_T)}{\alpha} + \log \frac{\pi_\phi(\tau)}{\pi_{\mathrm{prior}}(\tau)} + \log Z_\psi\right) \nabla_\phi \log \pi_\phi(\tau)\right] && (28)\\
&= \frac{1}{\alpha}\nabla_\phi \mathcal{L}_{\mathrm{RL}}(\phi), && (29)
\end{aligned}$$

If the baseline $b = \alpha \log Z_\psi$. Therefore both methods are equivalent, only differing their control variate: REINFORCE with KL regularization often use a local baseline estimated using the current batch of trajectories, whereas Z_ψ in on-policy RTB acts as a *global* baseline. This result is similar to the one derived by Malkin et al. [2023].