REGULARIZED OFFLINE GFLOWNETS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose the regularized offline generative flow networks (RO-GFlowNets) that does not rely on online sampling. Since offline datasets usually cannot cover the entire state space, traditional GFlowNets cannot accurately predict the action sampling probability for each state. To address this problem, RO-GFlowNet aims to minimize the flow matching loss while regularizing the distribution distance of policy and offline datasets. Experimental results show that RO-GFlowNets perform well on offline datasets.

1 INTRODUCTION

Generative Flow Networks (GFlowNets) Bengio et al. (2021b) aim to complement reinforcement learning (RL) exploration ability with reward-proportional sampling and have been widely used in molecular generation Bengio et al. (2021a); Malkin et al. (2022), causal discovery Li et al. (2022); Deleu et al. (2022), discrete probability modeling Zhang et al. (2022) and graph neural networks Li et al. (2023a). However, these methods typically rely on online sampling to stabilize the policy optimization process of GFlowNets. Although GFlowNets' objective function allows for offline training Bengio et al. (2021a), we found that GFlowNets often struggle to generalize to out-of-distribution (OOD) actions when the data is non-uniform, leading to slower convergence speeds.

The problem of OOD generalization is not unique to GFlowNets, and is also a challenge in the field of RL Levine et al. (2020). Some offline RL techniques, such as behavior regularization Wu et al. (2019); Kumar et al. (2020); Agarwal et al. (2020) and model-based policy optimization algorithms Yu et al. (2021); Wang et al. (2021); Yan et al. (2022), have achieved impressive performance using offline datasets. However, the issue of OOD generalization in GFlowNets remains unresolved. In this paper, we propose the Regularized Offline GFlowNets (RO-GFlowNets) algorithm to achieve OOD generalization capability on offline datasets. We utilize the offline action probability density to constrain the action distribution generated by the sampling policy, while optimizing the flow matching loss function. This approach allows the model to increase the penalty on OOD state-action flow functions, thereby accelerating the convergence rate of the model.

2 RO-GFLOWNETS ALGORITHM

Define $s' = T(s,a)$ and $F(s)$ as state transition and the total flow through s, where a is the action. The edge/action flow $F(s,a) = F(s \to s')$ is defined as the flow through an edge $s \to s'$. The training process of vanilla GFlowNets requires summing parent and child flows through nodes (states), and can be optimized by the flow consistency equation: $\sum_{s,a:T(s,a)=s'} F(s,a) = r(s') + \sum_{a' \in \mathcal{A}(s')} F(s',a')$, where $r(s')$ and $\mathcal{A}(s')$ are the reward and the feasible action set $\mathcal{A}(s')$ of state s'. Consider an offline dataset $\{s,a,r,s'\} \in \mathcal{D}$ sampled from the exploration strategy π_β. Since the rewards of OOD states and actions are not available in $\mathcal{D}$, the generalization performance of the learned policy is poor. Hence, we propose a constrained flow optimization loss:

$$\min_{\pi \in \Pi} \mathcal{L}(\theta) = \min_{\pi \in \Pi} \mathbb{E}_{\mathcal{D}} \left(\sum_{s,a:T(s,a)=s'} F_\theta(s,a) - r(s') - \sum_{a' \in \mathcal{A}(s')} F_\theta(s',a') \right)^2, \tag{1}$$

where $\pi(a|s)$ is defined as $\pi(a|s) = \frac{F(s,a)}{F(s)}$ and Π is the unknown feasible policy set. In practice, since the exploration policy π_β is unknown, we cannot directly solve equation 1.

To solve equation 1, we propose an approximation that constrains π to be Π with a differentiable constraint based on dual gradient descent, the optimization problem can be formulated as

$$\pi := \arg\min_{\pi} \mathcal{L}(\theta), \quad \text{s.t. } \mathbb{E}_{s\sim\mathcal{D}}[\text{Dist}(\mathcal{D}(s), \pi(\cdot|s))] \leq \varepsilon, \tag{2}$$

where $\mathcal{D}(s)$ is the action distribution of s in $\mathcal{D}$, ε is a small positive value and Dist is the distribution metric function, such as maximum mean discrepancy Gretton et al. (2012).

To constrain the learned policy, we additionally introduce a neural network G_ϕ to approximate the action distribution. For each training step t, RO-GFlowNets samples a batch of transition data $\{s_i, a_i, r_i, s'_i\}_{i=1}^N \in \mathcal{D}$ to train G_ϕ based on $\mathcal{L}(\phi) = \frac{1}{N}\sum_{i=1}^N H(G_\phi(\cdot|s_i), a_i)$, where $H(p,q) = -\mathbb{E}_p \log_2 q$ is the cross-entropy function. Then we train F_θ by the flow matching loss Bengio et al. (2021b); Li et al. (2023b) and transform the policy as $\pi(a|s) = F_\theta(s,a)\mathbb{I}(G_\phi(a|s) > \delta)/F(s)$ for sampling, where δ is the threshold. The overall algorithm is summarized in Algorithm 1.

Algorithm 1 Regularized Offline GFlowNets Algorithm

Input: dataset $\mathcal{D}$, mini-batch size N, threshold δ.
1: **for** iteration $t = 1, \cdots, T$ **do**
2: Sample mini-batch of transition data $\{s_i, a_i, r_i, s'_i\}_{i=1}^N \in \mathcal{D}$.
3: Update G_ϕ by minimizing the cross-entropy loss $\mathcal{L}(\phi)$.
4: Update F_θ by minimizing the flow-matching loss.
5: Update the policy π by $\pi(a|s) = F_\theta(s,a)\mathbb{I}(G_\phi(a|s) > \delta)/F(s)$.
6: **end for**
Output: Flow function F_θ, the sampling policy π.

3 Experiment Results

We compare the proposed RO-GFlowNets with vanilla GFlowNets (sampled directly from the offline dataset to obtain a near-optimal policy) and a proposed Model-based Offline GFlowNets (MO-GFlowNets) approach, which uses offline data to train a transition model to achieve online sampling and training. To measure the performance of these methods, we consider the empirical ℓ_1-error and mode found to demonstrate the superiority of the algorithm.

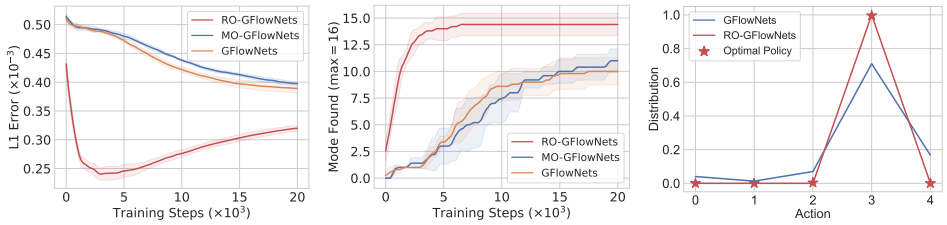

Figure 1: The performance of different algorithms on Hyper-Grid environment. **Left:** ℓ_1-Error (Lower is better) **Middle:** Mode Found (Higher is better) **Right:** Action Probability Distribution.

Figure 1 demonstrates the performance advantage of RO-GFlowNets, which can find the optimal sampling policy to sample all potential modes from the offline dataset (albeit with a little overfitting), while other algorithms fail. Online sampling training in MO-GFlowNets does not bring gains. This is because the estimated transition model is prone to errors, affecting the optimization of the policy. Figure 1-Right shows the action probability distribution of different algorithms, RO-GFlowNets can fit the optimal policy, which verifies its effectiveness.

4 Conclusion & Future Work

We propose an offline training algorithm for GFlowNets, which can avoid the problem of OOD state-action leading to unstable training. Experimental results show that RO-GFlowNets can well learn a sampling policy proportional to the reward function from offline datasets. Future work will be to extend the proposed algorithm to continuous space tasks.

REFERENCES

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in Neural Information Processing Systems*, 34:27381–27394, 2021a.

Yoshua Bengio, Tristan Deleu, Edward J Hu, Salem Lahlou, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *arXiv preprint arXiv:2111.09266*, 2021b.

Tristan Deleu, António Góis, Chris Emezue, Mansi Rankawat, Simon Lacoste-Julien, Stefan Bauer, and Yoshua Bengio. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pp. 518–528. PMLR, 2022.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Wenqian Li, Yinchuan Li, Shengyu Zhu, Yunfeng Shao, Jianye Hao, and Yan Pang. Gflowcausal: Generative flow networks for causal discovery. *arXiv preprint arXiv:2210.08185*, 2022.

Wenqian Li, Yinchuan Li, Zhigang Li, Jianye HAO, and Yan Pang. Dag matters! gflownets enhanced explainer for graph neural networks. In *International conference on learning representations*, 2023a.

Yinchuan Li, Shuang Luo, Haozhi Wang, and Jianye Hao. Cflownets: Continuous control with generative flow networks. In *International conference on learning representations*, 2023b.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *arXiv preprint arXiv:2201.13259*, 2022.

Jianhao Wang, Wenzhe Li, Haozhe Jiang, Guangxiang Zhu, Siyuan Li, and Chongjie Zhang. Offline reinforcement learning with reverse model-based imagination. *Advances in Neural Information Processing Systems*, 34:29420–29432, 2021.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Yuling Yan, Gen Li, Yuxin Chen, and Jianqing Fan. Model-based reinforcement learning is minimax-optimal for offline zero-sum markov games. *arXiv preprint arXiv:2206.04044*, 2022.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021.

Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. Generative flow networks for discrete probabilistic modeling. In *International Conference on Machine Learning*, pp. 26412–26428. PMLR, 2022.

A APPENDIX

We first present the experiment details. In the Hyper-Grid environment, the states are the cells of a N-dimensional hypercubic grid of side length H. The agent starts from the initialization point $x = (0, 0, \cdots)$, and can only increase coordinate i with action a_i. In addition, each agent has a stop action. When the agent chooses the stop action or reaches the maximum H of the episode length, the entire system resets for the next round of sampling. The reward function is designed as

$$r(x) = r_0 + r_1 \prod_i \mathbb{I}\left(0.25 < |x_i/H - 0.5|\right) + r_2 \prod_i \mathbb{I}\left(0.3 < |x_i/H - 0.5| < 0.4\right),$$

where $0 < r_0 \ll r_1 < r_2$. In our experiment, the dimension N and length H of Hyper-Grid are set as 4 and 8, respectively. The reward terms r_0, r_1 and r_2 are set as 1e-5, 0.5 and 2, respectively. F_θ and G_ϕ are optimized by the Adam optimizer with a learning rate 1e-5.

Figure 2 demonstrates the performance against different thresholds δ. We can see that proper thresholding can achieve good performance on both ℓ_1-error and mode found criteria. In addition, a higher threshold will reduce performance, because the output distribution of G_ϕ will be more uniform at this time, so a higher threshold will make some actions difficult to be sampled.

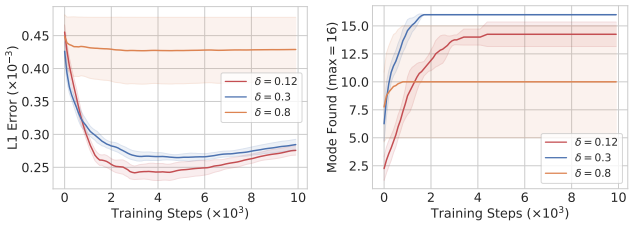

Figure 2: Performance of RO-GFlowNets against different thresholds.

We evaluate the proposed RO-GFlowNets algorithm on different offline datasets, including the random dataset and random-optimal dataset. The random dataset is sampled from a random policy, and the random-optimal dataset is obtained by using a mixture of optimal and random policies. Figure 3 shows the comparison results on the random and random-optimal datasets. Since high-reward states are difficult to cover with random datasets, all algorithms find only a few modes. For the random-optimal dataset, the performance of RO-GFlowNets is clearly ahead of other algorithms.

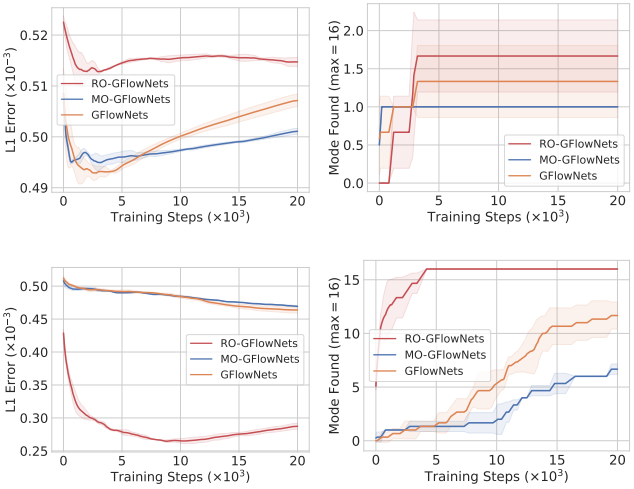

Figure 3: Comparison results on the different datasets. **Upper:** random dataset. **Bottom:** random-optimal dataset.