Order-Preserving GFlowNets

Yihang Chen [1] Lukas Mauch [2]

[1] EPFL [2] Sony Europe

Corresponding to yihang.chen@epfl.ch.

Published as a conference paper at ICLR 2024.

GFlowNets for Optimization

- Generative Flow Networks (GFlowNets, Bengio et al. [2021]) have been introduced as a method to sample a diverse set of candidates with probabilities proportional to a given reward.

Weakness of GFlowNets

- GFlowNets require an explicit formulation of a scalar reward $R(x)$ that measures the global quality of an object x. In the multi-objective optimization where $D > 1$, GFlowNets cannot be directly applied and $\boldsymbol{u}(x)$ has to be scalarized in prior [Jain et al., 2023; Roy et al., 2023].
- To prioritize the identification of candidates with high scalar $u(x)$ value, GFlowNets typically operate on the exponentially scaled reward $R(x) = (u(x))^{\beta}$. However, the optimal β balancing the exploration-exploitation is generally unknown.
- The exact computation of $u(x)$ might be costly, but the comparison the ordering of $u(x)$ and $u(x')$ may be more efficient.

Problem Statement

Problem Statement

We want to maximize a set of D objectives over $\mathcal{X}$, $\boldsymbol{u}(x) \in \mathbb{R}^D$. We define the the *Pareto dominance* on vectors $\boldsymbol{u}, \boldsymbol{u'} \in \mathbb{R}^D$, such that $\boldsymbol{u} \preceq \boldsymbol{u'} \Leftrightarrow \forall k, u_k \leq u'_k$. We remark that $\preceq$ induces a total order on $\mathcal{X}$ for $D = 1$, and a partial order for $D > 1$.

- We want to learn an order-preserving reward $\widehat{R}(x)$, such that $\widehat{R}(x) \leq \widehat{R}(x') \leftrightarrow \boldsymbol{u}(x) \preceq \boldsymbol{u}(x')$.
- We also want $\widehat{R}(x)$ to be almost uniform in the early training stages, and to concentrate on non-dominated candidates in the later training stages.

Idea

To use relative rather explicit boundary conditions to train GFNs.

GFlowNet Notations

DAG

- A directed acyclic graph $G = (\mathcal{S}, \mathcal{A})$ with state space $\mathcal{S}$ and action space $\mathcal{A}$.
- Let $s_0 \in \mathcal{S}$ be the *initial state*, the only state with no incoming edges; and *terminal states* set $\mathcal{X}$ be the states with no outgoing edges.
- Trajectory: a sequence of transitions $\tau = (s_0 \rightarrow s_1 \rightarrow \ldots \rightarrow s_n)$ going from the initial state s_0 to a terminal state $s_n = x$

Markovian Flow

- A *trajectory flow* is a nonnegative function $F : \mathcal{T} \rightarrow \mathbb{R}_{\geq 0}$.
- For any state s, define the state flow $F(s) = \sum_{s \in \tau} F(\tau)$, and, for any edge $s \rightarrow s'$, the edge flow $F(s \rightarrow s') = \sum_{\tau = (\ldots \rightarrow s \rightarrow s' \rightarrow \ldots)} F(\tau)$.
- The forward transition P_F and backward transition probability are defined as $P_F(s'|s) := F(s \rightarrow s')/F(s), P_B(s|s') = F(s \rightarrow s')/F(s')$ for the consecutive state s, s'.
- To approximate a Markovian flow F on the graph G such that

$$F(x) = R(x) \quad \forall x \in \mathcal{X}. \tag{1}$$

Algorithm

- Consider the terminal state set $X \subset \mathcal{X}$.
- The labeling distribution $\mathbb{P}_y$, indicator function of the Pareto front of X.

$$\mathbb{P}_y(x|X) := \frac{\mathbf{1}[x \in \mathsf{Pareto}(X)]}{|\mathsf{Pareto}(X)|}.$$

- The reward $\widehat{R}(\cdot)$ also induces a conditional distribution on the sample set X,

$$\mathbb{P}(x|X, \widehat{R}) : = \frac{\widehat{R}(x)}{\sum_{x' \in X} \widehat{R}(x')}, \forall x \in X.$$

- Minimizing

$$\mathcal{L}_{\mathrm{OP}}(X; \widehat{R}) := \mathsf{KL}(\mathbb{P}_y(\cdot|X) \| \mathbb{P}(\cdot|X, \widehat{R})).$$

Example

- Let us consider Trajectory Balance in the single-objective maximization.
- In the single-objective maximization, let $X = (x, x')$, i.e., pairwise comparison.

$$\mathbb{P}_y(x|X) = \frac{\mathbf{1}(u(x) > u(x')) + \mathbf{1}(u(x) \geq u(x'))}{2},$$

$$\mathbb{P}(x|X, \widehat{R}) = \frac{\widehat{R}(x)}{\widehat{R}(x) + \widehat{R}(x')},$$

- For the trajectory balance objective, let the trajectory $\tau \to x$, we define

$$\widehat{R}_{\mathrm{TB}}(x;\theta) := Z_\theta \prod_{t=1}^{n} P_F(s_t|s_{t-1};\theta)/P_B(s_{t-1}|s_t;\theta).$$

- For the non-trajectory balance objectives, $\mathcal{L}_{\mathrm{OP}}(X; \widehat{R})$ can also be easily integrated.

Theory

Mutually different

For $\{x_i\}_{i=0}^{n} \in \mathcal{X}$, assume that $u(x_i) < u(x_j), 0 \leq i < j \leq n$. The order-preserving reward $\widehat{R}(x) \in [1/\gamma, 1]$ is defined by the reward function that minimizes the order-preserving loss for neighbouring pairs $\mathcal{L}_{\mathrm{OP-N}}$, i.e.,

$$\begin{aligned}\widehat{R}(\cdot) &:= \arg \min_{r, r(x) \in [1/\gamma, 1]} \mathcal{L}_{\mathrm{OP-N}}(\{x_i\}_{i=0}^{n}; r) \\ &:= \arg \min_{r, r(x) \in [1/\gamma, 1]} \sum_{i=1}^{n} \mathcal{L}_{\mathrm{OP}}(\{x_{i-1}, x_i\}; r).\end{aligned}$$

We have $\widehat{R}(x_i) = \gamma^{i/n-1}, 0 \leq i \leq n$, and $\mathcal{L}_{\mathrm{OP-N}}(\{x_i\}_{i=0}^{n}; \widehat{R}) = n \log(1 + 1/\gamma)$.

General case (informal)

For $\{x_i\}_{i=0}^{n} \in \mathcal{X}$, assume that $u(x_i) \leq u(x_j), 0 \leq i < j \leq n$. When γ is sufficiently large, there exists α_γ, β_γ, dependent on γ, such that $\widehat{R}(x_{i+1}) = \alpha_\gamma \widehat{R}(x_i)$ if $u(x_{i+1}) > u(x_i)$, and $\widehat{R}(x_{i+1}) = \beta_\gamma \widehat{R}(x_i)$ if $u(x_{i+1}) = u(x_i)$, for $0 \leq i \leq n-1$. Also, minimize the $\mathcal{L}_{\mathrm{OP-N}}$ qith a variable γ will drive $\gamma \to \infty, \alpha_\gamma \to \infty, \beta_\gamma \to 1$.

Single Objective Experiments: NAS

- *NATS-Bench [Dong et al., 2021]*. The NAS can be regarded as a sequence generation problem to generate x, where the reward of each sequence of operations is determined by the accuracy of the corresponding architecture.
- Let $u_T(x)$ is the test accuracy of x's corresponding architecture with the weights at the T-th epoch during its standard training pipeline. We want to maximize u_{200}, but using only u_{12} in training. Since u_{12} is much more computationally efficient.
- We plot the u_{12} and u_{200} value of those who have the highest u_{12} value observed in training so far. The x-axis is measured by the time to compute u_{12} in the training so far.

Single Objective Experiments: NAS

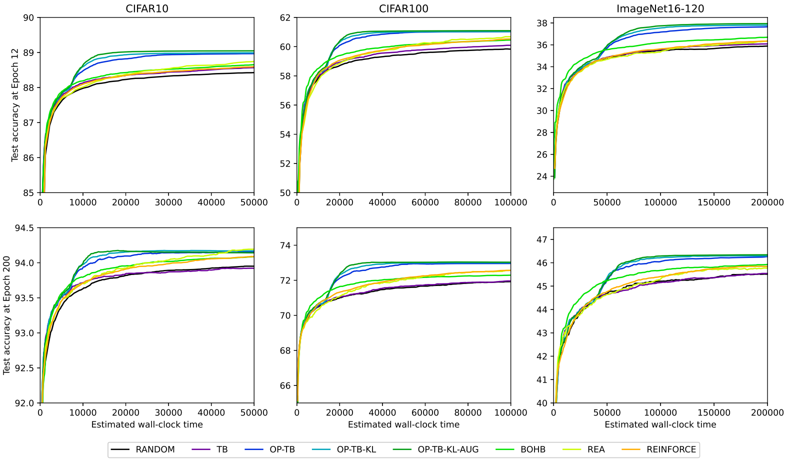
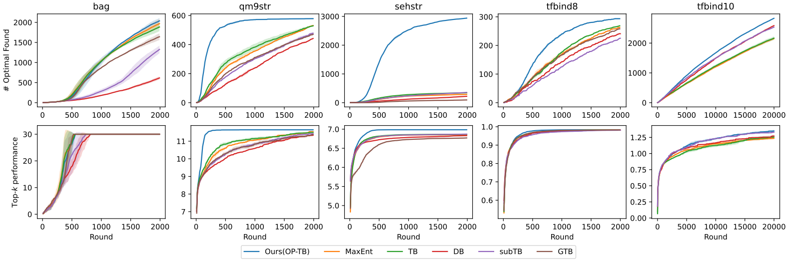


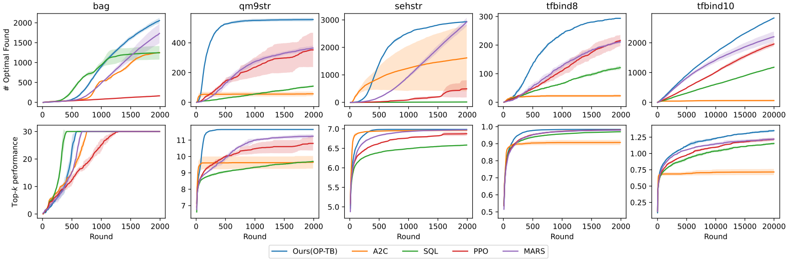
Figure: **Multi-trial training of a GFlowNet sampler**. Best test accuracy at epoch 12 and 200 of random baseline (Random), GFlowNet methods (TB, OP-TB, OP-TB-KL, OP-TB-KL-AUG), and other multi-trial algorithms (REA, BOHB, REINFORCE).

Single Objective Experiments: Molecular Generation

- We study various molecular designs environments [Bengio et al., 2021], including **Bag, TFBind8, TFBind10, QM9, sEH**.
- We consider previous GFN methods and reward-maximization methods as baselines. Previous GFN methods include TB, DB, subTB, maximum entropy (MaxEnt, Malkin et al. [2022]), and substructure-guided trajectory balance (GTB, Shen et al. [2023]). For reward-maximization methods, we consider a widely-used sampling-based method in the molecule domain, Markov Molecular Sampling (MARS), and RL-based methods, including actor-critic, Soft Q-Learning, and proximal policy optimization.

Single Objective Experiments: Molecular Generation





Multi Objective Experiments: HyperGrid

- We study two-dimensional HyperGrid, and consider five objectives.
- We compare the learned reward function of OP-GFNs and PC(Preference Conditioning)-GFNs. [Jain et al., 2023]

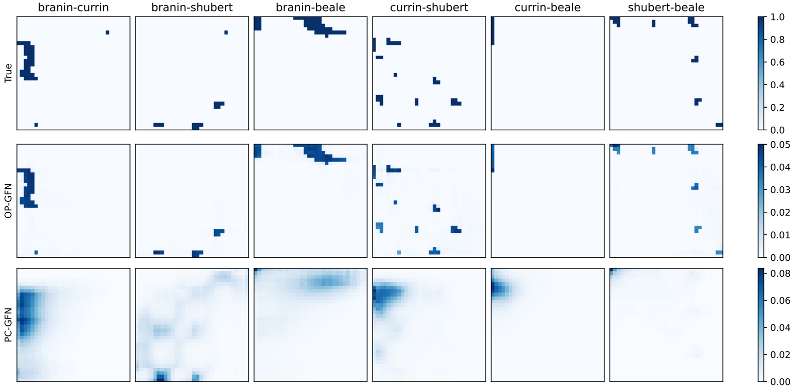
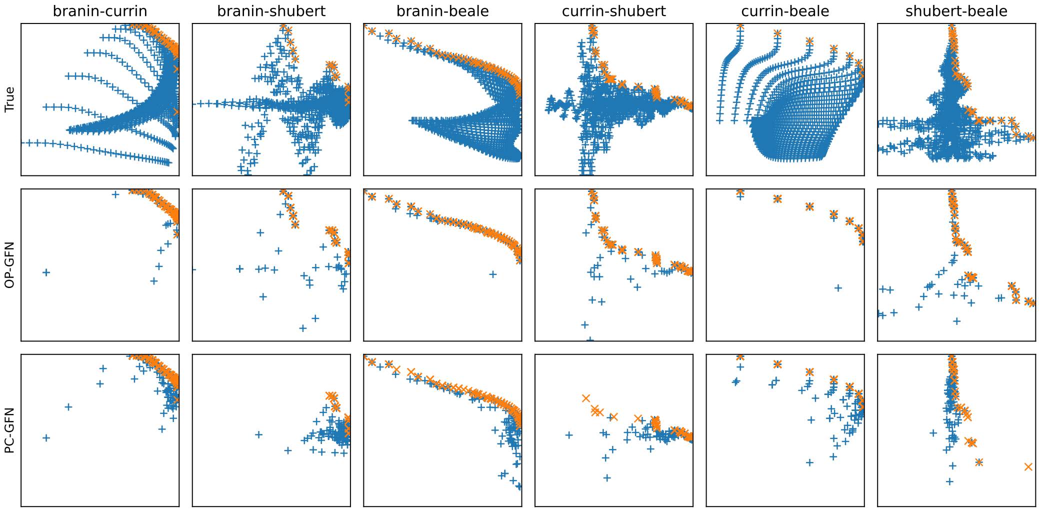


Figure: **Reward Landscape**: The first row of the above two figures contains all the states (blue) and the true Pareto front (orange).

Multi Objective Experiments: HyperGrid



Multi Objective Experiments: Molecular Generation

- Achieve comparable or better performance with PC-GFNs and GC (Goal Conditioning)-GFNs [Roy et al., 2023] without scalarization (no preference vectors, no temperature).

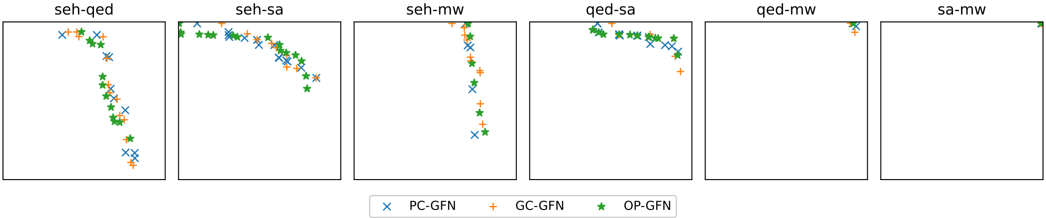


Figure: **Fragment-Based Molecule Generation**: We plot the estimated Pareto front of the generated samples in $[0,1]^2$. The x-, y-axis are the first, second objective in the title of respectively.

Future Work

- We currently resample from the replay buffer to ensure that the training of OP-GFNs does not collapse to part of the Pareto front. In the future, we hope that we can introduce more controllable guidance to ensure the diversity of the OP-GFNs' sampling.
- We want to find a specific task where the ordering is much easier to obtain than the exact reward value.

References I

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Neural Information Processing Systems (NeurIPS)*, 2021. (Cited on pages 2 and 10.)

Xuanyi Dong, Lu Liu, Katarzyna Musial, and Bogdan Gabrys. Nats-bench: Benchmarking nas algorithms for architecture topology and size. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3634–3646, 2021. (Cited on page 8.)

Moksh Jain, Sharath Chandra Raparthy, Alex Hernández-Garcia, Jarrid Rector-Brooks, Yoshua Bengio, Santiago Miret, and Emmanuel Bengio. Multi-objective gflownets. In *International Conference on Machine Learning*, pages 14631–14653. PMLR, 2023. (Cited on pages 2 and 12.)

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *arXiv preprint arXiv:2201.13259*, 2022. (Cited on page 10.)

Julien Roy, Pierre-Luc Bacon, Christopher Pal, and Emmanuel Bengio. Goal-conditioned gflownets for controllable multi-objective molecular design. *arXiv preprint arXiv:2306.04620*, 2023. (Cited on pages 2 and 14.)

Max W Shen, Emmanuel Bengio, Ehsan Hajiramezanali, Andreas Loukas, Kyunghyun Cho, and Tommaso Biancalani. Towards understanding and improving gflownet training. *arXiv preprint arXiv:2305.07170*, 2023. (Cited on page 10.)