Trajectory balance: Improved credit assignment in GFlowNets

Nikolay Malkin[1] Moksh Jain[1]

Emmanuel Bengio[2] Chen Sun[1]

Yoshua Bengio[1]

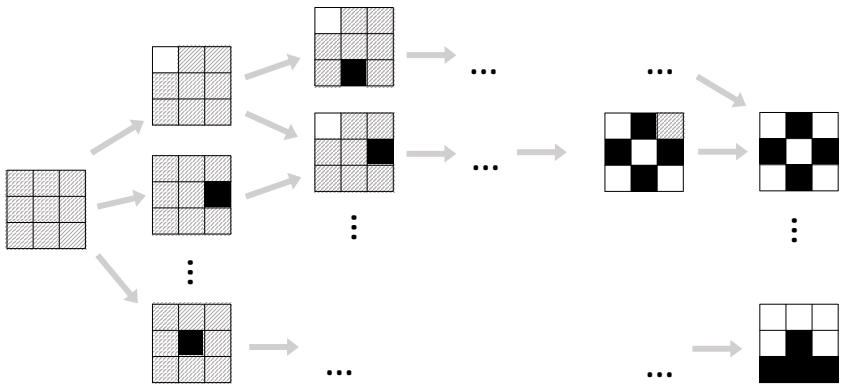
[1]Mila, Université de Montréal [2]Mila, McGill University, Recursion

NeurIPS 2022

Introduction to GFlowNets

[Bengio et al., NeurIPS'21]

- A generative flow network (GFlowNet) is a generative model that incrementally constructs objects x by sampling actions from a stochastic policy
- It is trained to make the probability of constructing x proportional to a given nonnegative reward function $R(x)$
- Sequential construction of x is convenient for sampling compositional objects
  - Many possible sequences of actions could lead to the same object

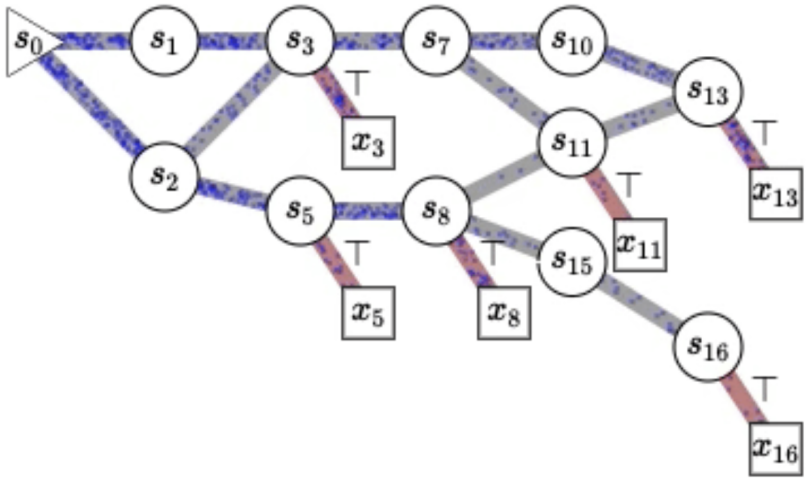


Introduction to GFlowNets

[Bengio et al., NeurIPS'21]

What 'flows'?

- Deterministic MDP: The states and actions form a directed acyclic graph; terminal (childless) states are complete objects
- A flow is a choice of nonnegative number for each action satisfying a 'flow in = flow out' condition at each state
- The edge flows $F(s \to s')$ are unnormalized action probabilities
  - The flow defines an action **policy**: $P_F(s'|s) \propto F(s \to s')$



Training GFlowNets by flow matching

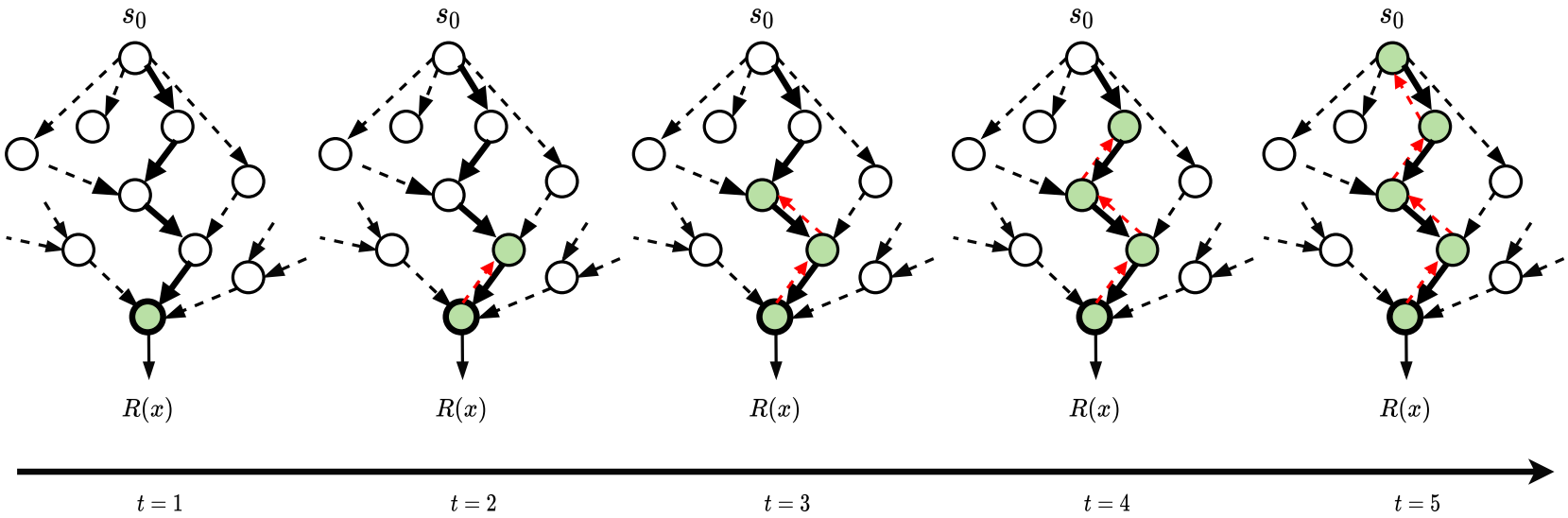
- **Flow matching (FM) loss:** Train a parametric estimate of the edge flow $F_\theta(s \to s')$ to enforce two conditions:
  - For terminal states x, in-flow equals $R(x)$
  - For other states, in-flow $=$ out-flow
- Both conditions can be converted to differentiable objective functions, e.g.:

$$\left(\log \frac{\text{in-flow at } s}{\text{out-flow at } s}\right)^2 = \left(\log \frac{\sum_{s':s'\to s} F_\theta(s' \to s)}{\sum_{s'':s\to s''} F_\theta(s \to s'')}\right)^2$$

- Can be minimized by SGD for states s seen along trajectories sampled on-policy (or use exploration tricks from RL)
- Variant: **detailed balance (DB) loss**

Problem: Slow credit assignment

Flow matching, akin to temporal difference learning, suffers from slow credit assignment along long action sequences



$$\mathcal{L}_{FM}(s) = \left(\log \frac{\sum_{(s'' \to s) \in \mathcal{A}} F_\theta(s'', s)}{\sum_{(s \to s') \in \mathcal{A}} F_\theta(s, s')} \right)^2$$

Trajectory balance (TB)

We propose a **trajectory-level** objective:

- Instead of parametrizing edge flows, learn three models:
  - The (forward) action policy $P_F(s'|s;\theta)$
  - A **backward policy** $P_B(s|s';\theta)$ (distribution over parents of any state s')
  - A scalar Z_θ, the estimated out-flow at the initial state s_0

Proposition: If the following is satisfied for all trajectories $s_0 \to s_1 \to \cdots \to s_n$, where s_n is terminal:

$$Z_\theta \prod_{i=0}^{n-1} P_F(s_{i+1}|s_i) = R(s_n) \prod_{i=0}^{n-1} P_B(s_i|s_{i+1}), \qquad (1)$$

then the likelihood that a trajectory sampled from the action policy P_F terminates at x is proportional to $R(x)$.

Trajectory balance (TB)

Proposition: If the following is satisfied for all trajectories $s_0 \to s_1 \to \cdots \to s_n$, where s_n is terminal:

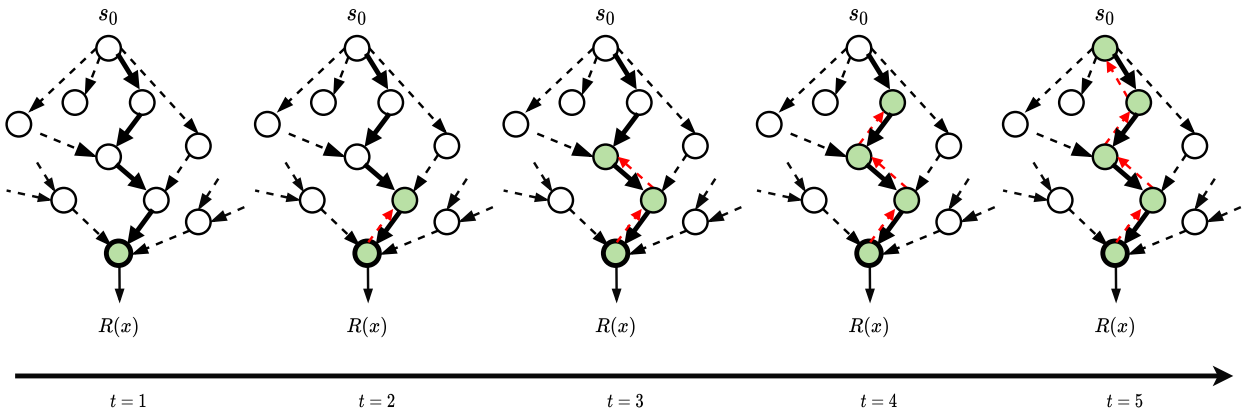
$$Z_\theta \prod_{i=0}^{n-1} P_F(s_{i+1}|s_i) = R(s_n) \prod_{i=0}^{n-1} P_B(s_i|s_{i+1}), \qquad (1)$$

then the likelihood that a trajectory sampled from the action policy P_F terminates at x is proportional to $R(x)$.

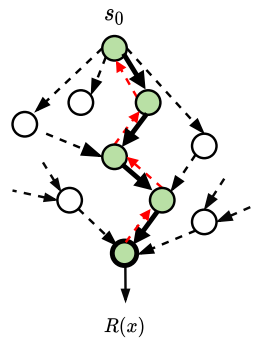
- **TB objective:** minimize square log-ratio of sides of (1)
- Trajectories for training can be sampled on-policy or from a more exploratory policy (tempering, replay buffer, . . .)

Why TB?

- Reward signal propagated along entire trajectory, unlike in FM
  - Limitation: TB introduces higher gradient variance
- On-policy TB gives an unbiased estimate of the gradient of a meaningful variational objective



$$\mathcal{L}_{FM}(s) = \left(\log \frac{\sum_{(s''\to s)\in\mathcal{A}} F_\theta(s'', s)}{\sum_{(s\to s')\in\mathcal{A}} F_\theta(s, s')}\right)^2$$



$$\mathcal{L}_{TB}(\tau) = \left(\log \frac{Z_\theta \prod_{t=1}^n P_F(s_t|s_{t-1};\theta)}{R(x)\prod_{t=1}^n P_B(s_{t-1}|s_t;\theta)}\right)^2.$$

Experiments

Experiments on four domains show the benefits of TB over other GFlowNet objectives and non-GFlowNet baselines:

- TB yields faster convergence on synthetic grid environment

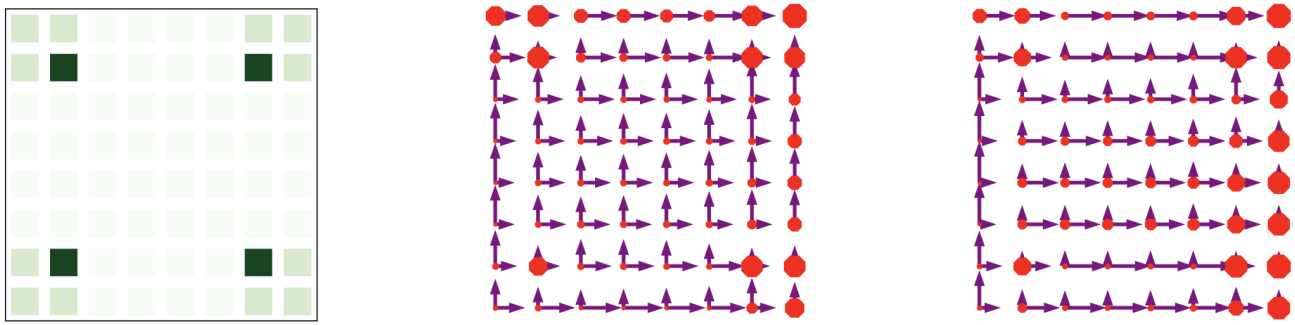


Figure 1: *Left:* The reward function on an 8×8 grid environment (§5.1) with $R_0 = 0.1$. *Centre and right:* Two forward action policies – with fixed uniform P_B and with a learned non-uniform P_B – that sample from this reward. The lengths of arrows pointing up and right from each state are proportional to the likelihoods of the corresponding actions under P_F, and the sizes of the red octagons are proportional to the termination action likelihoods.

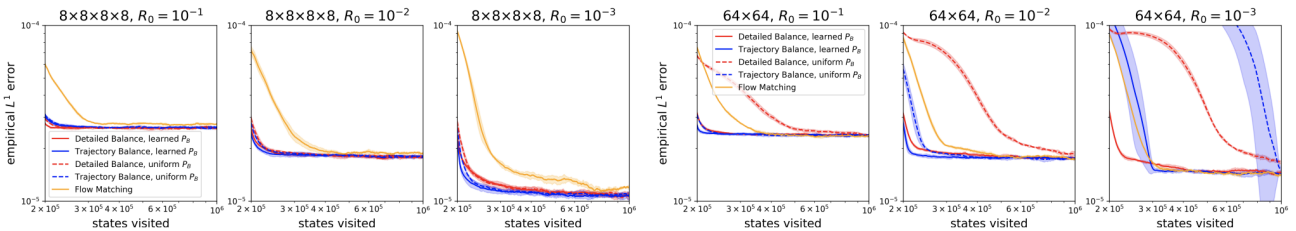


Figure 2: Empirical L^1 error between true and sampled state distributions on the grid environment with varying grid size and R_0. Mean and standard error over 5 seeds. The curves for PPO and MCMC baseline would lie outside the plot bounds.

Experiments

Experiments on four domains show the benefits of TB over other GFlowNet objectives and non-GFlowNet baselines:

- TB has a better fit to the target energy function on a molecule synthesis task

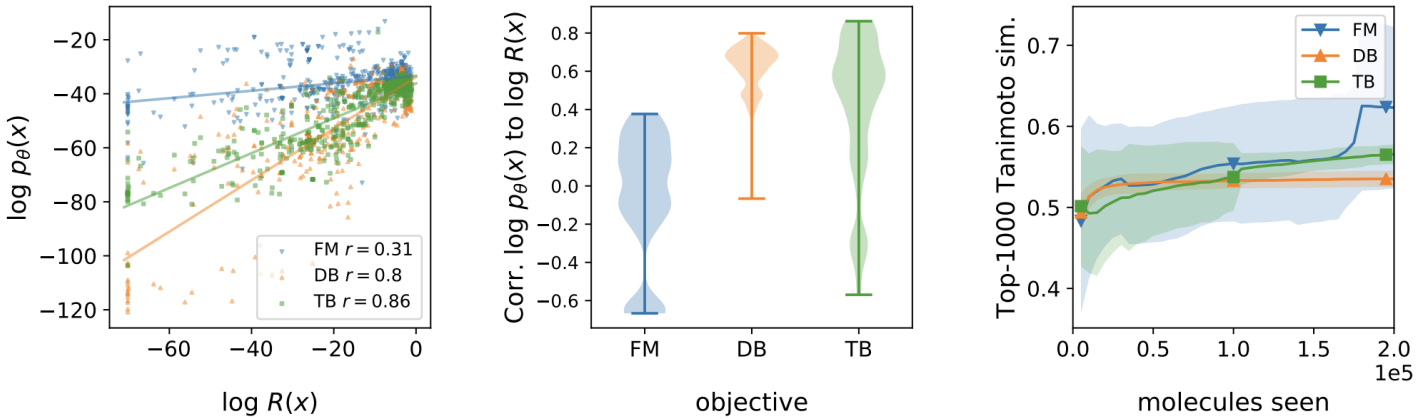


Figure 3: *Left, Centre:* Pearson correlations between rewards and sampling probability. $\log p_\theta(x)$ is the log-likelihood that a trajectory sampled from the learned policy $P_F(-|-;\theta)$ terminates at x. *Left:* Scatter plot on a test set of x's for the best hyperparameters of TB, FM, and DB. *Centre:* Violin plot of correlations for 16 hyperparameter settings and 3 seeds for each setting, showing TB being capable of fitting better. *Right:* Average pairwise Tanimoto similarity for the top 1000 samples generated by GFlowNets as training progresses. Lines are the average across runs, shaded regions the standard deviation. Models trained with TB have consistently lower similarity than those with FM, hence greater diversity. We hypothesize that the higher variance, in correlation and diversity, of TB relative to DB is related to high variance of the stochastic gradient; see [18].

Experiments

Experiments on four domains show the benefits of TB over other GFlowNet objectives and non-GFlowNet baselines:

- Models trained with TB find more high-reward states in synthetic and real-world sequence design problems

Table 1: Results on the AMP generation task.

	Top 100 Reward	Top 100 Diversity
GFN-$\mathcal{L}_{\text{TB}}$	$\mathbf{0.85} \pm 0.03$	$\mathbf{18.35} \pm 1.65$
GFN-$\mathcal{L}_{\text{FM}}$/$\mathcal{L}_{\text{DB}}$	0.78 ± 0.05	12.61 ± 1.32
SAC	0.80 ± 0.01	8.36 ± 1.44
AAC-ER	0.79 ± 0.02	7.32 ± 0.76
MCMC	0.75 ± 0.02	12.56 ± 1.45

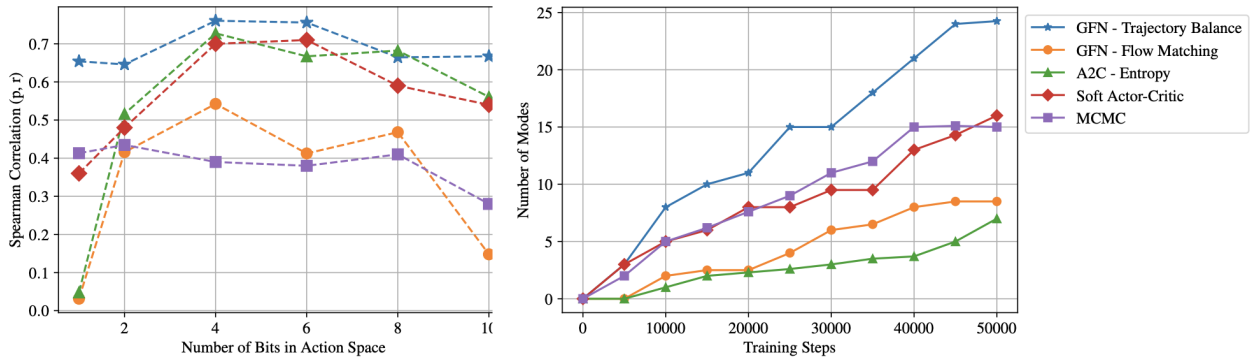


Figure 4: *Left:* Spearman correlation of the sampling probability under different learned policies and reward on a test set, plotted against the number of bits k in the symbols in V in the bit sequence generation task. GFlowNets trained with trajectory balance learn policies that have the highest correlation with the reward $R(x)$ and are robust to length and vocabulary size. *Right:* Number of modes discovered over the course of training on the bit sequence generation task with $k = 1$. GFlowNets trained with trajectory balance discover more modes faster.

Other applications

Since the paper appeared on arXiv, a few other works have used TB or its variants...

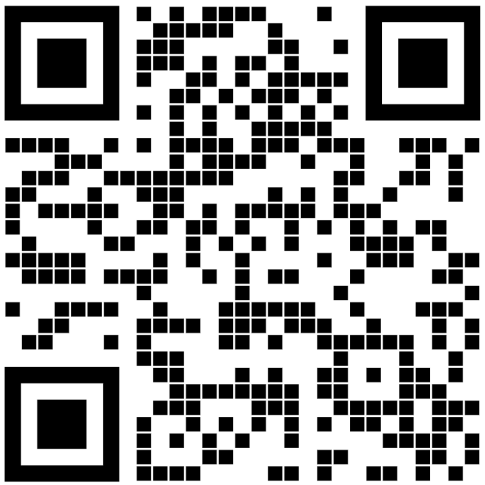
- Learning the reward function as an energy-based model
  GFlowNets for discrete probabilistic modeling [Zhang et al., ICML'22]
- Biological sequence design and active learning
  Biological sequence design with GFlowNets [Jain et al., ICML'22]
- Bayesian posterior over causal graphs
  Bayesian structure learning with GFlowNets [Deleu et al., UAI'22]
- Learning from **sub**trajectories to reduce gradient variance
  Learning GFlowNets from partial episodes [Madan et al., χ:2209.12782]
- Connections with variational methods
  GFlowNets and variational inference [Malkin et al., χ:2210.00580]
  Variational perspective on GFlowNets [Zimmermann et al., χ:2210.07992]

Conclusion

Trajectory balance (TB) yields faster and better training for GFlowNets than previously proposed losses.

TB discovers more modes of the reward function faster and is more robust to large action spaces and long action sequences.

TB has been applied to diverse problems; there is ongoing work on making TB more efficient and stable.



Thank you!

Paper: `arXiv:2201.13259`
Code: see paper for links