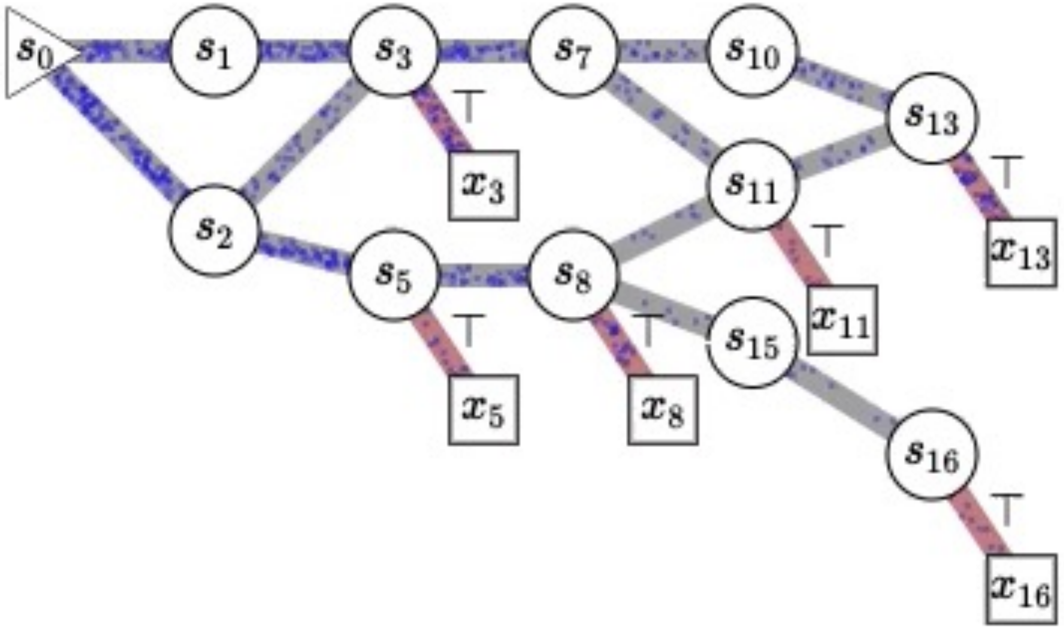
Generative Flow Networks for Discrete Probability Modeling

Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova,
Aaron Courville, Yoshua Bengio

GFlowNets Basics

- We want to generate some $x \in X$
- Trajectory "Flow" $F(\tau)$
- Flow runs through states, end in terminating states x
- Forward / backward policy $P_F(s'|s), P_B(s|s')$
- $\top$ = terminating action
- Set of terminal states = domain of X
- $P_T(x)$: terminating prob



A GFlowNet is uniquely determined by specifying either

1. Z (sum of all rewards) and P_F; or,
2. $R(x)$ and P_B

$$F(\tau) = Z \cdot P(\tau) = Z \cdot \prod_{t=0}^{n-1} \mathrm{P}_F(s_{t+1}|s_t) = R(s_n) \cdot \prod_{t=1}^{n} \mathrm{P}_B(s_{t-1}|s_t)$$

- Goal: learn a GFlowNet such that $P_T(x)$ is proportional to given reward $R(x)$

$$R(\mathbf{x}) = \sum_{\tau=(\mathbf{s}_0 \rightarrow \ldots \rightarrow \mathbf{s}_n), \mathbf{s}_n=\mathbf{x}} F(\tau)$$

- Training objective: for trajectory $\tau = (\mathbf{s}_0 \rightarrow \mathbf{s}_1 \rightarrow \ldots \rightarrow \ldots \rightarrow \mathbf{s}_n)$

$$\mathcal{L}_{\boldsymbol{\theta}}(\tau) = \left[\log \frac{Z_{\boldsymbol{\theta}} \prod_{t=0}^{n-1} P_F(\mathbf{s}_{t+1} \mid s_t; \boldsymbol{\theta})}{R(\mathbf{s}_n) \prod_{t=0}^{n-1} P_B(\mathbf{s}_t \mid \mathbf{s}_{t+1}; \boldsymbol{\theta})}\right]^2$$

(related derivation is omitted)

Discrete Probability Modeling

- Generate discrete data
- Start from all “void” states
- Action: assign a pixel value for one dimension
- Every trajectory has the same length

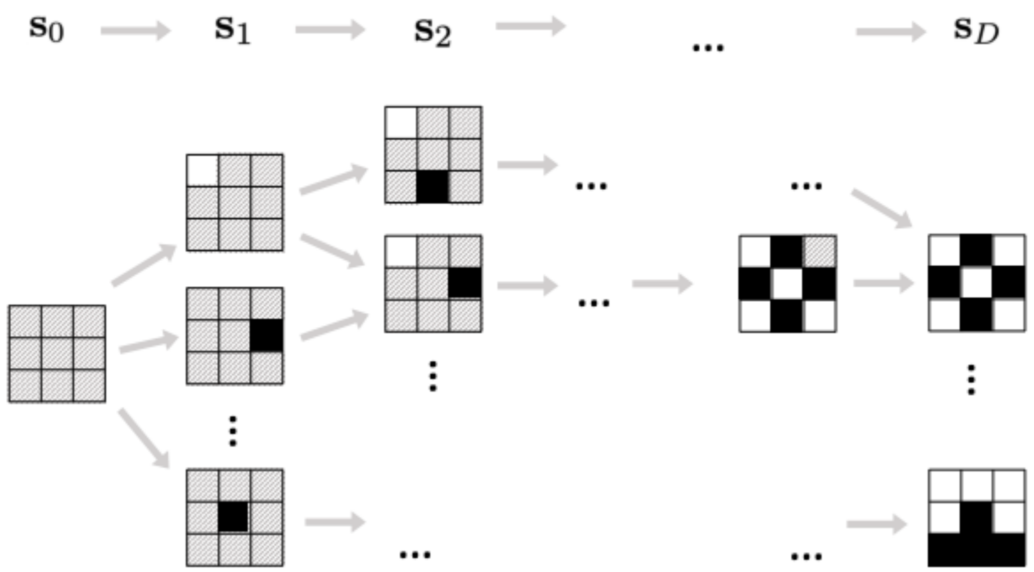


Figure 2. The state space $\mathcal{S}$ and the GFlowNet's forward modeling process in a 9-dimensional discrete data space. The states are the vertices of a DAG whose edges are the transitions – actions of painting a grey pixel into black (1) or white (0).

Discrete Probability Modeling

- Illustration of the forward and backward policy

- How should we obtain a useful reward from data?

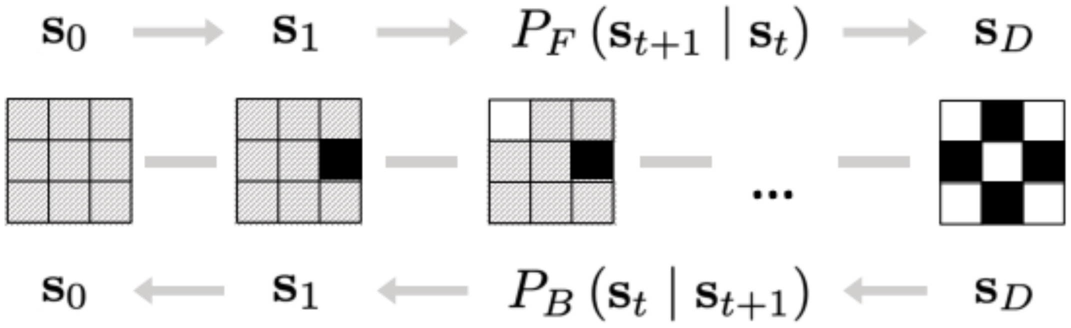


Figure 3. An illustration of the forward and backward GFlowNet policies in a 9-dimensional discrete space of the kind studied here. The forward policy transforms a state $\mathbf{s}_t$ into $\mathbf{s}_{t+1}$, while the backward policy does the opposite operation. We represent $0/1$ with black / white patches, and use grey patches to denote unspecified entries $\oslash$ in incomplete (non-terminal) states.

Energy-based Models (EBMs)

- Train an EBM as the reward $p_{\phi}(\mathbf{x}) = \frac{1}{Z_{\phi}} \exp(-\mathcal{E}_{\phi}(\mathbf{x}))$
- EBMs are usually trained with contrastive divergence (CD)

$$
\begin{aligned}
-\nabla_{\phi} \log p_{\phi}(\mathbf{x}) &= \nabla_{\phi} \mathcal{E}_{\phi}(\mathbf{x}) + \nabla_{\phi} \log Z_{\phi} \\
&= \nabla_{\phi} \mathcal{E}_{\phi}(\mathbf{x}) - \mathbb{E}_{\mathbf{x}' \sim p_{\phi}(\mathbf{x}')}[\nabla_{\phi} \mathcal{E}_{\phi}(\mathbf{x}')]
\end{aligned}
$$

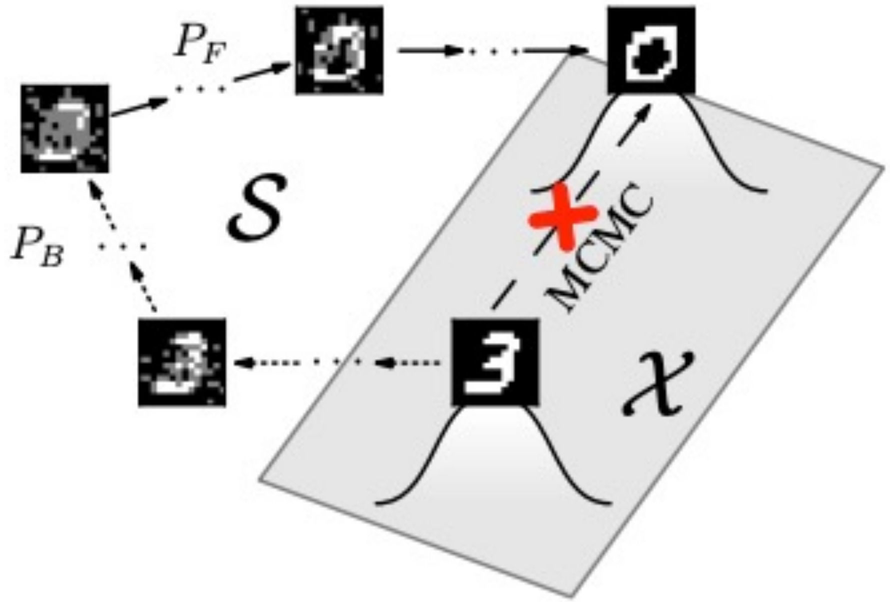
run MCMC chains for negative samples

- This MCMC could be computationally expensive, and suffer from slow mixing under multi-modal settings.

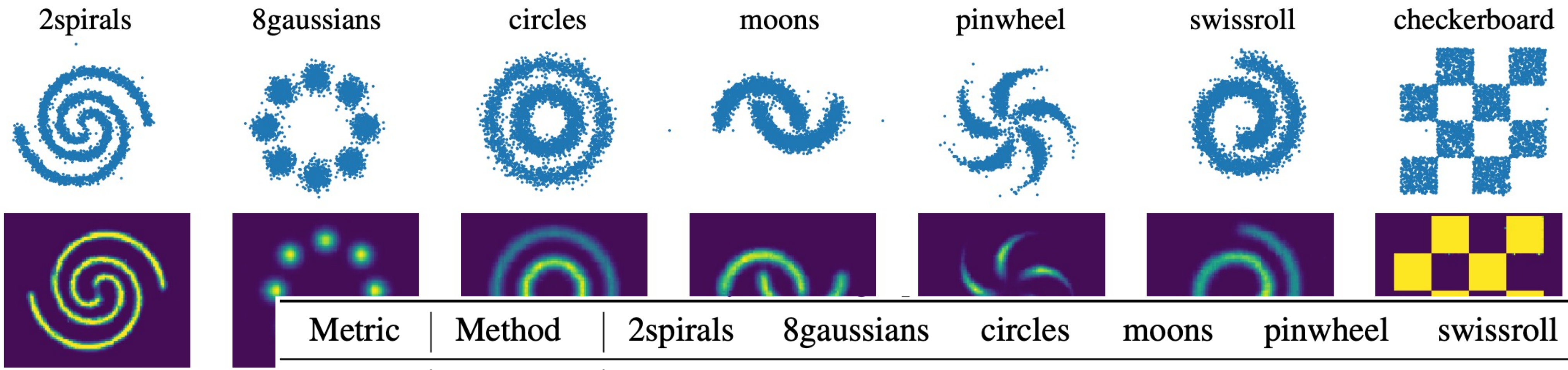
Energy-based GFlowNet

- We propose to jointly train an EBM and a GFlowNet
  - EBM serves as the reward for GFlowNet
  - GFlowNet provides negative samples for CD-like training

- (Detail) Back-forth proposal as transition kernel



Partial Results



Metric	Method	2spirals	8gaussians	circles	moons	pinwheel	swissroll	checkerboard
NLL↓	PCD	20.094	19.991	20.565	19.763	19.593	20.172	21.214
	ALOE	20.295	20.350	20.565	19.287	19.821	20.160	54.653
	ALOE+	20.062	19.984	20.570	19.743	19.576	20.170	21.142
	EB-GFN	**20.050**	**19.982**	**20.546**	**19.732**	**19.554**	**20.146**	**20.696**
MMD↓	PCD	2.160	0.954	0.188	0.962	0.505	1.382	2.831
	ALOE	21.926	107.320	0.497	26.894	39.091	0.471	61.562
	ALOE+	**0.149**	**0.078**	0.636	0.516	1.746	0.718	12.138
	EB-GFN	0.583	0.531	**0.305**	**0.121**	**0.492**	**0.274**	**1.206**

Thank you very much!