ADAPTIVE SAMPLING METHODS USING DIFFUSION MODELS

Marat Khusainov
Moscow Institute of Physics and Technology

Sergey Samsonov
HSE University

ABSTRACT

In this paper, we consider a novel approach to sampling from probability distributions using Generative Flow Networks (GFlowNets). GFlowNets are a class of amortized inference algorithms designed to address the challenge of generating samples proportional to given rewards in a Markov decision process (MDP). In this work, we adapt GFlowNets to the task of sampling from complex continuous probability distributions, leveraging their sequential decision-making framework to efficiently explore high-dimensional spaces.

1 Introduction

Sampling from complex multivariate distributions is a fundamental task which appears in many problems of machine learning (Izmailov et al. [2021], Hernández-Lobato and Adams [2015]), natural sciences (Bolhuis et al. [2002], Albergo et al. [2019]) and statistical inference (Neal [1998], Albergo et al. [2019]). A common strategy to tackle this problem is to resort to Markov chain Monte Carlo (MCMC) algorithms. These methods primarily guided by localized dynamics such as Hamiltonian Monte Carlo (Hoffman and Gelman [2011]) or Langevin dynamics (Roberts and Tweedie [1996], Roberts and Rosenthal [1998]). As well as being computationally expensive they commonly encounter difficulties in transitioning between modes. This challenge often results in long mixing times, correlation between samples or, in certain scenarios, complete failure to achieve equilibrium. This encourages employing amortized variational inference, which involves training parametric models to sample from the target distribution.

Diffusion models (Ho et al. [2020], Song et al. [2021]), a popular class of generative models providing state-of-the-art results in many domains. Gradually adding noise to the data via a diffusion process transforms the data distribution into a Gaussian distribution. Samples from this generative model are acquired by simulating an approximation of the reverse process of this diffusion, starting with Gaussian samples. Recent works (Zhang and Chen [2022], Vargas et al. [2023]) introduced sampling from unnormalized density functions as stochastic optimal control problems, where a diffusion model is trained as a stochastic process to generate the target samples.

In this work we consider recently introduced, generative flow networks (GFlowNets, Bengio et al. [2023]) being a family of amortized inference algorithms trained such that they sample through a sequence of decision-making process with probability proportional to a given reward function.

2 Problem statement

Suppose we have a probability density π on $\mathbb{R}^d$ of the form $\pi(x) = \frac{\mathcal{R}(x)}{Z}$, where $Z = \int_{\mathbb{R}^d} \mathcal{R}(x)\mathrm{d}x$ is an unknown normalizing constant. Define state space as

$$\mathcal{S} = \{(\mathbf{0}, 0) \cup \{(\mathbf{x}, t) : \mathbf{x} \in \mathbb{R}^d, t \in \{\Delta t, 2\Delta t, \ldots, 1\}\}$$

where $(\mathbf{x}, t)$ representing that the sampling agent is at position $\mathbf{x}$ at time t. Define a forward policy $\mathcal{P}_F$ that induces a distribution over complete trajectories $\tau = (\mathbf{x}_0 \to \mathbf{x}_{\Delta t} \to \cdots \to \mathbf{x}_1)$

The learning problem is to find the parameters θ, which may be the weights of neural network specifying a policy $\mathcal{P}_F$ whose terminating density $\mathcal{P}_F^{\top}$ is equal to target distribution π, i.e.,

$$\mathcal{P}_F^{\top}\left(\mathbf{x}_1;\theta\right)=\frac{\mathcal{R}\left(\mathbf{x}_1\right)}{Z}\quad,\forall \mathbf{x}_1\in\mathbb{R}^d. \tag{1}$$

3 Method

Diffusion models assume generative process given by a neural stochastic differential equation

$$\mathbf{x}_{t+\Delta t}=\mathbf{x}_t+u\left(\mathbf{x}_t,t;\theta\right)\Delta t+g\left(\mathbf{x}_t,t;\theta\right)\sqrt{\Delta t}\mathbf{z}_t,\quad \mathbf{z}_t\sim\mathcal{N}\left(\mathbf{0},\mathbf{I}_d\right). \tag{2}$$

Here initial state $\mathbf{x}_0$ follows a fixed tractable distribution μ_0, which could be chosen as a mass point or Gaussian. The transition density from states $\mathbf{x}_t$ to $\mathbf{x}_{t+\Delta t}$ in terms of forward policy can explicitly be written as

$$\mathcal{P}_F\left(\mathbf{x}_{t+\Delta t}\mid\mathbf{x}_t,\theta\right)=\mathcal{N}\left(\mathbf{x}_{t+\Delta t};\mathbf{x}_t+u\left(\mathbf{x}_t,t;\theta\right)\Delta t,g\left(\mathbf{x}_t,t;\theta\right)\Delta t\mathbf{I}_d\right). \tag{3}$$

A policy $\mathcal{P}_F$ generates a distribution across entire sequences of trajectories τ via

$$\mathcal{P}_F(\tau;\theta)=\prod_{i=0}^{T-1}\mathcal{P}_F\left(\mathbf{x}_{(i+1)\Delta t}\mid\mathbf{x}_{i\Delta t};\theta\right). \tag{4}$$

A backward policy $\mathcal{P}_B\left(\mathbf{x}_{t-\Delta t}\mid\mathbf{x}_t,\psi\right)$ is a collection of conditional probability densities, which represent the transition density from states $\mathbf{x}_{t-\Delta t}$ to $\mathbf{x}_t$. Here ψ also represent the parameters, which may be the weights of neural network. A policy $\mathcal{P}_B$ generates a distribution across entire sequences of trajectories τ via

$$\mathcal{P}_B\left(\tau\mid\mathbf{x}_1;\psi\right)=\prod_{i=1}^{T}\mathcal{P}_B\left(\mathbf{x}_{(i-1)\Delta t}\mid\mathbf{x}_{i\Delta t};\psi\right).$$

In Lahlou et al. [2023] authors showed that forward policy $\mathcal{P}_F$ samples from target distribution π if and only if there exists a backward policy $\mathcal{P}_B$ and scalar Z_θ such that the trajectory balance (TB) conditions are satisfied for every complete trajectory τ:

$$Z_\theta\mathcal{P}_F(\tau;\theta)=\pi(x_1)\mathcal{P}_B\left(\tau\mid\mathbf{x}_1;\psi\right). \tag{5}$$

If this equation holds then Z_θ represents the true partition function Z. In practise trajectory balance condition leads to the trajectory balance objective, which we want to minimize, w.r.t. parameters θ and ψ:

$$\mathcal{L}_{\mathrm{TB}}(\tau;\theta,\psi)=\left(\log\frac{Z_\theta\mathcal{P}_F(\tau;\theta)}{\pi\left(\mathbf{x}_1\right)\mathcal{P}_B\left(\tau\mid\mathbf{x}_1;\psi\right)}\right)^2. \tag{6}$$

Next goal is to propose new objective for GFlowNets which could take early stopping condition into account, when forward policy $\mathcal{P}_F$ samples with some probability α_θ from the non-terminating state $i\Delta t$. This idea could led to prevent mode seeking behaviour. [TODO]

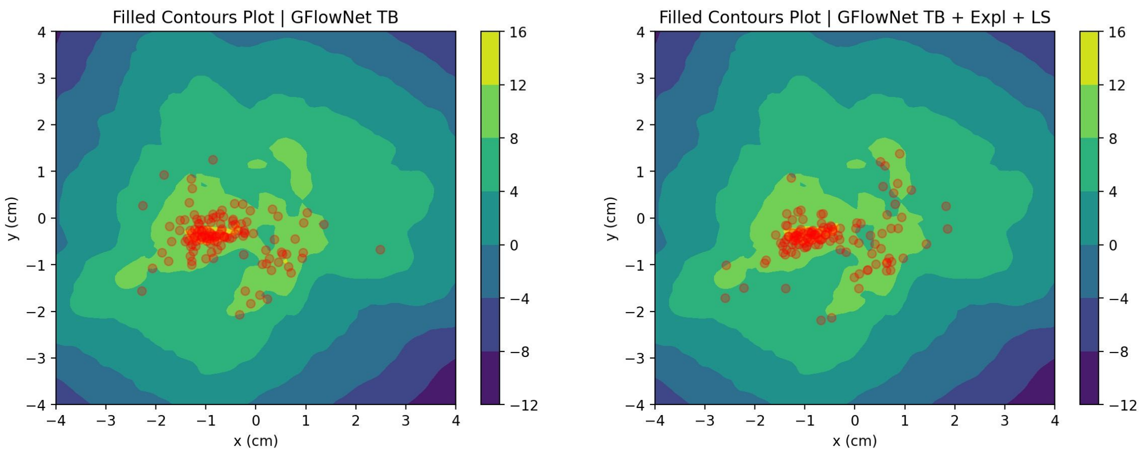
4 Experiments

In this section the experiments mainly based on code Zhang et al. [2024] with complex geometry distribution. Consider the Energy-Based Model (EBM) induced by the GAN in latent space. An EBM is defined by a Boltzmann distribution $\pi(\mathbf{x})=e^{-E(\mathbf{x})}/Z,\mathbf{x}\in\mathbb{R}^d$, where E(z) is the energy function and Z is the normalizing constant. Consider Wasserstein GAN (WGAN) trained on the MNIST dataset with latent space dimension $d=2$. The energy function is given by

$$E_W(\mathbf{x})=-\log p_0(\mathbf{x})-D(G(\mathbf{x}),\mathbf{x}\in\mathbb{R}^d. \tag{7}$$

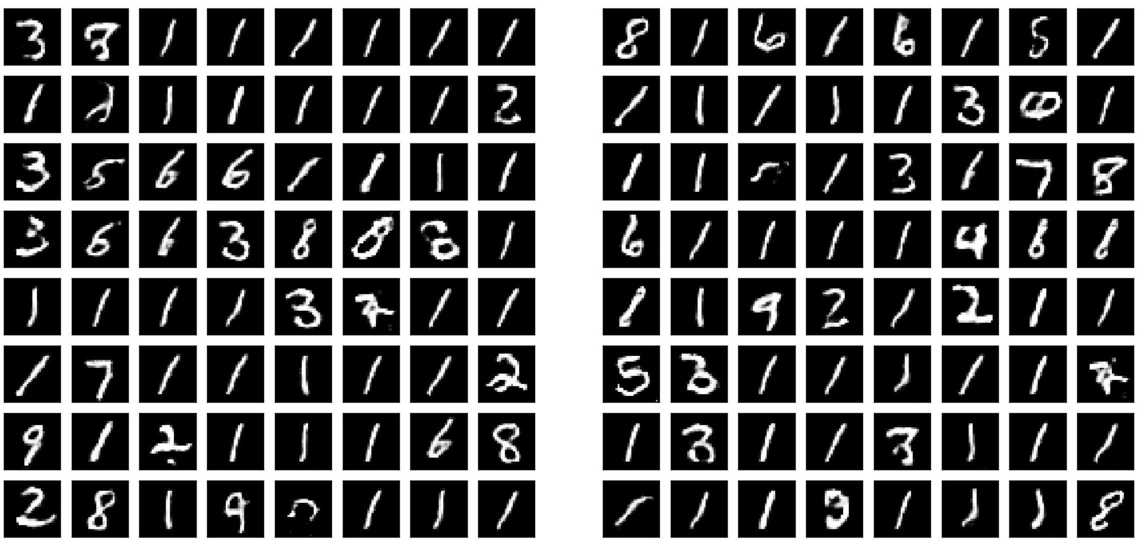
Here $p_0=\mathcal{N}\left(\mathbf{0},\mathbf{I}_d\right)$. In WGAN was used fully connected networks with 3 convolutional layers for discriminator and 3 linear $+3$ convolutional layers for generator.

To achieve more exploration efficiency local search (LS) methodology Zhang et al. [2022] was also considered. In detail, it works as follows: sample M candidates from the sampler: $\left\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(M)}\right\}\sim\mathcal{P}_F^{\top}(\cdot)$. Deploy parallel Langevin dynamics sampling across M chains, with the initial states of the Markov chain being $\left\{\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(M)}\right\}$. This process involves executing K transitions, during which certain candidates may be rejected based on the Metropolis-Hastings acceptance rule. After the burn-in transitions, the accepted samples are stored in the local search buffer. Results of sampling proposed in Fig. 1 and Fig. 2 Visually, it can be seen that using local search strategy improved sampling making objects more diverse.



(a) TB (b) TB + LS

Figure 1: WGAN on MNIST latent space visualizations and GFlowNet samples



(a) TB (b) TB + LS

Figure 2: MNIST samples visualization.

References

Pavel Izmailov, Sharad Vikram, Matthew D. Hoffman, and Andrew Gordon Wilson. What are bayesian neural network posteriors really like?, 2021.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks, 2015.

Peter G. Bolhuis, David W. Chandler, Christoph Dellago, and Phillip L. Geissler. Transition path sampling: throwing ropes over rough mountain passes, in the dark. *Annual review of physical chemistry*, 53:291–318, 2002. URL `https://api.semanticscholar.org/CorpusID:16361128`.

Michael S Albergo, Gurtej Kanwar, and Phiala E. Shanahan. Flow-based generative models for markov chain monte carlo in lattice field theory. *ArXiv*, abs/1904.12072, 2019. URL `https://api.semanticscholar.org/CorpusID:139104868`.

Radford M. Neal. Annealed importance sampling, 1998.

Matthew D. Hoffman and Andrew Gelman. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo, 2011.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2:341–363, 1996. URL `https://api.semanticscholar.org/CorpusID:18787082`.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling of discrete approximations to langevin diffusions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60, 1998. URL `https://api.semanticscholar.org/CorpusID:5831882`.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021.

Qinsheng Zhang and Yongxin Chen. Path integral sampler: a stochastic control approach for sampling, 2022.

Francisco Vargas, Will Grathwohl, and Arnaud Doucet. Denoising diffusion samplers, 2023.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J. Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations, 2023.

Salem Lahlou, Tristan Deleu, Pablo Lemos, Dinghuai Zhang, Alexandra Volokhova, Alex Hernández-García, Léna Néhale Ezzine, Yoshua Bengio, and Nikolay Malkin. A theory of continuous generative flow networks, 2023.

Dinghuai Zhang, Ricky T. Q. Chen, Cheng-Hao Liu, Aaron Courville, and Yoshua Bengio. Diffusion generative flow samplers: Improving learning signals through partial trajectory optimization, 2024.

Dinghuai Zhang, Nikolay Malkin, Zhen Liu, Alexandra Volokhova, Aaron Courville, and Yoshua Bengio. Generative flow networks for discrete probabilistic modeling, 2022.