Diffusion Generative Flow Samplers: Improving Learning Signals Through Partial Trajectory Optimization (ICLR 2024)

Dinghuai Zhang*, Ricky T. Q. Chen, Cheng-Hao Liu, Aaron Courville & Yoshua Bengio

Thomas Mousseau

October 29, 2025

Overview

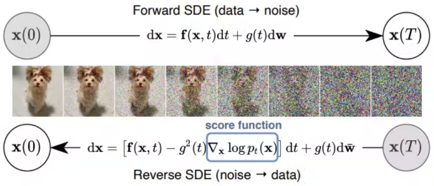
Generative Modeling

Task

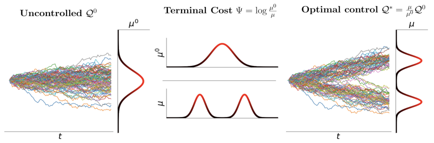
Sample from a complex (high-dimensional and multimodal) distribution D.

D can be given under the form of:

- A dataset of samples $\{x_i\}_{i=1}^N \sim D$ (e.g., images, text, audio)


Forward SDE (data → noise)
$\mathbf{x}(0)$
$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)dt + g(t)d\mathbf{w}$
$\mathbf{x}(T)$
score function
$\mathbf{x}(0)$
$d\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\boxed{\nabla_{\mathbf{x}} \log p_t(\mathbf{x})}\right] dt + g(t)d\bar{\mathbf{w}}$
$\mathbf{x}(T)$
Reverse SDE (noise → data)

- An unnormalized density $\mu(x)$ where D has density $\pi(x) \propto \mu(x)$ (e.g., energy-based models, physics/chemistry)



Sampling from Unnormalized Densities

Context. Sample from a D-dimensional target with unnormalized density $\mu(x)$ where $\mathbb{R}^D \to \mathbb{R}$.

$$\pi(x) = \frac{\mu(x)}{Z}, \qquad Z = \int_{\mathbb{R}^D} \mu(x)\, dx \text{ (unknown)}.$$

We assume we can evaluate $\mu(x)$, but we have no samples from π and do not know Z.

Goal. We seek a *sampler* (similar to MCMC/VI) that produces calibrated samples and, ideally, estimates Z *without* any dataset from π.

Chemistry (molecule conformers).

Molecules admit many 3D conformations scored by an energy $E(x)$ (lower is better). We set $\mu(x) = \exp(-E(x))$ so larger μ means more desirable. Because many chemical properties are unknown or costly, a well-calibrated sampler draws conformers in proportion to μ, yielding high-quality and diverse candidates for ranking, ensemble estimates, and realistic screening than chasing a single best pose.

Controlled and Reference Processes

Goal.

Learn f_θ so that the terminal marginal $Q(x_N)$ matches $\mu(x)$ and be proportional to $\pi(x)$ (no data, Z unknown).

Controlled forward transition (learned drift).

$$P_F(x_{n+1} \mid x_n) \;=\; \mathcal{N}\big(x_{n+1};\; x_n + h\, f_\theta(x_n, n),\; h\sigma^2 I\big)$$

Controlled process and marginals.

$$Q(x_{0:N}) \;=\; p_0^{\text{ref}}(x_0) \prod_{n=0}^{N-1} P_F(x_{n+1} \mid x_n)$$

$$p_n(x) = \int Q(x_{0:N})\, dx_{0:n-1} dx_{n+1:N}, \text{ no closed form.}$$

Uncontrolled/Reference forward transition.

$$P_F^{\text{ref}}(x_{n+1} \mid x_n) \;=\; \mathcal{N}\big(x_{n+1};\; x_n,\; h\sigma^2 I\big)$$

Uncontrolled/Reference process and marginals.

$$Q^{\text{ref}}(x_{0:N}) \;=\; p_0^{\text{ref}}(x_0) \prod_{n=0}^{N-1} P_F^{\text{ref}}(x_{n+1} \mid x_n),$$

$$p_n^{\text{ref}}(x) = \mathcal{N}\left(x; x_0, \Sigma_0 + nh\sigma^2 I\right), \text{ is closed form.}$$

Closed Form of Reference Marginal

Uncontrolled Sampling.

$$x_N = x_0 + \sqrt{h}\sigma \sum_{k=0}^{N-1} \varepsilon \quad \varepsilon \sim \mathcal{N}(0, I) \quad \text{since we are using Brownian motion.}$$

Gaussian Form (Reparametrization Trick).

$$x_N \sim \mathcal{N}(x_0, \Sigma_0 + Nh\sigma^2 I) \quad \text{where } \Sigma_0 \text{ is the initial covariance.}$$

Probability Density Function of Terminal Marginal Distribution.

$$p_N^{\text{ref}}(x_N) = \frac{1}{(2\pi)^{D/2}|\Sigma_0 + Nh\sigma^2 I|^{1/2}} \exp\left(-\frac{1}{2}(x_N - x_0)^\top (\Sigma_0 + Nh\sigma^2 I)^{-1}(x_N - x_0)\right).$$

Guiding the Controlled Process

Adding Terminal Tilt to the Uncontrolled Q^{ref} Process to Create the Target Process P

$$P(\tau) = Q^{\text{ref}}(x_{0:N-1}|x_N) \cdot \pi(x_N) = Q^{\text{ref}}(\tau) \cdot \frac{\pi(x_N)}{p_N^{\text{ref}}(x_N)} \propto Q^{\text{ref}}(\tau) \cdot \frac{\mu(x_N)}{p_N^{\text{ref}}(x_N)}.$$

$$P(x_N) = \int_{\text{subtrajectories}} \underbrace{Q^{\text{ref}}(x_{0:N-1}|x_N)\, dx_{0:N-1} \cdot p_N^{\text{ref}}(x_N)}_{Q^{\text{ref}}(x_{0:N})} \cdot \frac{\pi(x_N)}{p_N^{\text{ref}}(x_N)} = \pi(x_N).$$

Path space KL objective

Target path measure via terminal reweighting.

$$P(x_{0:N}) \propto \; Q^{\mathrm{ref}}(x_{0:N})\,\frac{\mu(x_N)}{p_N^{\mathrm{ref}}(x_N)} \qquad \Longrightarrow \qquad P(x_N) \propto \mu(x_N).$$

KL decomposition using Girsanov theorem (path space).

$$\mathrm{KL}(Q\|P) = \mathbb{E}_Q\left[\log\frac{Q}{P}\right] = \mathbb{E}_Q\left[\log\frac{Q}{Q^{\mathrm{ref}}}\right] + \mathbb{E}_Q\left[\log\frac{p_N^{\mathrm{ref}}(x_N)}{\mu(x_N)}\right] + \log Z.$$

Running cost.

$$\mathbb{E}_Q\left[\log\frac{Q}{Q^{\mathrm{ref}}}\right] = \mathbb{E}_Q\Big[\sum_{n=0}^{N-1}\frac{h}{2\sigma^2}\,\|f_\theta(x_n,n)\|^2\Big].$$

Terminal cost.

$$\mathbb{E}_Q\left[\log\frac{p_N^{\mathrm{ref}}(x_N)}{\mu(x_N)}\right] = \mathbb{E}_Q\big[\log p_N^{\mathrm{ref}}(x_N) - \log\mu(x_N)\big].$$

Credit Assignment Problem in SOC objective

SOC Discrete-time objective

$$\min_{f_\theta} \mathbb{E}_Q\left[\underbrace{\sum_{n=0}^{N-1} \frac{h}{2\sigma^2} \left\| f_\theta(x_n, n) \right\|^2}_{\text{Running cost}} + \underbrace{\log p_N^{\text{ref}}(x_N) - \log \mu(x_N)}_{\text{Terminal cost}} \right]$$

This objective is used on the seminar paper *Path Integral Sampler: Diffusion-based Sampling for Unnormalized Densities* **zhang2022** which presented the sampling from unnormalized densities as a stochastic optimal control problem.

SOC as a GFlowNet

Comparison Table: GFlowNet vs. SOC Framework

Concept	**GFlowNet**	**SOC**
Forward Process	Trajectory sampling on DAG	Controlled diffusion path
Forward Transition Probability	$P_F(s'\|s)$	$P_F(x_{n+1}\|x_n) = \mathcal{N}(x_{n+1}; x_n + hf_\theta(x_n), h\sigma^2 I)$
Backward Transition Probability	$P_B(s\|s')$	$P_B^{\text{ref}}(x_n\|x_{n+1})$ (known)
Reward Function	$R(x)$ (unnormalized)	$\mu(x)$ (unnormalized density)
Terminal Marginal Distribution	$P_T(x) \propto R(x)$	$Q(x_N) \propto \mu(x_N)$
Flow State	Flow $F(s)$ at states	Learned flow $F_n(x)$

Insight. Since SOC can be viewed as a GFlowNet, we can apply GFlowNet tools (detailed balance loss or subtrajectory balance) to solve the credit assignment problem.

Adressing Credit Assignment via Target Process

SOC Discrete-time objective

$$\min_{f_\theta} \mathbb{E}_Q\Big[\sum_{n=0}^{N-1} \tfrac{h}{2\sigma^2}\|f_\theta(x_n, n)\|^2 + \log p_N^{\text{ref}}(x_N) - \log \mu(x_N)\Big]$$

Could we write the target process differently to allow intermediate supervision?

Conditional Form. Since the reweighting only affects the terminal state, the joint can be written as:

$$P(x_{0:N}) = \pi(x_N) \prod_{n=0}^{N-1} Q_B^{\text{ref}}(x_n|x_{n+1}),$$

where $Q_B^{\text{ref}}(\cdot|\cdot)$ is the backward transition probability (derived from the target joint). This is tractable because Q_B^{ref} is known.

Rewriting the Target Process with Marginal

If We Had Access to $p_n(x_n)$ we could write the partial joint which would allow training on subtrajectories and thus have better credit assignment:

$$P(x_{0:n}) = p_n(x_n) \prod_{k=0}^{n-1} Q_B^{\text{ref}}(x_k|x_{k+1}),$$

But There's No Closed Form for $p_n(x_n)$. To calculate $p_n(x_n)$, we would need to compute the integral:

$$p_n(x_n) = \int_{\mathbb{R}^{(N-n)D}} \pi(x_N) \prod_{k=n}^{N-1} Q_B^{\text{ref}}(x_k|x_{k+1})\, dx_{n+1:N}.$$

$$p_n(x_n) \propto \int_{\mathbb{R}^{(N-n)D}} \mu(x_N) \prod_{k=n}^{N-1} Q_B^{\text{ref}}(x_k|x_{k+1})\, dx_{n+1:N}.$$

Why is $p_n(x_n)$ hard to compute?

To calculate this this high-dimensional integral (from n to N over D dimensions), we would need a solution like Monte Carlo quadratures which would be infeasible in practice, especially since training requires computing it repeatedly for subtrajectories as intermediate signals.

Detailed and SubTB Balance with Learned Flow

Detailed Balance Condition with Learned Flow

$$F_n(x_n;\phi)P_F(x_{n+1}|x_n;\theta) = F_{n+1}(x_{n+1};\phi)P_B(x_n|x_{n+1}).$$

Subtrajectory Balance Loss for Partial Trajectories

$$\ell_{\text{SubTB}}(x_{m:n};\theta,\phi) = \left(\log \frac{F_m(x_m;\phi)\prod_{k=m}^{n-1} P_F(x_{k+1}|x_k;\theta)}{F_n(x_n;\phi)\prod_{k=m}^{n-1} P_B(x_k|x_{k+1})}\right)^2$$

Using a mix of different subtrajectory lengths (m,n) allows better credit assignment thus more stable training of both the forward policy and the learned flow.

Overall Training Objective

Recall SubTB Loss

$$\ell_{\text{SubTB}}(x_{m:n};\theta,\phi) = \left(\log \frac{F_m(x_m;\phi)\prod_{k=m}^{n-1} P_F(x_{k+1}|x_k;\theta)}{F_n(x_n;\phi)\prod_{k=m}^{n-1} P_B(x_k|x_{k+1})}\right)^2$$

Diffusion Generative Flow Sampler (DGFS) Loss

$$L(\tau;\theta;\phi) = \frac{\sum_{0\le m<n\le N} \lambda^{n-m}\ell_{\text{SubTB}}(x_{m:n})}{\sum_{0\le m<n\le N}\lambda^{n-m}}, \quad \tau = (x_0,\ldots,x_N)$$

This combines signals from all subtrajectory lengths, reducing variance and improving credit assignment

- τ: Full trajectory $(x_0,\ldots,x_N)$.
- λ: $[0,1]$ to weight shorter subtrajectories more since they have lower variance.
- The denominator stabilizes the loss scale across different trajectory lengths.

Algorithm 1 DGFS Training

```
Require: $\mu(\cdot)$, $\bar{\sigma}$, $N$, $\lambda$, $B$, $\eta$
 1: Init $\theta = (\theta_f, \phi)$
 2: repeat
 3:     Sample trajectories:
 4:     for $b = 1$ to $B$ do
 5:         $\tau^{(b)} = (x_0^{(b)}, \dots, x_N^{(b)})$ under $x_{n+1} = x_n + hf_\theta(x_n, n) + \sqrt{h}\bar{\sigma}\varepsilon_n$, $\varepsilon_n \sim \mathcal{N}(0, I)$
 6:     end for
 7:     Build subtrajectories: $\mathcal{S}(\tau^{(b)})$ of $(m, n)$ with $0 \le m < n \le N$
 8:     Compute SubTB loss:
            $$\mathcal{L}(\tau^{(b)};\theta) = \frac{\sum_{(m,n)\in\mathcal{S}(\tau^{(b)})} \lambda^{n-m} \left[\log \frac{F_\phi(x_m)\prod_{l=m}^{n-1} P_F(x_{l+1}|x_l;\theta_f)}{F_\phi(x_n)\prod_{l=m}^{n-1} P_B^{\mathrm{ref}}(x_l|x_{l+1})}\right]^2}{\sum_{(m,n)\in\mathcal{S}(\tau^{(b)})} \lambda^{n-m}}$$
 9:     $g \leftarrow \nabla_\theta \frac{1}{B}\sum_{b=1}^{B} \mathcal{L}(\tau^{(b)};\theta)$
10:     $\theta \leftarrow \mathrm{Adam}(\theta, g, \eta)$
11: until convergence
```

Reduction of Gradient Variance

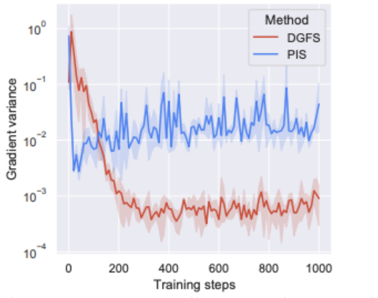


Figure: Gradient variance comparison between DGFS and PIS. DGFS shows significantly lower variance, leading to more stable training.

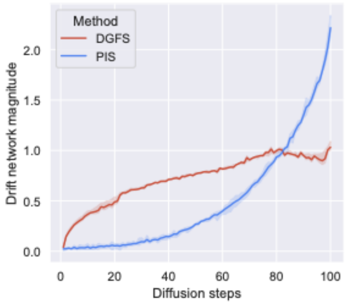


Figure: Drift network magnitude of DGFS and PIS.

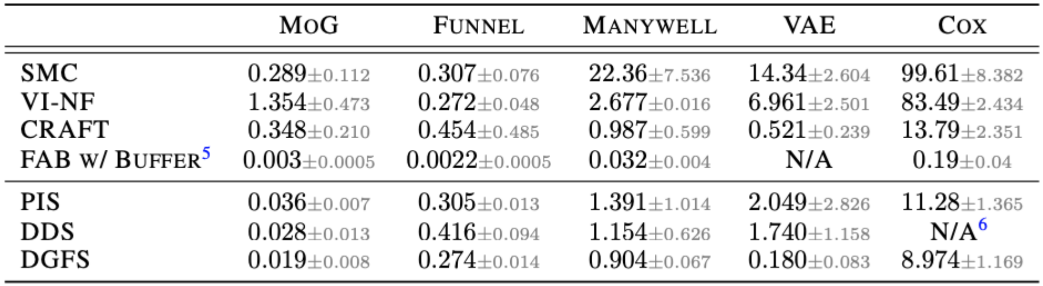
Partition Function Estimation

Partition Function Estimation

$$\log \sum_{\tau}^{\mathrm{B}} \exp(\log(P(\tau)) - \log(Q(\tau))) - \log \mathrm{B} \leq \log Z, \quad \tau \sim Q(\cdot;\theta).$$

The equation computes a lower bound on $\log Z$ via importance sampling: it samples trajectories τ from Q, reweights them by $\log(P(\tau)/Q(\tau))$, and uses log-sum-exp to approximate log of the normalized weight sum, which underestimates $\log Z$ due to Jensen's inequality.

Partition Function Estimation Results



	MoG	Funnel	Manywell	VAE	Cox
SMC	$0.289_{\pm 0.112}$	$0.307_{\pm 0.076}$	$22.36_{\pm 7.536}$	$14.34_{\pm 2.604}$	$99.61_{\pm 8.382}$
VI-NF	$1.354_{\pm 0.473}$	$0.272_{\pm 0.048}$	$2.677_{\pm 0.016}$	$6.961_{\pm 2.501}$	$83.49_{\pm 2.434}$
CRAFT	$0.348_{\pm 0.210}$	$0.454_{\pm 0.485}$	$0.987_{\pm 0.599}$	$0.521_{\pm 0.239}$	$13.79_{\pm 2.351}$
FAB w/ Buffer[5]	$0.003_{\pm 0.0005}$	$0.0022_{\pm 0.0005}$	$0.032_{\pm 0.004}$	N/A	$0.19_{\pm 0.04}$
PIS	$0.036_{\pm 0.007}$	$0.305_{\pm 0.013}$	$1.391_{\pm 1.014}$	$2.049_{\pm 2.826}$	$11.28_{\pm 1.365}$
DDS	$0.028_{\pm 0.013}$	$0.416_{\pm 0.094}$	$1.154_{\pm 0.626}$	$1.740_{\pm 1.158}$	N/A[6]
DGFS	$0.019_{\pm 0.008}$	$0.274_{\pm 0.014}$	$0.904_{\pm 0.067}$	$0.180_{\pm 0.083}$	$8.974_{\pm 1.169}$

Figure: The lower the better, DGFS achieves the lowest bias in estimating the partition function across various benchmarks except when compared to FAB w/ BUFFER.

Mode coverage results

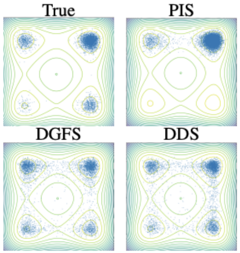


Figure: Manywell plots. DGFS and DDS but not PIS recover all modes

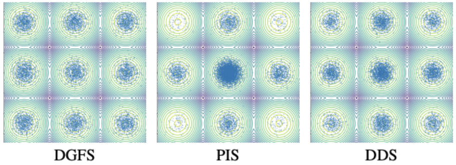


Figure: MoG visualization of DGFS and other diffusion-based samplers shows that DGFS could capture the diverse modes well. The contours display the landscape of the target density

Strengths of DGFS

Strengths

- **Many intermediate signals** lead to better credit assignment, thus improving convergence speed, training stability, and reducing gradient variance.
- **Better credit assignment** also led to better mode discovery and thus helped reducing the lower bound on many of the benchmarks compared to other path space samplers (PIS Zhang et al. 2022 and DDS Vargas et al. 2023).
- Proposed a new dynamic between **Stochastic Optimal Control**, **Sampling from an unnormalized density**, and **GFlowNet**.
- Kept **convergence guarantees** from PIS seminar work.

Weaknesses of DGFS

Weaknesses / Limitations

- **Hyperparameter sensitivity**: A great part of the Appendix was dedicated to ablation studies, thus showing that the hyperparameters do vary a lot depending on the experiments.
- **New framework introduces a second neural network**, which increases the training difficulty of this algorithm.
- Even if it showed the best results among the path samplers in the benchmark, it was still magnitudes behind the FAB algorithm and **showed a lot of difficulties in the intermediate to hard level experiments**, thus increasing my doubts regarding the usage of this algorithm in a true complex highly dimensional setting for real world applications.

Future Work

- **Rare event sampler:** By changing the framework, we would be able to emphasize rare event samples given an unnormalized distribution. By showing good mode discovery, we could train this sampler to retrieve many low-likelihood situations in order to prepare critical systems or safety systems in order to prepare them for all possibilities, not only the most likely.
- **Latent space path:** We have seen many diffusion models papers taking the approach of diffusing on a latent space instead of the sample space (Stable Diffusion and DALL-E). Would this technique be as effective in a latent path space as it is in a latent sample space?
- **Introducing more tricks from RL:** We could see that one of the really good performing samplers was FAB w/ BUFFER and they even talked slightly about off-policy in DGFS. It would be very interesting to implement replay buffer, off-policy exploration strategies, and other tricks which RL relies on to improve its performances.

Questions and hopefully answers :)

Questions!

References I

Vargas, Francisco, Will Sussman Grathwohl, and Arnaud Doucet (2023). “Denoising Diffusion Samplers”. In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=8pvnfTAbu1f.

Zhang, Qinsheng and Yongxin Chen (2022). “Path Integral Sampler: a stochastic control approach for sampling”. In: *International Conference on Learning Representations*.