Journal of Algorithms and Computation

journal homepage: http://jac.ut.ac.ir

Benchmarking GFlowNets against MCMC: The Role of Peak Sharpness and Dimensionality in Discrete Sampling

H. M. Fard*[1]

[1]School of Engineering Science, College of Engineering, University of Tehran, Tehran, Iran

ABSTRACT

With the emergence of Generative Flow Networks (GFlowNets) as a new paradigm in amortized inference, significant questions have arisen regarding the standing of traditional sampling methods such as Markov Chain Monte Carlo (MCMC). While generative models promise to mitigate "mode mixing" challenges, their precise performance boundaries compared to computationally cheaper classical methods remain ambiguous. In this study, we conduct a comprehensive comparative evaluation between major GFlowNet objectives (including TB, DB, and FM) and the Metropolis-Hastings algorithm within discrete environments. The primary focus of this investigation is to analyze the sensitivity of these models to "Reward Landscape Geometry" and dimensional complexity. We examine under which conditions the computational overhead of training a deep model is justifiable and identifying the critical points where traditional methods maintain their robustness. The findings of this research provide novel insights into selecting the optimal sampling strategy, regarding the universal superiority of learning-based approaches.

ARTICLE INFO

Article history:
Research paper
Received 10, December 2025
Accepted 21, December 2025
Available online 29, December 2025

Keyword: MCMC, Generative Flow Networks, Amortized Inference, Reward Geometry, Sensitivity Analysis.

AMS subject Classification: 05C21, 60K20.

*Email: hesam.m.fard@ut.ac.ir

1 Introduction

Sampling from unnormalized probability distributions in high-dimensional combinatorial spaces constitutes a fundamental challenge in computer science, statistical physics, and Bayesian inference. This problem is particularly critical in applications such as de novo drug design, material discovery, and causal structure learning, where the search space is vast and the reward landscape is often multimodal and sparse.

For decades, Markov Chain Monte Carlo (MCMC) algorithms [1, 2], particularly the Metropolis-Hastings algorithm, have served as the gold standard for these tasks. The widespread adoption of MCMC stems from its implementation simplicity, strong theoretical guarantees of asymptotic convergence to the target distribution, and negligible computational overhead per sampling step. Consequently, numerous operational systems in scientific computing continue to rely on the robustness of MCMC. However, MCMC methods fundamentally rely on local exploration. In energy landscapes where modes are separated by vast regions of low probability (high energy barriers), MCMC chains often suffer from "slow mixing," effectively becoming trapped in local optima for impractical durations.

In recent years, a new paradigm known as Generative Flow Networks (GFlowNets) [3] has emerged to address these limitations. Adopting an "amortized inference" approach, GFlowNets attempt to shift the computational burden of sampling from inference time to training time. Unlike MCMC, which wanders through the state space stochastically, a GFlowNet learns a constructive policy to generate diverse samples proportional to their rewards. To stabilize the training of these deep generative models, various objectives have been proposed, including Flow Matching (FM) [3], Trajectory Balance (TB) [4], and Detailed Balance (DB) [5].

Existing literature often highlights the superiority of GFlowNets in complex tasks, suggesting they universally overcome the mode-mixing issues of classical methods. However, a critical and less scrutinized question remains: *Is the architectural complexity and high training cost of GFlowNets justifiable in all scenarios?*

In this paper, we challenge the assumption of the universal superiority of deep generative samplers. We hypothesize that the efficacy of GFlowNets is strictly conditional on the geometry of the reward function. By conducting systematic, controlled experiments in the standard HyperGrid environment [3], we investigate the sensitivity of both paradigms to "Reward Peak Sharpness" (σ) and dimensional complexity.

Our primary contribution is a rigorous sensitivity analysis demonstrating that:

1. In regimes with "needle-like" (Dirac) distributions or lower dimensions, classical MCMC significantly outperforms GFlowNets in both speed and accuracy.

2. GFlowNets suffer from performance collapse in the absence of smooth gradient signals (sparse rewards), a limitation often overlooked in benchmarks optimized for deep learning.

3. The advantage of GFlowNets (specifically with the Detailed Balance objective) is

realized only when the state space is sufficiently large and the reward landscape possesses enough smoothness to guide the learning agent.

We argue that blindly replacing MCMC with deep generative models is not always the optimal strategy. Instead, the choice of sampler should be dictated by the topological features of the energy landscape.

2 Preliminaries and Problem Formulation

2.1 Problem Statement: Sampling from Unnormalized Densities

Let $\mathcal{S}$ be a discrete compositional state space (e.g., a grid or a graph). We are given a non-negative reward function $R : \mathcal{S} \to \mathbb{R}_{\geq 0}$, which defines an unnormalized probability distribution over the states. Our goal is to sample objects $x \in \mathcal{S}$ with probability proportional to their reward:

$$P(x) = \frac{R(x)}{Z}, \quad \text{where} \quad Z = \sum_{x' \in \mathcal{S}} R(x') \tag{1}$$

Here, Z is the partition function. In high-dimensional combinatorial spaces, computing Z directly is intractable due to the exponential size of $|\mathcal{S}|$. The challenge is to generate independent and identically distributed (i.i.d.) samples from $P(x)$ efficiently.

2.2 Markov Chain Monte Carlo (MCMC)

The standard approach for this task is Markov Chain Monte Carlo (MCMC). Specifically, we utilize the Metropolis-Hastings algorithm [1, 2] to construct a reversible Markov chain. At each time step t, given the current state x_t, a candidate state x' is proposed from a distribution $q(x'|x_t)$. The candidate is accepted with probability α:

$$\alpha(x_t, x') = \min\left(1, \frac{R(x')q(x_t|x')}{R(x_t)q(x'|x_t)}\right) \tag{2}$$

In our grid environment, we employ a symmetric random walk proposal where $q(x'|x) = q(x|x')$, simplifying the acceptance ratio to $\min(1, R(x')/R(x_t))$. While MCMC guarantees asymptotic convergence to the target distribution as $t \to \infty$, it suffers from "slow mixing" in multimodal landscapes. When modes are separated by vast low-reward regions (energy barriers), the acceptance probability for traversing these regions becomes negligible, causing the chain to get trapped in local optima.

2.3 Generative Flow Networks (GFlowNets)

Generative Flow Networks [3] frame the sampling problem as a sequential decision-making process on a directed acyclic graph (DAG). An agent starts at a special source state s_0

and selects a sequence of actions to construct a terminal state $x \in \mathcal{S}$. The objective is to learn a forward policy $P_F(s'|s)$ such that the marginal probability of generating x matches the target distribution $P(x) \propto R(x)$. Unlike MCMC, GFlowNets also learn to estimate the partition function Z. We compare three primary training objectives:
1. Flow Matching (FM): Based on the principle of flow conservation, FM ensures that for any state s, the total incoming flow equals the total outgoing flow [3]:

$$\mathcal{L}_{FM}(s) = \left(\sum_{s' \in \mathcal{P}(s)} F(s', s) - \sum_{s'' \in \mathcal{C}(s)} F(s, s'') - R(s)\mathbb{I}(s \in \mathcal{X}) \right)^2 \tag{3}$$

where $\mathcal{P}(s)$ and $\mathcal{C}(s)$ denote the parents and children of s, respectively.
2. Trajectory Balance (TB): Instead of local states, TB optimizes compatibility over complete trajectories $\tau = (s_0 \to \cdots \to x)$. This objective typically yields faster convergence and lower variance [4]:

$$\mathcal{L}_{TB}(\tau) = \left(\log Z + \sum_t \log P_F(s_{t+1}|s_t) - \log R(x) - \sum_t \log P_B(s_t|s_{t+1}) \right)^2 \tag{4}$$

Here, P_B is a backward policy that learns to navigate from terminal states back to the source.
3. Detailed Balance (DB): This objective enforces a balance constraint locally on each transition (edge) rather than the whole path or node accumulation [5]:

$$\mathcal{L}_{DB}(s, s') = \left(\log F(s) + \log P_F(s'|s) - \log F(s') - \log P_B(s|s') \right)^2 \tag{5}$$

DB is particularly computationally efficient as it avoids the expensive summation over parents required by FM.

2.4 The HyperGrid Environment

To benchmark the exploration capabilities of these algorithms, we employ the HyperGrid environment [3]. The state space is a D-dimensional grid of side length H, defined as $\mathcal{S} = \{0, \ldots, H-1\}^D$. The reward function is designed to be highly multimodal, consisting of a base reward R_0 and Gaussian peaks located at the corners of the hypercube $\mathcal{C}$:

$$R(x) = R_0 + \sum_{c \in \mathcal{C}} \exp\left(-\frac{\|x - c\|_2^2}{\sigma^2} \right) \tag{6}$$

Two key parameters control the difficulty of the landscape:

- R_0 **(Base Reward):** Controls the depth of the "valleys" between modes. Lower R_0 makes mode-switching for MCMC exponentially harder.
- σ **(Peak Sharpness):** Controls the width of the modes. Lower σ results in "needle-in-a-haystack" rewards, testing the gradient propagation capabilities of GFlowNets.

3 Experimental Setup

To conduct a rigorous and comparative evaluation of the exploration capabilities of GFlowNets versus MCMC, we designed a suite of systematic benchmarks within the HyperGrid environment. This section details the environmental configurations, model architectures, and the specific "saturation" conditions under which the algorithms were tested.

3.1 Environment Configuration and Complexity

We fixed the grid side length to $H = 8$ and varied the spatial dimensionality $D \in \{2, 4, 8\}$ to simulate the "Curse of Dimensionality." The complexity of the state space grows exponentially:

- **Low-Dimensional ($D = 2$):** $|S| = 64$ states. Trivial for any solver.
- **Mid-Dimensional ($D = 4$):** $|S| = 4,096$ states. A manageable search space.
- **High-Dimensional ($D = 8$):** $|S| \approx 1.67 \times 10^7$ states with 256 distinct modes. This regime presents a significant challenge for exploration algorithms.

To analyze the sensitivity to reward geometry, we systematically swept over two key parameters:

1. **Peak Width ($\sigma \in [0.05, 2.0]$):** Varying σ allows us to transition from "smooth" landscapes (where gradient information is available) to "needle-in-a-haystack" landscapes (where rewards appear as Dirac deltas).
2. **Valley Depth ($R_0 \in [0.001, 0.1]$):** Varying the base reward tests the algorithms' ability to traverse low-probability regions without getting trapped in local optima.

3.2 MCMC Configuration: The Saturation Hypothesis

A critical aspect of our experimental design is the initialization of the MCMC sampler. We instantiated $N = 2048$ parallel Metropolis-Hastings chains, all initialized at the origin $(0, \ldots, 0)$. This setup challenges the chains to escape the initial mode and discover distant high-reward regions.

Saturation Effect: We explicitly analyze the performance in the context of "State Space Saturation." In lower dimensions (e.g., $D = 4$), the number of parallel chains is comparable to the total size of the state space ($2048 \approx 0.5 \times |S|$). Under these conditions, MCMC effectively performs a brute-force exhaustive search rather than a stochastic sampling process. This mimics the asymptotic behavior of MCMC where $t \to \infty$. However, in $D = 8$, the saturation ratio drops to $\approx 0.01\%$, revealing the true mixing limitations of the algorithm.

3.3 GFlowNet Architecture and Training

For all GFlowNet agents (trained with Flow Matching, Trajectory Balance, and Detailed Balance), we employed a standardized Multi-Layer Perceptron (MLP) architecture to ensure fair comparison:

- **Architecture:** Two hidden layers with 512 units each.
- **Activation:** LeakyReLU.
- **Optimizer:** Adam with a learning rate of $\eta = 10^{-3}$.
- **Batch Size:** 2048 (matching the number of MCMC chains).

The models were trained for 3,000 iterations. We utilized a uniform backward policy P_B for TB and DB objectives, as implies by the grid structure.

3.4 Evaluation Metric

Our primary quantitative metric is the **Number of Modes Found**. Given the ground truth locations of the modes (the hypercube corners), a mode c is considered "found" if the generative model produces at least one sample x within the immediate neighborhood of c:

$$\min_{x \in \mathcal{D}_{gen}} \|x - c\|_1 \leq 1 \tag{7}$$

where $\mathcal{D}_{gen}$ is the batch of generated samples. We report the mean performance over 3 independent seeds.

4 Empirical Results and Analysis

4.1 Scalability and the MCMC Saturation Effect

We first examine the performance scaling with respect to dimensionality (D). As illustrated in Figure 1, in low-dimensional settings ($D = 2$ and $D = 4$), the MCMC baseline achieves near-perfect coverage (100% modes found) almost instantaneously.
This phenomenon corroborates our "Saturation Hypothesis" and can be explained through the lens of **Asymptotic Equivalence**. In our setup for $D = 4$, the population of $N = 2048$ chains covers $\approx 50\%$ of the state space ($|S| = 4096$). Running such a dense population is mathematically equivalent to running a single chain for an infinite time horizon ($t \to \infty$). According to the ergodic theorem, MCMC is theoretically guaranteed to converge to the target distribution given sufficient time. By saturating the space, we effectively provided MCMC with this "infinite time" resource instantly, rendering its success inevitable.
Crucially, this guarantee does not apply to GFlowNets. Even if given infinite time or infinite agents, a gradient-based learner cannot converge if the learning signal is absent.

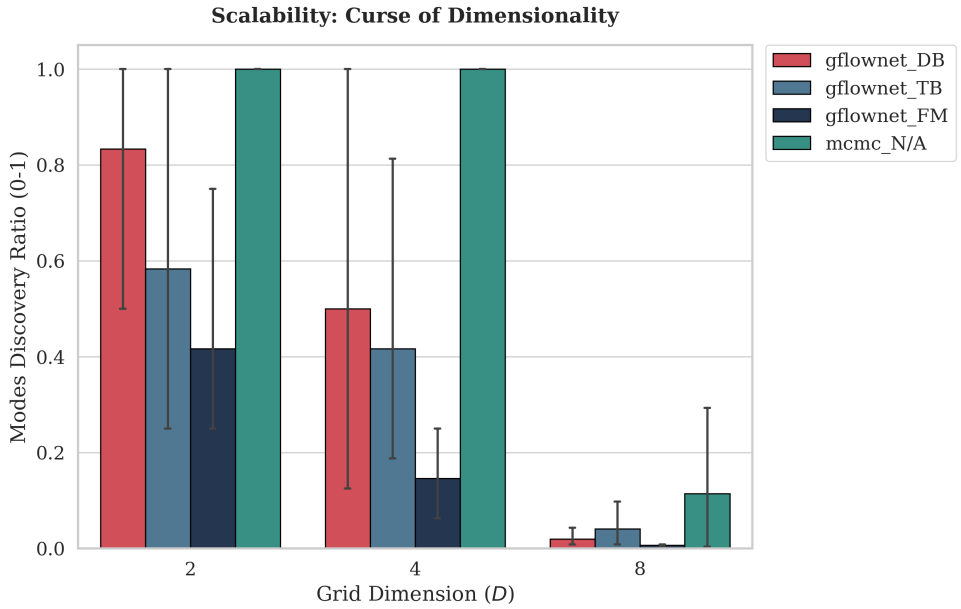

Figure 1: Scalability analysis across dimensions. In $D = 2, 4$, MCMC saturates the state space, achieving 100% coverage. In $D = 8$, the saturation effect vanishes, revealing the true exploration limits of all methods.

However, as the dimension increases to $D = 8$ (16.7 million states), the saturation ratio drops to $\approx 0.01\%$. Here, the "infinite time" illusion vanishes, and MCMC's performance drops significantly (finding $< 5\%$ of modes). This confirms that MCMC's dominance in small benchmarks is often an artifact of finite-size effects rather than algorithmic superiority.

4.2 Sensitivity to Reward Landscape Geometry

The most critical finding of this study is the sensitivity of amortized inference to the "sharpness" of the reward function. Figure 2 and Table 1 present the results of the σ-sweep in the 8D HyperGrid.
We observe a clear **"Crossover Point"** around $\sigma \approx 1.0$:

- **Regime I: Wide Peaks ($\sigma = 2.0$):** The reward signal is smooth and globally informative. Here, GFlowNet (specifically with the Detailed Balance objective) demonstrates its strength, discovering **38 modes** compared to MCMC's 8. The generative policy successfully generalizes across the high-dimensional space.
- **Regime II: Sharp Peaks ($\sigma \leq 0.5$):** As the peaks approach Dirac deltas, the gradient signal becomes extremely sparse. GFlowNet performance collapses to near-zero (finding only 1-2 modes). In stark contrast, MCMC exhibits superior robustness, maintaining a discovery rate of 11-16 modes.

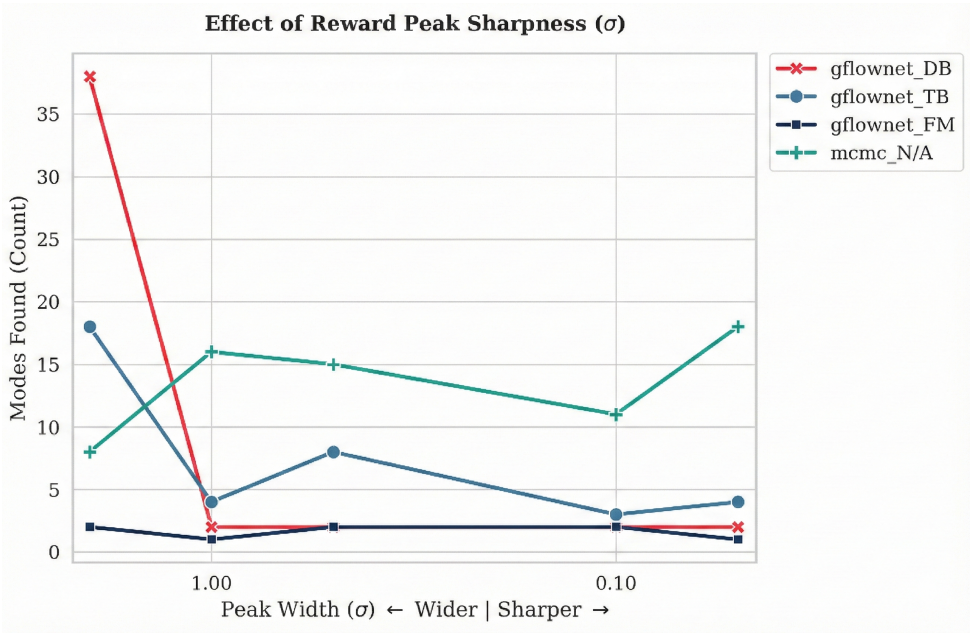

Figure 2: Sensitivity to Reward Peak Sharpness (σ). As peaks become sharper ($\sigma \to 0.1$), GFlowNet performance collapses due to vanishing gradients, while MCMC maintains a robust baseline.

Table 1: Modes found in 8D HyperGrid under varying Peak Width (σ). Total Modes = 256. Data indicates a crossover in performance as sparsity increases.

Method	$\sigma = 2.0$ **(Wide)**	$\sigma = 1.0$	$\sigma = 0.5$	$\sigma = 0.1$ **(Sharp)**
GFN - Flow Matching	2	1	2	1
GFN - Trajectory Balance	18	4	8	3
GFN - Detailed Balance	**38**	2	2	2
MCMC (Metropolis)	8	**16**	**15**	**11**

This result challenges the prevailing narrative that GFlowNets are universally better for multimodal distributions. In "needle-in-a-haystack" scenarios, the random walk of MCMC—being blind to gradients—is paradoxically more effective than a gradient-dependent learner that suffers from signal vanishing.

4.3 Computational Efficiency and Objective Comparison

Analyzing the training dynamics (Figure 3) and computational costs reveals a significant trade-off. According to our data, GFlowNet training required approximately 1500 seconds on average, whereas MCMC completed its sampling in under 3 seconds.
While GFlowNet-DB found 4.75× more modes than MCMC in the wide-peak regime ($\sigma = 2.0$), this came at a $\approx 500\times$ increase in computational cost. Regarding the objectives,

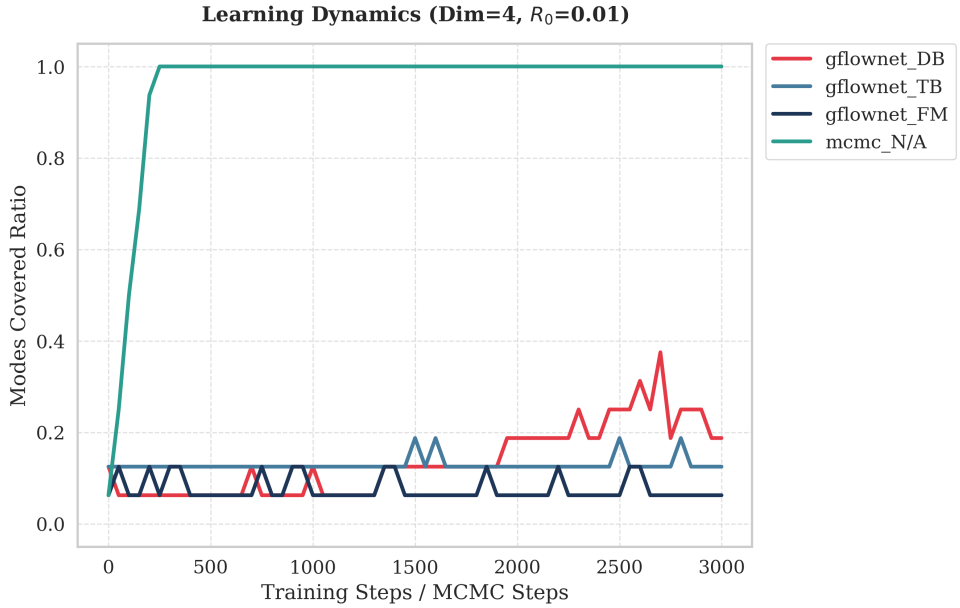

Figure 3: Learning dynamics of GFlowNet objectives. Detailed Balance (DB) shows faster and more stable convergence compared to Trajectory Balance (TB) and Flow Matching (FM).

Detailed Balance (DB) consistently outperformed Trajectory Balance (TB) and Flow Matching (FM). FM failed to converge in most high-dimensional experiments, suggesting that the variance of the flow consistency loss is too high for sparse discrete graphs.

5 Conclusions and Future Directions

In this study, we rigorously investigated the performance boundaries of emerging GFlowNet models compared to the classical MCMC baseline in discrete combinatorial environments. Contrary to the growing narrative of deep generative supremacy, our findings demonstrate that the superiority of GFlowNet is strictly conditional on the *reward landscape geometry* and the *dimensionality* of the problem.
Our key conclusions are threefold:

1. **The Myth of Universal Superiority:** We demonstrated that in lower dimensions or when the agent-to-state ratio is high (saturation regime), MCMC is theoretically and practically superior. Its guarantee of asymptotic convergence makes it the most cost-effective solution for small to medium-scale problems, rendering the high training cost of deep models unjustifiable in these settings.

2. **Gradient Dependency:** GFlowNets are heavily reliant on the existence of a continuous learning signal. Our sensitivity analysis revealed that as reward peaks become sharper ($\sigma \leq 0.5$), the performance of GFlowNets collapses due to vanishing

gradients in the sparse state space. In these "needle-in-a-haystack" scenarios, the gradient-free local search of MCMC proves to be significantly more robust.

3. **The High-Dimensional Advantage:** GFlowNets, particularly with the **Detailed Balance (DB)** objective, justify their computational cost only in high-dimensional spaces ($D = 8$) characterized by smooth reward structures. In such regimes, they successfully bridge the energy barriers that trap MCMC chains, discovering $4.75\times$ more modes than the baseline.

Future Directions: Based on these insights, we argue that GFlowNets should not be viewed as a blanket replacement for MCMC. Instead, future research should focus on **Hybrid Architectures**. Promising directions include:

- Combining the global exploration capability of GFlowNets to identify high-probability regions with the local precision of MCMC for fine-tuning samples.
- Integrating *Tempering* techniques into GFlowNet training to artificially smooth sharp reward landscapes during the early stages of learning, thereby mitigating the vanishing gradient problem observed in our sparse experiments.

References

[1] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6), 1087-1092.

[2] Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1), 97-109.

[3] Bengio, E., Jain, M., Korablyov, M., Precup, D., & Bengio, Y. (2021). Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. *Advances in Neural Information Processing Systems*, 34, 27381-27394.

[4] Malkin, N., Jain, M., Bengio, E., Sun, C., & Bengio, Y. (2022). Trajectory Balance: Improved Credit Assignment in GFlowNets. *Advances in Neural Information Processing Systems*, 35, 5915-5927.

[5] Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., & Bengio, E. (2023). GFlowNet Foundations. *Journal of Machine Learning Research*, 24(210), 1-55.