Streaming, Distributed, and Asynchronous Amortized Inference

Tiago da Silva Henrique[1], **Diego Mesquita**[1]

[1]School of Applied Mathematics – Getulio Vargas Foundation
Rio de Janeiro, RJ – Brazil

{tiago.henrique, diego.mesquita}@fgv.br

Abstract. *We address the problem of amortized inference over a compositional finite space in a non-localized environment, i.e., when data are observed in a distributed, streaming, or mixed (asynchronous) fashion. This setting comprises applications in causal discovery, phylogenetic inference, natural language processing, and other problems. In particular, we focus on Generative Flow Networks (GFlowNets), an emergent family of deep generative models that cast amortized inference as finding a balanced flow assignment in a flow network. To accomplish this, a GFlowNet parameterizes the flow function as a neural network and optimizes its parameters via stochastic gradient descent. Drawing on this, we make both practical and theoretical contributions. On the practical side, we introduce three algorithms — Streaming Bayes (SB) GFlowNets, Embarrassingly Parallel (EP) GFlowNets, and Subgraph Asynchronous Learning (SAL) — along with efficient gradient estimators that significantly accelerate GFlowNet training when compared against traditional approaches. Also, we develop the first computationally amenable and sound metric for assessing the correctness of a trained GFlowNet. From a theoretical perspective, we delineate the limitations and present the first non-vacuous generalization guarantees for the learning of GFlowNets. All in all, our work paves the road for a better understanding, usability, and fair assessment of amortized inference algorithms. This extended abstract provides an overview of our research, which was published at the proceedings of ICML [da Silva et al. 2024c], NeurIPS [da Silva et al. 2024a, da Silva et al. 2024b], and ICLR [da Silva et al. 2025b, da Silva et al. 2025a].*

1. Introduction

Sampling from an unnormalized distribution is a fundamental challenge in machine learning and artificial intelligence — without a clear one-size-fits-all solution [Blei and et al. 2017, Buesing et al. 2020]. Historically, Markov Chain Monte Carlo (MCMC) methods have been the predominant approach in addressing this problem [Liu and Liu 2001]. Nonetheless, MCMC often suffers from slow mixing times and is notoriously hard to diagnose [Roy 2020]. For continuous distributions, existing algorithms mitigate these issues by leveraging the differential structure of the underlying state space to aid exploration [Neal et al. 2011, Carpenter et al. 2017]. In discrete spaces, however, gradient-based operators are unavailable — and MCMC becomes remarkably inefficient [Rhodes and Gutmann 2019, Grathwohl et al. 2021]. To overcome this difficulty, [Bengio et al. 2021, Bengio et al. 2023] introduced Generative Flow Networks (GFlowNets; see Section 2 and Figure 2) to tackle the sampling problem in finite spaces. Since then, numerous studies have consistently demonstrated

(a) Markov Decision Process (MDP). (b) Flow network.

Figure 1. A GFlowNet for generating 2-sized subsets of $\{1,2,3\}$ ($\mathcal{X} \coloneqq [\{1,2,3\}]^2$). In (a), we define an MDP over an extension of $\mathcal{X}$ describing how the elements of $\mathcal{X}$ can be constructed through the iterative application of simple actions to a single initial state, $\{\}$. In (b), we present the corresponding flow network, in which edges replace actions and flows replace probabilities.

that GFlowNets drastically outperform conventional Markov chain-based methods [Bengio et al. 2021, Lahlou et al. 2023, Zhou et al. 2024].

In a nutshell, a GFlowNet learns a stochastic policy in a Markov Decision Process (MDP) over an extension of the target distribution's support in such a way that each element in the original space is sampled in proportion to its unnormalized probability. To achieve this, [Bengio et al. 2021] reframed the MDP as a single-source flow network and the policy learning problem as a flow assignment problem; see Figure 2 for an example. Building upon this hermeneutical shift, [Bengio et al. 2023] derived simple *balance conditions* and demonstrated that enforcing them is equivalent to solving the original MDP. In doing so, the authors laid the groundwork for a series of breakthroughs in drug discovery [Bengio et al. 2021, Pandey et al. 2024], phylogenetic inference [Zhou et al. 2024, da Silva et al. 2024c], natural language processing [Hu et al. 2023, Bengio and Malkin 2024], combinatorial optimization [Zhang et al. 2023b, Zhang et al. 2023a], and structure learning [Deleu et al. 2022, Deleu and Bengio 2023] fueled by GFlowNets. Despite their success, recent advances have been mostly restricted to relatively small problems, and the scaling of GFlowNets remains challenging. To alleviate this difficulty, our first contribution consists of the design of provably low-variance learning algorithms for the sample-efficient training of GFlowNets, as briefly described in Research Question I (**RQI**) and outlined in Section 3.

Research Question I: *The training of GFlowNets is compute- and time-intensive. Can we speed-up this process by designing more efficient learning objectives requiring a smaller number of gradient steps to achieve a good approximation to the target distribution?*

As epitomized by Richard Sutton's Bitter Lesson [Sutton 2019], however, history has repeatedly shown us that increased computation often supersedes enhanced algorithmic approaches in the solution of computationally hard problems. With this in mind, an alternative pathway to improve GFlowNet learning is through algorithmic changes enabling the training of these models in modern computer clusters through non-localized algorithms. At the very least, this allows for more computation to be performed in less time in the pursuit of solving the flow assignment problem. This is the defining issue posed by **RQII** that will be considered in detail in Section 4. As we will show, our streaming, distributed, and asynchronous algorithms dramatically accelerate the training of GFlowNets relatively to a centralized and monolithic approach. Due to their compositional nature, our methods also facilitate the modular reuse of GFlowNets, which is a topic of growing interest in the deep learning community [Garipov et al. 2023, Du and Kaelbling 2024].

Research Question II: *The success of contemporary machine learning is largely due to the huge computational resources available in modern computer clusters. In this regard, how to adapt GFlowNet training to the distributed and continual learning paradigms?*

Additionally, in spite of significant practical advancements, the theoretical properties of GFlowNets remain largely underexplored by the literature. Indeed, the foundational work of [Bengio et al. 2023] only established that the flow assignment problem *has a solution*, but not whether such a solution can be adequately approximated by a given family of models via a chosen learning algorithm, e.g., stochastic gradient descent. As stated by **RQIII**, although the empirical effectiveness of GFlowNets has been attributed to their generalization capabilities [Bengio et al. 2021, Pandey et al. 2024, Atanackovic and Bengio 2024], no work so far has rigorously shown this. Moreover, there is a lack of consensus on how to efficiently probe the proximity of a GFlowNet to its learning objective via a tractable *risk functional* [Shen et al. 2023, Kim et al. 2024b, Lau et al. 2024]. To fill this gap, we describe in Section 5 the first empirical and theoretical guarantees on the learning and generalization of GFlowNets.

Research Question III: *The generalization and distributional limits of GFlowNets remain elusive. Are GFlowNets capable of learning a (provably) generalizable flow function given a hypothesis space and a training algorithm with a fixed computational budget?*

From a broader perspective, our work is positioned at the intersection between Deep Reinforcement Learning (Deep RL), the importance of which has been recently recognized by the 2024 Turing Award [Association for Computing Machinery 2025], and approximate statistical inference. Notably, this has been a major research line that aims at the unification of planning and inference [Lázaro-Gredilla et al. 2025] through Friston's free-energy principle [Parr et al. 2022]. The central theme in this literature is how to train an agent to solve a given inferential task and, conversely, how to interpret an agent's behavior through the well-founded framework of statistical inference. Moreover, our discussion connects to the debate on how to reliably integrate System 2 thinking into deep learning algorithms [Bengio 2022, Bengio and Malkin 2024] — a challenge that played a key role in the original design of GFlowNets [Bengio et al. 2021].

Finally, our contributions also represent a significant advancement in understanding and improving the usability of amortized inference algorithms based on sequential decision making — with GFlowNets as their most prominent representative [Boussif et al. 2024, Kim et al. 2024a]. On the one hand, the theoretical underpinnings laid out in Sections 3 and 5 provide a framework for thinking about and effectively measuring the difficulty and potential failure modes of GFlowNets. On the other hand, the efficient non-localized methods introduced in Section 4 promote the seamless scaling of GFlowNet training to larger problems with negligible algorithmic changes and minimal inter-instance communication in a distributed computing context. Overall, we are confident that our contributions provide a strong basis for future research on the use of GFlowNets in large-scale applications of amortized inference in a principled manner.

Paper outline. Section 2 introduces GFlowNets, while Section 3 lays out our algorithmic contributions for their efficient training (**RQI**). Then, Section 4 summarizes our methods for the non-localized learning of GFlowNets (**RQII**). Lastly, Section 5 overviews our novel theoretical results on their learnability (**RQIII**). Section 6 wraps up the discussion.

2. Background

Notations and definitions. Let $\mathcal{X}$ be a finite set and $R\colon \mathcal{X} \to \mathbb{R}_+$ be a positive function on $\mathcal{X}$, which is called a *reward function* due to a terminological inheritance from the RL literature. Our objective is to sample objects from $\mathcal{X}$ in proportion to R. Also, let $\mathcal{S} \supseteq \mathcal{X}$ be the *state space* and $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ be a directed acyclic graph (DAG) with edges $\mathcal{E}$, which we call the *state graph*. In Figure 1, $\mathcal{X} = \{\{1,2\},\{1,3\},\{2,3\}\}$ and $\mathcal{S} = \{\{\},\{1\},\{2\},\{3\}\} \cup \mathcal{X}$. We assume $\mathcal{G}$ is *pointed*, i.e., there are distinguished *initial* $s_o \in \mathcal{S}$ and *final* $s_f \in \mathcal{S}$ states such that there are no incoming (resp. outgoing) edges from s_o (resp. s_f), and that only states in $\mathcal{X}$ are connected to s_f. Conversely, s_f is the only child of each $x \in \mathcal{X}$. For this reason, we refer to $\mathcal{X}$ as the set of *terminal states*. Illustratively, Figure 2 shows a state graph for the task of generating 2-sized multisubsets of $\{1,2\}$. In this context, a *forward policy* $p_F\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ in $\mathcal{G}$ is a transition kernel for which $p_F(s,\cdot)$ is a probability measure supported on the children of s in $\mathcal{G}$; we use $p_F(s'|s)$ and $p_F(s,s')$ interchangeably. A *backward policy* $p_B\colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+$ is a forward policy on the transpose $\mathcal{G}^\top$ of $\mathcal{G}$. Throughout the text, we assume that $p_B(s,\cdot)$ is uniform for every s [Bengio et al. 2023]. Clearly, both p_F (resp. p_B) induce a probability distribution over trajectories in $\mathcal{G}$ (resp. $\mathcal{G}^\top$) via $p_F(\tau|s_1) = \prod_{i=2}^{n} p_F(s_i|s_{i-1})$ (resp. $p_B = \prod_{i=2}^{n} p_B(s_{i-1}|s_i)$) for $\tau = (s_1,\ldots,s_n)$. We call a trajectory *complete* when it starts at s_o and ends at s_f, and we denote by $|\tau|$ its number of transitions. For example, (s_o, s_1, s_2, s_f) is a complete trajectory in Figure 2 with $|\tau| = 3$. Finally, we refer to $(\mathcal{G}, p_F, p_B, R)$ as a *Markov Decision Process* (MDP).

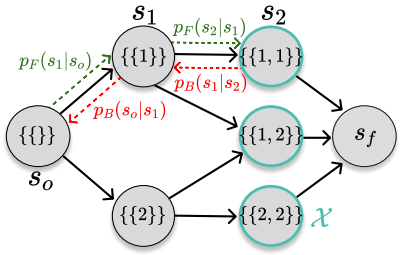

Figure 2. A state graph.

Under the hermeneutics of flow networks, the forward policy $p_F(s,s')$ measures the relative capacity of the edge $s \to s'$ with respect to the flow in s [Bengio et al. 2021]. Similarly, $p_B(s',s)$ represents the relative capacity of $s \to s'$ with respect to the flow in s'. In this context, $R(x)$ represents the desired amount of flow reaching each $x \in \mathcal{X}$ (recall Figure 1). In view of this, we define a *flow function* $F\colon \mathcal{S} \to \mathbb{R}_+$ characterizing the amount of flow within a state s and satisfying $F|_{\mathcal{X}} = R$. We say that a network is *balanced* when the flow in s equals the flow reaching s: $F(s) = \sum_{s' \in \mathrm{Ch}(s)} F(s') p_B(s|s')$, in which $\mathrm{Ch}(s) = \{s'\colon s \to s' \in \mathcal{E}\}$ denotes the children of s in $\mathcal{G}$.

GFlowNets. Generative Flow Networks (GFlowNets) [Bengio et al. 2021] are defined as a family $\{(\mathcal{G}, p_F^\theta, p_B, R, F^\theta)\colon \theta \in \Theta\}$ consisting of an MDP $(\mathcal{G}, p_F^\theta, p_B, R)$ and a flow function F^θ, in which $\theta \in \Theta$ typically represents neural network parameters (e.g., a multilayer perceptron) [Lahlou et al. 2023, Bengio et al. 2023]. The objective is to learn p_F^θ such that its marginal $p_\top^\theta$ over $\mathcal{X}$ matches R up to a normalization constant,

$$p_\top^\theta(x) \coloneqq \sum_{\tau\colon s_o \rightsquigarrow x} p_F^\theta(\tau|s_o) \propto R(x) \tag{1}$$

for $x \in \mathcal{X}$; $\tau\colon s_o \rightsquigarrow x$ means that the trajectory τ starts at s_o and ends at s_f through the transition $x \to s_f$. For example, $p_\top^\theta(s_2) = p_F((s_o, s_1, s_2, s_f)|s_o)$ in Figure 2, as there is a single trajectory from s_o to x. Henceforth, we will omit θ from p_F, F. Clearly, samples from $p_\top$ can be efficiently generated by starting at s_o and following p_F until reaching $\mathcal{X}$.[1]

[1]Readers might consult [Bengio et al. 2023, Sutton and Barto 2018, Owen 2013, Robert et al. 2007] for a review of GFlowNets, RL, Monte Carlo statistical methods, and Bayesian inference, respectively.

3. Efficient learning objectives for GFlowNets

Learning objectives for GFlowNets. The training of GFlowNets searches for a parameter θ in a space Θ minimizing the balance violation in the flow network, i.e., optimizing

$$\left(\log F(s)p_F(s'|s) - \log F(s')p_B(s|s')\right)^2 \tag{2}$$

over all transitions $s \to s'$ in the state graph — with the restriction that $F(x) = R(x)$ for $x \in \mathcal{X}$. The reader is invited to note that, when this condition is zeroed out for all (s, s'),

$$F(s) = F(s) \underbrace{\sum_{s' \in \text{Ch}(s)} p_F(s'|s)}_{=1} = \sum_{s' \in \text{Ch}(s)} F(s')p_B(s|s') \text{ and } p_\top(x) = \frac{F(x)}{F(s_o)} \propto R(x), \tag{3}$$

that is, the flow network is balanced and the proportional constraint on the marginal distribution in Equation (1) is satisfied. As with many deep learning problems, however, directly minimizing the objective function in Equation (2) is infeasible due to the potentially intractable size of the underlying state graph [Bengio et al. 2023]. Instead, we introduce a probability distribution p_E over the space of pairs (s, s') and minimize the expected value of Equation (2) under p_E via stochastic gradient descent [Kingma and Ba 2014]. Inspired by the RL literature, we refer to p_E as an *exploratory policy* [Sutton and Barto 2018]. Unless stated otherwise, we set p_E as an ϵ-greedy version of the learned policy, i.e., $p_E(s'|s) = (1-\epsilon)p_F(s'|s) + \epsilon p_U(s'|s)$ [Malkin et al. 2023], in which $p_U(\cdot|s)$ is an uniform probability distribution over the children of s in the state graph. Under these conditions, the parameter ϵ controls the trade-off between exploration ($\epsilon \to 1$) and exploitation ($\epsilon \to 0$) in the learning algorithm. Finally, as suggested by [Malkin et al. 2022, Bengio et al. 2023], we adopt a trajectory-level optimization objective for efficient sampling and enhanced training stability,

$$\mathcal{L}_{DB}(p_F, F) = \mathbb{E}_{\tau \sim p_E}\left[\frac{1}{|\tau|}\sum_{(s,s') \in \tau}\left(\log \frac{F(s)p_F(s'|s)}{F(s')p_B(s|s')}\right)^2\right], \tag{4}$$

which we term the *detailed balance* (DB) loss due to its connection to the detailed balance condition of Markov chains [Geyer 1991, Deleu and Bengio 2023].

GFlowNet as an amortized inference algorithm. [Malkin et al. 2023] also demonstrated that GFlowNets can be trained via the minimization of a statistical divergence D between the forward and backward policies [Blei and et al. 2017]. To this end, they define $p_B(\tau|s_o) \propto p_B(\tau|x)R(x)$ as the unnormalized probability of a backward complete trajectory starting at x and ending at s_o (e.g., (x, s_2, s_1, s_o) in Figure 2), and propose solving

$$\min_{\theta \in \Theta} D(p_F^\theta(\tau|s_o), p_B(\tau|s_o)). \tag{5}$$

to train a GFlowNet. Common choices for D include the Kullback-Leibler divergence [Kullback and Leibler 1951] or the Renyi-α divergence [Rényi 1961]. Nonetheless, this approach was regarded as clearly inferior than minimizing the balance-based objective presented in Equation (4) [Malkin et al. 2023, Lahlou et al. 2023] — despite the long-standing success of divergence-minimization algorithms in RL and approximate statistical inference problems [Schulman et al. 2017, Blei and et al. 2017, Ahmadian et al. 2024].

Control variates for efficient gradient estimation. In [da Silva et al. 2024b] and [da Silva et al. 2024c], we empirically demonstrated that the reason for this performance gap stems from the large gradient variance associated with divergence-based learning objectives [Richter et al. 2020]. Intuitively, large gradient variance hampers optimization by making it harder to navigate the parameter space effectively. To mitigate this issue, we presented efficiently computable control variates for low-variance gradient estimation [Owen 2013]. Remarkably, the resulting algorithms reduced the variance of the stochastic gradients by up to two orders of magnitude [da Silva et al. 2024b, Figure 2] and significantly accelerated training convergence in standard benchmark tasks [da Silva et al. 2024b, Figure 3], [da Silva et al. 2024c, Figure 7] (**RQI**). As a consequence, we showed that the previously discredited divergence-minimizing methods are often the most effective option for training GFlowNets. Importantly, these methods also form a core component of the non-localized algorithms discussed in the next section.

4. Non-localized training of GFlowNets

Streaming and distributed inference. While algorithmic improvements lay the foundation for the improvement of deep learning-based models, history has shown us that that major breakthroughs in solving real-world problems have primarily stemmed from their ability to scale horizontally across computer clusters and to be continually updated as new data arrives [Sutton 2019, Hooker 2020]. Correspondingly, the remarkable increase in our capacity to collect and store data, which is the trademark of the so-called *big data era*, raises concerns regarding the maintenance of up-to-date models without having to repeatedly retrain them from scratch to adapt to an ever-changing environment. During the COVID-19 pandemic, for example, advances in genomic sequencing technologies led to the rapid expansion of already extensive genetic databases. This challenged researchers to maintain accurate models for the genomic surveillance of SARS-CoV-2, a core prerequisite for effectively tracking the outbreak dynamics [Cappello et al. 2021]. To address these issues, we introduced SB-GFlowNets [da Silva et al. 2024a] and EP-GFlowNets [da Silva et al. 2024c] as the first methods for performing streaming and parallel Bayesian inference in discrete spaces, respectively (**RQII**).

SB-GFlowNets. The reader should recall that, given N independently sampled datasets $\mathbf{D} = \{\mathcal{D}_1, \ldots, \mathcal{D}_N\}$, a Bayesian posterior distribution π over a parameter space $\mathcal{X}$ with likelihood function f and prior π_o may be decomposed as

$$\pi(x|\mathbf{D}) \propto \pi_o(x) f(x|\mathcal{D}_1) \cdot \prod_{2 \leq n \leq N} f(x|\mathcal{D}_n) = \prod_{1 \leq n \leq N} \pi_o(x)^{\frac{1}{N}} f(x|\mathcal{D}_n). \tag{6}$$

Resting on this identity, a SB-GFlowNet iteratively updates an initial model as datasets $\mathcal{D}_1, \ldots, \mathcal{D}_N$ arrive sequentially. First, a GFlowNet having $\pi_o(x) f(x|\mathcal{D}_1)$ as a reward function is trained. Successively, for each $n \geq 2$, the nth GFlowNet is trained on an unbiased estimate of the product between the $(n-1)$th GFlowNet and $f(x|\mathcal{D}_n)$. That is, the $(n-1)$th GFlowNet works as a prior for the nth approximated posterior. As succintly put by [Lindley 1972], "today's posterior is tomorrow's prior".

EP-GFlowNets. On the other hand, an EP-GFlowNet applies a divide-and-conquer approach to combine GFlowNet-based surrogates of each *subposterior* $\pi_o(x)^{1/N} f(x|\mathcal{D}_n)$. First, these surrogates are independently trained on separate nodes in a computer cluster.

Then, the trained GFlowNets are stored in a central server. Finally, a global GFlowNet is trained to sample from an estimate of their product. Notably, this approach requires minimal communication between clients and server — with negligible implementation overhead. We rigorously prove the correctness of both SB- and EP-GFlowNets in [da Silva et al. 2024a, Proposition 1] and [da Silva et al. 2024c, Theorem 3.1]. Empirically, our results in [da Silva et al. 2024a, Table 2] and [da Silva et al. 2024c, Figure 12] show that both methods reduce training time by half when compared against a centralized GFlowNet in standard benchmark tasks, while maintaining an equivalent accuracy.

5. On the assessment and generalization of GFlowNets

Flow Consistency in Subgraphs. Despite strong performance on benchmark tasks, the assessment of the accuracy of a GFlowNet in realistic large-scale problems remained elusive. The reason for this is that, for large state spaces (e.g., with more than 10^{10} elements), direct comparison of the learned to the target distribution is infeasible. Drawing on the RL literature, past works have relied on the rate with which a GFlowNet finds high-probability (high-rewarding) regions of the state space as a proxy for its distributional accuracy [Bengio et al. 2021, Malkin et al. 2023]. As we demonstrated in [da Silva et al. 2025a, Figure 8], however, this metric fails at measuring the correctness of a GFlowNet. Instead, we introduced *Flow Consistency in Subgraphs* (FCS) as a tractable and provably correct surrogate for a GFlowNet's accuracy. Briefly, FCS measures the absolute distance between the learned and target distributions on random and small subsets of $\mathcal{X}$, i.e.,

$$\mathrm{FCS}(p_F, p_B) = \mathbb{E}_{\tau_1,\cdots,\tau_N \sim p_E}\left[\frac{1}{2}\sum_{1\leq n\leq N}\left|\frac{p_\top(x_n)}{\sum_{m=1}^{N} p_\top(x_m)} - \frac{R(x_n)}{\sum_{m=1}^{N} R(x_m)}\right|\right]. \quad (7)$$

In [da Silva et al. 2025a, Figure 8], we experimentally showed FCS is up to three orders of magnitude faster to evaluate than and achieves a Spearman correlation above 0.90 with the standard total variation distance between the learned and target distributions (**RQIII**).

Generalization bounds for GFlowNets. The fact that FCS is both local — depending only on random subsets of the state space — and bounded enable the analysis of GFlowNet generalization within the probably approximately correct framework of [McAllester 1999]. In [da Silva et al. 2025b], we build upon this insight to derive the first generalization bounds for GFlowNets demonstrating that these models learn a policy network capable of generalizing to unseen regions of the state graph (**RQIII**). Moreover, our results reveal that the diameter of the underlying state graph is a key factor limiting the generalization of GFlowNets. To alleviate this problem, we introduced SAL as a divide-and-conquer algorithm that decomposes the state graph into smaller, more manageable components: a separate GFlowNet is trained for each component in parallel and later combined in a final, efficient aggregation step. Our experimental evaluation in [da Silva et al. 2025b, Figures 6-10] show that SAL significantly outperforms a centralized, monolithic GFlowNet in terms of both FCS and traditional metrics (**RQII**).

6. Conclusions

We presented the first streaming, distributed, and asynchronous algorithms for GFlowNet training. Also, we established the theoretical foundations for a clearer understanding and fair assessment of GFlowNets. We believe our work will significantly broaden the applicability of these models to larger domains and inspire further theoretical advances.

References

[Ahmadian et al. 2024] Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. (2024). Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs.

[Association for Computing Machinery 2025] Association for Computing Machinery (2025). 2024 Turing Award. Accessed: 2025-03-15.

[Atanackovic and Bengio 2024] Atanackovic, L. and Bengio, E. (2024). Investigating generalization behaviours of generative flow networks.

[Bengio et al. 2021] Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. (2021). Flow network based generative models for non-iterative diverse candidate generation. In *NeurIPS*.

[Bengio 2022] Bengio, Y. (2022). System 2 Deep Learning: Higher-Level Cognition, Agency, Out-of-Distribution Generalization and Causality. In *IJCAI invited talk*.

[Bengio et al. 2023] Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. (2023). GFlowNet Foundations. *Journal of Machine Learning Research (JMLR)*.

[Bengio and Malkin 2024] Bengio, Y. and Malkin, N. (2024). Machine learning and information theory concepts towards an AI Mathematician.

[Blei and et al. 2017] Blei, D. M. and et al. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*.

[Boussif et al. 2024] Boussif, O., Ezzine, L. N., Viviano, J. D., Koziarski, M., Jain, M., Malkin, N., Bengio, E., Assouel, R., and Bengio, Y. (2024). Action abstractions for amortized sampling.

[Buesing et al. 2020] Buesing, L., Heess, N., and Weber, T. (2020). Approximate inference in discrete distributions with monte carlo tree search and value functions. In *AISTATS*.

[Cappello et al. 2021] Cappello, L., Kim, J., Liu, S., and Palacios, J. A. (2021). Statistical Challenges in Tracking the Evolution of SARS-CoV-2.

[Carpenter et al. 2017] Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., and Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*.

[da Silva et al. 2025a] da Silva, T., Alves, R., Silva, E., Souza, A., Garg, V., Kaski, S., and Mesquita, D. (2025a). When do GFlowNets learn the right Distribution? *ICLR*.

[da Silva et al. 2024a] da Silva, T., de Souza, D., and Mesquita, D. (2024a). Streaming Bayes GFlowNets. *NeurIPS*.

[da Silva et al. 2024b] da Silva, T., Silva, E., and Mesquita, D. (2024b). On Divergence Measures for Training GFlowNets. *NeurIPS*.

[da Silva et al. 2024c] da Silva, T., Souza, A., Carvalho, L., Kaski, S., and Mesquita, D. (2024c). Embarrassingly Parallel GFlowNets. *ICML*.

[da Silva et al. 2025b] da Silva, T., Souza, A., Rivasplata, O., Garg, V., Kaski, S., and Mesquita, D. (2025b). Generalization and Distributed Learning of GFlowNets. *ICLR*.

[Deleu and Bengio 2023] Deleu, T. and Bengio, Y. (2023). Generative Flow Networks: a Markov Chain Perspective.

[Deleu et al. 2022] Deleu, T., Góis, A., Emezue, C. C., Rankawat, M., Lacoste-Julien, S., Bauer, S., and Bengio, Y. (2022). Bayesian structure learning with generative flow networks. In *UAI*.

[Du and Kaelbling 2024] Du, Y. and Kaelbling, L. (2024). Compositional generative modeling: A single model is not all you need.

[Garipov et al. 2023] Garipov, T., Peuter, S. D., Yang, G., Garg, V., Kaski, S., and Jaakkola, T. (2023). Compositional sculpting of iterative generative processes. In *NeurIPS*.

[Geyer 1991] Geyer, C. J. (1991). Markov Chain Monte Carlo Maximum Likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pages 156–163.

[Grathwohl et al. 2021] Grathwohl, W., Swersky, K., Hashemi, M., Duvenaud, D., and Maddison, C. J. (2021). Oops I Took A Gradient: Scalable Sampling for Discrete Distributions.

[Hooker 2020] Hooker, S. (2020). The Hardware Lottery.

[Hu et al. 2023] Hu, E. J., Jain, M., Elmoznino, E., Kaddar, Y., and et al. (2023). Amortizing intractable inference in large language models.

[Kim et al. 2024a] Kim, M., Choi, S., Yun, T., Bengio, E., Feng, L., Rector-Brooks, J., Ahn, S., Park, J., Malkin, N., and Bengio, Y. (2024a). Adaptive teachers for amortized samplers.

[Kim et al. 2024b] Kim, M., Yun, T., Bengio, E., Dinghuai Zhang, Y. B., Ahn, S., and Park, J. (2024b). Local search GFlowNets. In *ICLR (ICLR)*.

[Kingma and Ba 2014] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[Kullback and Leibler 1951] Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*.

[Lahlou et al. 2023] Lahlou, S., Deleu, T., Lemos, P., Zhang, D., Volokhova, A., Hernández-García, A., Ezzine, L. N., Bengio, Y., and Malkin, N. (2023). A theory of continuous generative flow networks. In *ICML*.

[Lau et al. 2024] Lau, E., Lu, S. Z., Pan, L., Precup, D., and Bengio, E. (2024). QGFN: Controllable Greediness with Action Values.

[Lindley 1972] Lindley, D. V. (1972). *Bayesian statistics: A review*. SIAM.

[Liu and Liu 2001] Liu, J. S. and Liu, J. S. (2001). *Monte Carlo strategies in scientific computing*, volume 10. Springer.

[Lázaro-Gredilla et al. 2025] Lázaro-Gredilla, M., Ku, L. Y., Murphy, K. P., and George, D. (2025). What type of inference is planning?

[Malkin et al. 2022] Malkin, N., Jain, M., Bengio, E., Sun, C., and Bengio, Y. (2022). Trajectory balance: Improved credit assignment in GFlownets. In *NeurIPS*.

[Malkin et al. 2023] Malkin, N., Lahlou, S., Deleu, T., Ji, X., Hu, E., Everett, K., Zhang, D., and Bengio, Y. (2023). GFlowNets and variational inference. *ICLR (ICLR)*.

[McAllester 1999] McAllester, D. A. (1999). PAC-Bayesian model averaging. In *COLT*, pages 164–170.

[Neal et al. 2011] Neal, R. M. et al. (2011). MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*.

[Owen 2013] Owen, A. B. (2013). *Monte Carlo theory, methods and examples*.

[Pandey et al. 2024] Pandey, M., Subbaraj, G., and Bengio, E. (2024). GFlowNet Pretraining with Inexpensive Rewards. *arXiv preprint arXiv:2409.09702*.

[Parr et al. 2022] Parr, T., Pezzulo, G., and Friston, K. J. (2022). *Active Inference: The Free Energy Principle in Mind, Brain, and Behavior*. The MIT Press, Cambridge, MA.

[Rényi 1961] Rényi, A. (1961). On measures of entropy and information. In *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. University of California Press.

[Rhodes and Gutmann 2019] Rhodes, B. and Gutmann, M. U. (2019). Variational noise-contrastive estimation. In *AISTATS*.

[Richter et al. 2020] Richter, L., Boustati, A., Nüsken, N., Ruiz, F. J. R., and Akyildiz, Ö. D. (2020). VarGrad: A low-variance gradient estimator for variational inference. In *NeurIPS*.

[Robert et al. 2007] Robert, C. P. et al. (2007). *The Bayesian choice: from decision-theoretic foundations to computational implementation*, volume 2. Springer.

[Roy 2020] Roy, V. (2020). Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, 7(1):387–412.

[Schulman et al. 2017] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

[Shen et al. 2023] Shen, M. W., Bengio, E., Hajiramezanali, E., Loukas, A., Cho, K., and Biancalani, T. (2023). Towards Understanding and Improving GFlowNet Training. In *ICML*.

[Sutton 2019] Sutton, R. (2019). The bitter lesson. *Incomplete Ideas (blog)*, 13(1):38.

[Sutton and Barto 2018] Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2nd edition.

[Zhang et al. 2023a] Zhang, D., Dai, H., Malkin, N., Courville, A., Bengio, Y., and Pan, L. (2023a). Let the flows tell: Solving graph combinatorial optimization problems with gflownets. In *NeurIPS*.

[Zhang et al. 2023b] Zhang, D. W., Rainone, C., Peschl, M., and Bondesan, R. (2023b). Robust scheduling with GFlowNets. In *ICLR (ICLR)*.

[Zhou et al. 2024] Zhou, M. Y., Yan, Z., Layne, E., Malkin, N., Zhang, D., Jain, M., Blanchette, M., and Bengio, Y. (2024). PhyloGFN: Phylogenetic inference with generative flow networks. In *ICLR*.