FUNDAÇÃO GETULIO VARGAS
SCHOOL OF APPLIED MATHEMATICS

TIAGO DA SILVA

STREAMING, DISTRIBUTED, AND ASYNCHRONOUS AMORTIZED INFERENCE

Rio de Janeiro
2024

TIAGO DA SILVA

STREAMING, DISTRIBUTED, AND ASYNCHRONOUS AMORTIZED INFERENCE

Doctoral thesis presented to the School of Applied Mathematics (FGV/EMAp) to obtain the degree of Doctor of Science in Applied Mathematics and Data Science.

Area of Study: Machine Learning.

Advisor: Diego Mesquita

Rio de Janeiro
2024

Ficha catalográfica elaborada pela BMHS/FGV

da Silva, Tiago
Streaming, Distributed, and Asynchronous
Amortized Inference/ Tiago da Silva. – 2024.
124f.

Doctoral thesis – School of Applied Mathematics.

Advisor: Diego Mesquita.
Includes bibliography.

1. Amortized Inference. 2. Generative Flow Networks. 3. Distributed algorithms. I. Mesquita, Diego. II. School of Applied Mathematics. III. Streaming, Distributed, and Asynchronous
Amortized Inference

Acknowledgements

I am grateful to everyone that helped in the development of this thesis. In particular, I would like to thank my coauthors, Diego, Amauri, Luiz, Vikas, Eliezer, Omar, Rodrigo, Daniel, and Sami, for their invaluable feedback and suggestions for improving the clarity and strenghtening the arguments of the text. During this period, I was financially backed by the School of Applied Mathematics, the Center for the Development of Mathematics and Science (CDMC, in Portuguese), and the São Paulo Research Foundation (FAPESP, in Portuguese). I am also thankful to FGV's TIC and Aalto Scientific Computing's team, who provisioned the computational resources required for our experiments.

The works in this thesis were funded by the Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro FAPERJ (SEI-260003/000709/2023), the São Paulo Research Foundation FAPESP (2023/00815-6), and the Conselho Nacional de Desenvolvimento Científico e Tecnológico CNPq (404336/2023-0).

But one learns from books and reels only that
certain things can be done. Actual learning requires that you do those things.
Children of Dune, Frank Herbert

Abstract

We address the problem of sampling from an unnormalized distribution defined in a *compositional space*, i.e., a continuous or discrete set whose elements can be sequentially constructed from an initial state through the application of simple actions. This definition accommodates the space of (directed acyclic) graphs, natural language sentences of bounded size, and Euclidean n-spaces, among others, and is at the core of many applications in (Bayesian) statistics and machine learning. In particular, we focus on Generative Flow Networks (GFlowNets), a family of amortized samplers which cast the problem of sampling as finding a flow assignment in a flow network such that the total flow reaching a sink node equals that node's unnormalized probability. Despite their remarkable success in drug discovery, structure learning, and natural language processing, important questions regarding the scalability, generalization, and limitations of these models remain largely underexplored by the literature. In view of this, this thesis contributes with both methodological and theoretical advances for a better usability and understanding of GFlowNets.

From a computational perspective, we design novel algorithms for the non-localized training of GFlowNets. This enables learning these models in a streaming and distributed fashion, which is crucial for managing ever-increasing data sizes and exploiting the architecture of modern computer clusters. The central idea of our methods is to break up the flow assignment problem into easier subproblems solved by separately trained GFlowNets. Once trained, these models are aggregated by a global GFlowNet. To do so efficiently, we also revisit the relationship between GFlowNets and variational inference and devise low-variance estimators for their learning objective's gradients to achieve faster training convergence. Overall, our experiments show that our non-localized procedures often lead to better approximations in a shorter time relatively to a centralized monolithic GFlowNet.

Additionally, we demonstrate that the models corresponding to the global minimizers of the proposed surrogate learning objectives sample in proportion to the unnormalized target. This fact raises the questions of *when* a GFlowNet can reach such a global minimum and *how close* a trained model is to it. Towards answering them, we first present a family of discrete distributions that cannot be approximated by a GFlowNet when the flow functions are parameterized by 1-WL graph neural networks. Then, we develop a computationally amenable metric to probe the distributional accuracy of GFlowNets. Finally, as GFlowNets rely exclusively on a subgraph of the (potentially huge) flow network to learn a flow assignment, we argue that *generalization* plays a critical role in their success and derive the first non-vacuous (PAC-Bayesian) statistical guarantees for these models.

Keywords: GFlowNets, Bayesian inference, distributed methods, geometric deep learning.

Resumo

Nós endereçamos o problema de amostragem de uma distribuição não normalizada definida em um *espaço composicional*, i.e., um conjunto contínuo ou discreto cujos elementos podem ser construídos sequencialmente a partir de um estado inicial por meio da aplicação de ações simples. Esta definição abrange o espaço de grafos (acíclicos direcionados), sentenças em linguagem natural de tamanho limitado e espaços euclidianos de dimensão n, entre outros, e é central em muitas aplicações em estatística (Bayesiana) e aprendizado de máquina. Em particular, nós focamos em Generative Flow Networks (GFlowNets), uma família de amostradores amortizados que formulam o problema de amostragem como a busca por uma atribuição de fluxo em uma rede de fluxo tal que o volume total chegando a um nó de sumidouro seja igual à probabilidade não normalizada desse nó. Apesar de seu sucesso notável em descoberta de medicamentos, aprendizado de estrutura e processamento de linguagem natural, questões importantes sobre escalabilidade, generalização e limitações desses modelos permanecem amplamente inexploradas na literatura. Assim, esta tese contribui com avanços metodológicos e teóricos para uma melhor usabilidade e compreensão de GFlowNets.

Sob uma perspectiva computacional, projetamos novos algoritmos para o treinamento não localizado de GFlowNets. Isso permite o aprendizado desses modelos de forma dinâmica e distribuída, o que é crucial para lidar com o aumento constante no tamanho dos conjuntos de dados e para aproveitar a arquitetura dos modernos e poderosos clusters de computadores. Em resumo, a ideia central de nossos métodos consiste em dividir o problema de atribuição de fluxo em subproblemas mais simples, que são resolvidos por GFlowNets treinadas separadamente. Uma vez treinados, esses modelos são agregados por uma GFlowNet global. Para fazer isso de maneira eficiente, também revisitamos a relação entre GFlowNets e inferência variacional, desenvolvendo estimadores de baixa variância para os gradientes da sua função de perda e, em consequência, acelerando a convergência do treinamento. Além disso, nossos experimentos mostram que nosso procedimento não localizado frequentemente leva a melhores aproximações em um tempo mais curto em relação a uma GFlowNet monolítica e centralizada.

Importantemente, também demonstramos que os modelos correspondentes aos minimizadores globais dos objetivos de aprendizado propostos amostram corretamente da distribuição alvo não normalizada. Isso levanta naturalmente as questões de *quando* uma GFlowNet pode alcançar esse mínimo global e *quão próximo* está um dado modelo desse ótimo. Para responder a essas perguntas, primeiro construímos uma família explícita de distribuições discretas que não podem ser aproximadas por uma GFlowNet quando as funções de fluxo são parametrizadas por redes neurais de grafos com expressividade 1-WL. Em seguida, desenvolvemos uma métrica computacionalmente viável para investigar a acurácia distribucional das GFlowNets. Por fim, como as GFlowNets utilizam apenas um subgrafo da (geralmente enorme ou infinita) rede de fluxo para aprender uma atribuição de fluxo, nós argumentamos que a *generalização* desempenha um papel crítico em seu sucesso e derivamos as primeiras garantias estatísticas não vazias para esses modelos.

Palavras-chave: GFlowNets, inferência bayesiana, métodos distribuídos, aprendizado profundo geométrico.

List of Figures

List of Tables

List of Contributions

The contents in this thesis were independently published in the following works.

I Tiago da Silva, Amauri Souza, Luiz Carvalho, Samuel Kaski, and Diego Mesquia. Embarrassingly Parallel GFlowNets. In *International Conference on Machine Learning*, 2024.

II Tiago da Silva, Daniel Augusto de Souza, Diego Mesquita. Streaming Bayes GFlowNets. In *Advances in Neural Information Processing Systems*, 2024.

III Tiago da Silva, Eliezer de Souza da Silva, Diego Mesquita. On Divergence Measures for Training GFlowNets. In *Advances in Neural Information Processing Systems*, 2024.

IV Tiago da Silva, Amauri Souza, Omar Rivasplata, Vikas Garg, Samuel Kaski, and Diego Mesquita. Generalization and Distributed Learning of GFlowNets. *Under review*, 2024.

V Tiago da Silva, Amauri Souza, Vikas Garg, Samuel Kaski, and Diego Mesquita. When do GFlowNets learn the right distribution? *Under review*, 2024.

We will often refer to these works throughout the text for further details on both the theoretical and empirical results, which are freely available in the internet.

Contents

1 Introduction

Sampling from an unnormalized distributions is a fundamental problem in statistics and machine learning without a clear one-size-fits-all solution (LIU; LIU, 2001; BLEI; AL., 2017; BUESING; HEESS; WEBER, 2020). While Markov Chain Monte Carlo (MCMC) methods have demonstrated exceptional performance in a diverse range of applications, they frequently suffer from slow mixing times and are notoriously hard to diagnose (ROY, 2020). These problems become critical for discrete spaces, which lack a differential structure that exempts the use of the celebrated Hamiltonian Monte Carlo (HMC) algorithm (NEAL et al., 2011; CARPENTER et al., 2017). In light of these issues, (BENGIO; JAIN, et al., 2021; BENGIO; LAHLOU, et al., 2023) introduced Generative Flow Networks (GFlowNets) as an alternative to Markov chain methods to address the sampling problem.

In a nutshell, a GFlowNet solves the *flow assignment problem* in a single-source *flow network* whose sink nodes correspond to the support of the target distribution. When the flow reaching each sink node is constrained to be the node's unnormalized probability, we obtain samples from the target distribution by starting at the network's source and proceeding to the next node by choosing each transition with probability proportional to the flow therein. To search for such a flow assignment, we parameterize a *flow function* with neural networks and minimize via stochastic gradient descent (SGD) a stochastic objective enforcing a *balance condition*, which ensures the incoming and outgoing flows are the same in each node. In Chapter 2, we formalize this intuitive description in the language of Markov Decision Processes (MDPs) (LAHLOU et al., 2023).

Remarkably, GFlowNets were very successful in finding a good solution to difficult problems such as drug discovery (BENGIO; JAIN, et al., 2021; PANDEY; SUBBARAJ; BENGIO, 2024a), phylogenetic inference (ZHOU et al., 2024; SILVA et al., 2024), language processing (HU et al., 2023), combinatorial optimization (ZHANG, D. W. et al., 2023; ZHANG; DAI, et al., 2023), and structure learning (DELEU; GÓIS, et al., 2022; DELEU; NISHIKAWA-TOOMEY, et al., 2023). Nonetheless, most of these tasks are of relatively small-scale and the scaling of GFlowNets to larger problems remains a challenging endeavor. From a statistical viewpoint, this scaling can be achieved through more sample-efficient learning objectives (**RQI**), which we have pursued in Publications **I** and **III**.

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, we have historically observed 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. At the very least, this allows for more computation to be performed in less time in the pursuit of solving the flow assignment problem. A related issue concerns the *reuse* of expensively trained GFlowNets to solve future problems (e.g., in a streaming Bayes context). Although it is mostly unclear how to generally achieve this flexibility due to the complex nature of the flow network and of the target distribution (**RQII**), Publications **I**, **II**, and **IV** propose strategies for the streaming, distributed, and asynchronous learning of GFlowNets in specific settings.

Research Question II: *The success of contemporary machine learning is largely due to the huge computational resources available in modern computer clusters. How to adapt the training of GFlowNets to fit in distributed and continual learning paradigms?*

Strikingly, the theoretical properties of GFlowNet learning have been mostly underexplored in the literature. Indeed, the foundational work of (BENGIO; LAHLOU, 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 hypothesis space (e.g., fixed-width and fixed-depth MLPs) via a chosen learning algorithm (e.g., SGD). 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; YUN; BENGIO; DINGHUAI ZHANG, et al., 2024; LAU et al., 2024). Correspondingly, although the empirical effectiveness of GFlowNets has been attributed to their generalization capabilities (BENGIO; JAIN, et al., 2021; PANDEY; SUBBARAJ; BENGIO, 2024a; ATANACKOVIC; BENGIO, 2024), no work so far has provided non-vacuous statistical guarantees for the population risk of GFlowNets (**RQIII**). In this scenario, Publications **IV** and **V** present the first theoretical results on the limitations 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?*

Thesis organization. Chapter 2 introduces the notations and terminologies used throghout this thesis. Chapter 3 revisits the relationship between GFlowNets and (hierarchical) variational inference and introduces low-variance gradient estimators for divergence-inspired learning objectives. Chapter 4 builds upon these estimators to develop an efficient algorithm for training GFlowNets in a streaming Bayesian context, i.e., when the posterior distribution changes over time. In a similar fashion, Chapter 5 outlines a GFlowNet-based divide-and-conquer method to perform embarrassingly parallel Bayesian inference in discrete parameters spaces. Chapter 6 takes a closer look at the distributional limits and the assessment of GFlowNets. Chapter 7 derives the first non-vacuous generalization bounds for GFlowNets and proposes a distributed approach for learning a flow assignment. Chapter 8 summarizes our contributions and discusses potential directions for future research.

2 Preliminaries

This chapter presents the background material for this thesis. For the most part, we focus on distributions on finite spaces. In Chapter 3, we briefly discuss an extension of this formalism to the context of probability measures supported on uncountably infinite sets. Readers may consult our publications and the references therein for additional information regarding GFlowNets, geometric deep learning, and (PAC-)Bayesian theory.

2.1 Notations and terminology

Markov Decision Processes (MDPs). Let $\mathcal{X}$ be a finite set and $R \colon \mathcal{X} \rightarrow \mathbb{R}_+$ be a positive function on $\mathcal{X}$, which is called a *reward function* due to the terminological inheritance from the reinforcement learning 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* of $\mathcal{G}$ and $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ be a directed acyclic graph (DAG) with edges $\mathcal{E}$, which we call the *state graph*. 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 $x \in \mathcal{X}$. For this reason, we refer to $\mathcal{X}$ as the set of *terminal states*. A *forward policy* $p_F \colon \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+$ 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* $p_B \colon \mathcal{S} \times \mathcal{S} \rightarrow \mathbb{R}_+$ is a forward policy on the transpose $\mathcal{G}^\top$ of $\mathcal{G}$. 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. $p_B = \prod_{i=2}^{n} p_B(s_i|s_{i-1})$) for $\tau = (s_1, \ldots, s_n)$. We call a trajectory *complete* when it starts at s_o and finishes at s_f. We refer to the tuple $(\mathcal{G}, p_F, p_B, R)$ as a *Markov decision process.*

Under the hermeneutics of flow networks, the forward policy $p_F(s, s')$ measures the relative capacity of the edge $s \rightarrow s'$ with respect to the flow in s. Similarly, $R(x)$ represents the desired amount of flow reaching each $x \in \mathcal{X}$. In this context, we define a *flow function* $F \colon \mathcal{S} \rightarrow \mathbb{R}_+$ 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 $\mathrm{Ch}(s) = \{s' \colon s \rightarrow s' \in \mathcal{E}\}$ denotes the children of s in $\mathcal{G}$.

2.2 GFlowNets

GFlowNets. Generative Flow Networks (BENGIO; JAIN, et al., 2021; BENGIO; LAHLOU, et al., 2023; LAHLOU et al., 2023, GFlowNets) are a family $\{(\mathcal{G}, p_F^\theta, p_B^\theta, R, F^\theta) \colon \theta \in \Theta\}$ composed of a MDP $(\mathcal{G}, p_F^\theta, p_B^\theta, R)$ and a flow function F^θ parameterized by a $\theta \in \Theta$, which often corresponds to the weights of a neural network with fixed architecture. The learning objective of a GFlowNet is to find a p_F^θ such that the marginal distribution $p_\top^\theta$ of $p_F^\theta(\cdot|s_o)$ on $\mathcal{X}$ equals R up to a normalizing constant, i.e.,

$$p_\top^\theta(x) \coloneqq \sum_{\tau\colon s_o \rightsquigarrow x} p_F^\theta(\tau|s_o) \propto R(x) \tag{2.1}$$

for $x \in \mathcal{X}$; $\tau\colon s_o \rightsquigarrow x$ means that the trajectory τ starts at s_o and finishes at x. When there is no risk of ambiguity, we will omit θ when referring to a GFlowNet. Clearly, samples from $p_\top$ can be efficiently generated by starting at s_o and following p_F until reaching a $x \in \mathcal{X}$.

Training GFlowNets. (BENGIO; LAHLOU, et al., 2023) proved that to achieve Equation (2.1) it is sufficient to ensure that flow network $(\mathcal{G}, p_F, p_B, F)$ is balanced. As mentioned, the search for a balanced flow assignment consists of minimizing a stochastic objective enforcing a balance condition. More precisely, let p_E be a forward *exploratory* policy in $\mathcal{G}$. The *trajectory balance* (TB) condition and the TB loss are respectively defined as

$$F(s_o)p_F(\tau|s_o) = F(x)p_B(\tau|x) \text{ and } \mathcal{L}_{\text{TB}}(F, p_F, p_B) = \mathbb{E}_{\tau\sim p_E}\left[\left(\log \frac{p_F(\tau|s_o)F(s_o)}{p_B(\tau|x)R(x)}\right)^2\right];$$

the former must hold for every trajectory τ and the latter is only valid when p_E assigns positive probability to each τ (MALKIN; JAIN, et al., 2022). Since $F(s_o) = \sum_{x\in\mathcal{X}} R(x)$ corresponds to the partition function of the distribution over $\mathcal{X}$ induced by R, we also denote $F(s_o) = Z$. Similarly, the *detailed balance* (DB) condition and DB loss are characterized by

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

in which $\#\tau$ represents the number of transitions in τ (BENGIO; LAHLOU, et al., 2023). Finally, the *subtrajectory balance* (SubTB) condition and the SubTB loss interpolate between their DB and TB counterparts via a parameter λ (MADAN et al., 2022). For this, let

$$\mathcal{L}_{\text{SubTB}}^{i,j}(\tau) = \left(\log \frac{F(\tau_i)p_F(\tau_{i:j}|\tau_i)}{F(\tau_j)p_B(\tau_{j:i}|\tau_j)}\right)^2 \text{ and } \mathcal{L}_{\text{SubTB}}(F, p_F, p_B) = \mathop{\mathbb{E}}_{\tau\sim p_E}\left[\frac{\left(\lambda^{j-i}\mathcal{L}_{\text{SubTB}}^{i,j}(\tau)\right)}{\sum_{1\le i<j\le \#\tau+1}\lambda^{j-i}}\right].$$

Notably, $\mathcal{L}_{\text{SubTB}} = \mathcal{L}_{\text{DB}}$ for $\lambda \to 0$ and $\mathcal{L}_{\text{SubTB}} = \mathcal{L}_{\text{TB}}$ for $\lambda \to \infty$. In practice, p_E is often set as an α-greedy version of p_F, namely, $p_E = \alpha p_F + (1-\alpha)p_U$, in which p_U is the uniform policy in $\mathcal{G}$. Other strategies for sampling trajectories, e.g., replay buffer (VEMGAL; LAU; PRECUP, 2023) and local search (KIM; YUN; BENGIO; DINGHUAI ZHANG, et al., 2024), have also been developed. We refer to the choice of $p_E \neq p_F$ to sample trajectories during training as an *off-policy approach*. Finally, we frequently use the total variation (TV) distance to quantify the distributional accuracy of a trained GFlowNet. For probability measures p and q on the set $\mathcal{X}$, we define

$$\text{TV}\,(p, q) \coloneqq \sup_{A\subseteq\mathcal{X}} |p(A) - q(A)| = \frac{1}{2}\sum_{x\in\mathcal{X}} |p(x) - q(x)|, \tag{2.2}$$

in which $p(A) = \sum_{x\in\mathcal{X}} p(x)$ is the measure of A under the probability measure p.

2.3 Graph neural networks

Graph neural networks. Graph neural networks (GNNs) are the leading paradigm for graph representation learning (HAMILTON, 2020; WANG; VELIČKOVIĆ, et al., 2024; CORSO et al., 2024). Most GNNs employ a multi-layered message-passing (MP) scheme that interleaves neighborhood aggregation and update operations at each layer: for each node v at layer ℓ, the aggregation is a nonlinear function of the $(\ell-1)$-layer representations of v's neighbors. The update step computes a new representation for v based on its representation at layer $\ell-1$ and the aggregated messages (output of the aggregation step). The resulting representation is then inputted to a predictive model to solve a downstream task.

Expressive power of GNNs. Despite their notable accomplishments, MP-GNNs have inherently limited expressivity, namely, there are non-isomorphic graphs (depending on the architecture) for which a GNN computes the same representation. This should be contrasted with the universality of MLPs (HORNIK; STINCHCOMBE; WHITE, 1989) and the Turing completeness of RNNs (SIEGELMANN; SONTAG, 1992). The expressive power of GNNs is often quantified in terms of the Weisfeiler-Leman hierarchy, which refers to a family of color refinement algorithms (WEISFEILER; LEHMAN, 1968) for addressing the graph isomorphism problem, albeit alternative frameworks have been proposed (ZHANG, B. et al., 2023; WANG, Q. et al., 2023; GRAZIANI et al., 2024). Importantly, the expressivity of popular GNN architectures (KIPF; WELLING, 2016; VELIČKOVIĆ et al., 2017; XU; HU, et al., 2019) is upper-bounded by the 1-WL isomorphism test, which we review in Appendix A.1. Chapter 6 leverages this understanding to delineate the distributional limits of GFlowNets parameterized by MP-GNNs.

2.4 Generalization bounds for neural networks

PAC-Bayesian theory. Statistical learning theory (VAPNIK, V., 1998; VAPNIK, V. N., 2000) studies the development of statistical certificates for learning algorithms. This is often achieved by designing a high-probability bound of the population error of an estimator as a function of the observed empirical error. In the context of GFlowNets, our interest lies in estimating the difference between the true intractable distributional accuracy of a GFlowNet and its empirical counterpart. This discrepancy is called the *generalization gap*. We are particularly interested in *inductive* statistical guarantees for the generalization gap, namely, those based on a training set (instead of on a test set). To address this issue, McAllester's PAC-Bayes framework (MCALLESTER, D. A., 1998, 1999; MCALLESTER, D., 2013) provides the tightest bounds (LOTFI; FINZI, et al., 2024; LOTFI; KUANG, et al., 2024; DZIUGAITE; ROY, 2017, 2018). Briefly, consider data $\mathbf{X} = \{X_i\}_{i=1}^{m}$ drawn from some data distribution, a significance level δ, an empirical loss $\hat{\mathcal{L}}(\theta, \mathbf{X})$ and a population loss $\mathcal{L}(\theta) = \mathbb{E}_{\mathbf{X}}[\hat{\mathcal{L}}(\theta, \mathbf{X})]$ associated to the model's parameters θ. Given a “prior" (independent of $\mathbf{X}$) distribution Q over θ, a PAC-Bayes bound typically assumes the form

$$\mathbb{E}_{\theta\sim P}[\mathcal{L}(\theta)] \leq \mathbb{E}_{\theta\sim P}[\hat{\mathcal{L}}(\theta, \mathbf{X})] + \phi(\delta, P, Q, m). \tag{2.3}$$

The inequality holds with probability $1 - \delta$ over draws of $\mathbf{X}$ simultaneously for all "posterior" distributions P over θ. Analogously to the Vapnik-Chervonenkis' dimension (VAPNIK; CHERVONENKIS, 2015), ϕ is a term penalizing the model's complexity (MCALLESTER, D. A., 1999; CATONI, 2007). Readers may consult (CATONI, 2007; GUEDJ, 2019; ALQUIER, 2024) for an introduction to PAC-Bayesian analysis.

Data-dependent priors for PAC-Bayes bounds. When the prior Q is naively chosen (e.g., a standard Gaussian), Equation (2.3) is frequently *vacuous.* In other words, the right-hand side of Equation (2.3) can be larger than an upper bound of the loss $\mathcal{L}(\theta)$, and the inequality becomes trivially true. This issue is of particular concern when θ is embedded in a high-dimensional space , in which case the complexity term ϕ may be orders of magnitude larger than the empirical loss (DZIUGAITE; ROY, 2017). In view of this, Dziugaite et al. (DZIUGAITE; ROY, 2018; DZIUGAITE; HSU, et al., 2020) proposed the use of *data-dependent priors* Q estimated on a subset of training data and *regularized posteriors* P learned by minimizing the upper bound in Equation (2.3). A tight statistical certificate is then obtained by evaluating Equation (2.3) on the data points unused for learning Q. With this approach, summarized in Proposition 2.4.1, (DZIUGAITE; HSU, et al., 2020) derived non-vacuous generalization bounds even for overparameterized neural networks.

Proposition 2.4.1 (Data-dependent PAC-Bayesian priors)**.** *Assume $\mathcal{L}(\theta, \mathbf{X}) \in [0, 1]$ for all θ and $\mathbf{X}$. For any distribution ζ on parameters θ, let $\mathcal{L}(\zeta) = \mathbb{E}_{\theta\sim\zeta}[\mathcal{L}(\theta)]$ and define $\hat{\mathcal{L}}(\zeta, \mathbf{X})$ similarly. Also, let $\alpha \in (0, 1)$, P be a distribution on θ, and $n = |\mathbf{X}|$ be the number of data points. Denote by $\mathbf{X}_{1-\alpha}$ an uniformly random $\lfloor(1-\alpha)n\rfloor$-sized subset of $\mathbf{X}$. Then,*

$$\mathcal{L}(P) \leq \hat{\mathcal{L}}(P, \mathbf{X}_{1-\alpha}) + \min \begin{cases} \eta + \sqrt{\eta(\eta + 2\hat{\mathcal{L}}(P, \mathbf{X}_{1-\alpha}))}, \\ \sqrt{\frac{\eta}{2}}, \end{cases} \tag{2.4}$$

with probability at least $1 - \delta$ over $\mathbf{X}_{1-\alpha}$, in which $\eta \coloneqq \frac{\mathrm{KL}(P||Q)+\log 2\sqrt{\lfloor(1-\alpha)n\rfloor}/\delta}{\lfloor(1-\alpha)n\rfloor}$ and Q is a distribution that does not depend on $\mathbf{X}_{1-\alpha}$ but may depend on $\mathbf{X}_\alpha \coloneqq \mathbf{X} \setminus \mathbf{X}_{1-\alpha}$.

The choice of α is a trade-off between the size of the data for evaluating the generalization gap in Equation (2.4) and the precision of the prior distribution with respect to P. For completeness, we provide a proof of Proposition 2.4.1 in Appendix B. Chapter 7 builds upon this result to derive the first non-vacuous statistical guarantees for GFlowNets.

3 On Divergence Measures for Training GFlowNets

Malkin et al. showed that GFlowNets are describable as hierarchical variational inference (HVI) algorithms and can be trained via the minimization of a statistical divergence between the forward and backward policies (MALKIN; LAHLOU, et al., 2023). Nonetheless, this approach was regarded as clearly inferior than minimizing the traditional balance-based objectives presented in Chapter 2 (MALKIN; LAHLOU, et al., 2023).

In this scenario, we empirically demonstrate that the difficult for training GFlowNets in a HVI fashion arises from the large gradient variance of the divergence-based loss functions. For this, we show that learning can be drastically accelerated by using a low-variance and cheap-to-compute gradient estimator for GFlowNets. The resulting approach, which performs competitively with or better than their traditional counterparts, is utilized in later chapters for the design of non-localized training algorithms for GFlowNets (see Chapter 4).

This chapter is thus focused on **RQI**. First, Section 3.1 reviews the concepts of measurable pointed DAGs (MP-DAGs) and continuous GFlowNets introduced by (LAHLOU et al., 2023), which may be skipped by the knowledgeable reader. Then, Section 3.2 revisits the relationship between GFlowNets and HVI and presents a series of statistical divergence and their corresponding gradient estimators. Finally, Section 3.3 devises *control variates* (CVs) for low-variance gradient esimation of a GFlowNet's learning objective, while Section 3.4 validates the ensuing algorithm in a comprehensive range of experiments.

3.1 Continuous GFlowNets

The transition from a discrete to a continuous GFlowNet requires that we abandon the notion of a state graph in favor of a *reference kernel* representing the connectedness of a possibly uncountably infinite state space. We start by fixing some notations.

Notations. Let $(\mathcal{S}, \mathcal{T})$ be a topological space with topology $\mathcal{T}$ and Σ be the corresponding Borel σ-algebra, and let $\nu \colon \Sigma \rightarrow \mathbb{R}_+$ be a measure over Σ and $\kappa_f, \kappa_b \colon \mathcal{S} \times \Sigma \rightarrow \mathbb{R}_+$ be transition kernels over $\mathcal{S}$. For each $(B_1, B_2) \in \Sigma \times \Sigma$, we denote by $\nu \otimes \kappa(B_1, B_2) \coloneqq \int_{B_1} \nu(\mathrm{d}s) k(s, B_2)$. Likewise, we recursively define the *product kernel* as $\kappa^{\otimes 0}(s, \cdot) = \kappa(s, \cdot)$ and, for $n \geq 1$, $\kappa^{\otimes n}(s, \cdot) = \kappa^{\otimes n-1}(s, \cdot) \otimes \kappa$ for a transition kernel κ and $s \in \mathcal{S}$. In particular, $\kappa^{\otimes n}$ is a function from $\mathcal{S} \times \Sigma^{\otimes n+1}$ to $\mathbb{R}_+$ with $\Sigma^{\otimes n+1}$ representing the product σ-algebra of Σ (WILLIAMS, D., 1991; AXLER, 2020). Moreover, if μ is an absolutely continuous measure relatively to ν, denoted $\mu \ll \nu$, we write ${}^{\mathrm{d}\mu}/_{\mathrm{d}\nu}$ for the corresponding density (Radom-Nikodym derivative) (AXLER, 2020). We also let $[d] = \{1, \dots, d\}$.

Continuous GFlowNets. We first define a measurable pointed DAG (MP-DAG).

Definition 3.1.1 (Measurable pointed DAG (LAHLOU et al., 2023))**.** Let $(\bar{\mathcal{S}}, \mathcal{T}, \Sigma)$ be a measurable topological space endowed with a reference measure ν and *forward* κ_f and *backward* κ_b kernels. Also, let $s_o \in \bar{\mathcal{S}}$ and $s_f \in \bar{\mathcal{S}}$ be distinguished elements in $\bar{\mathcal{S}}$, called *initial* and *final* states, and $\mathcal{S} = \bar{\mathcal{S}} \setminus \{s_f\}$. We assume that $\{s_f\}$ is open. A *measurable pointed DAG* (MP-DAG) is a tuple $(\mathcal{S}, \mathcal{T}, \Sigma, \kappa_f, \kappa_b, \nu)$ satisfying the properties below.

1. **(Terminality)** If $\kappa_f(s, \{s_f\}) > 0$, then $\kappa_f(s, \mathcal{S}) = 0 \; \forall s \in \bar{\mathcal{S}}$. Also, $\kappa_f(s_f, \cdot) = \delta_{s_f}$.
2. **(Reachability)** $B \in \Sigma$, $\exists \, n \in \mathbb{N}$ s.t. $\kappa_f^{\otimes n}(s_o, B) > 0$, i.e., B is reachable from s_o.
3. **(Consistency)** For every $(B_1, B_2) \in \Sigma \times \Sigma$ such that $(B_1, B_2) \notin \{(s_o, s_o), (s_f, s_f)\}$, $\nu \otimes \kappa_f(B_1, B_2) = \nu \otimes \kappa_b(B_2, B_1)$. Moreover, $\kappa_b(s_o, B) = 0$ for every $B \in \Sigma$.
4. **(Continuity)** $s \mapsto \kappa_f(s, B)$ is continuous for $B \in \Sigma$.
5. **(Finite absorption)** There is a $N \in \mathbb{N}$ such that $\kappa_f^{\otimes N}(s, \cdot) = \delta_{s_f}$ for every $s \in \mathcal{S}$, i.e., the MP-DAG is *finitely absorbing* and s_f is its only absorbing state.

The elements in $\mathcal{X} = \{s \in \mathcal{S} \setminus \{s_f\} \colon \kappa_f(s, \{s_f\}) > 0\}$ are called *terminal states.* Illustratively, when $\mathcal{S}$ is finite, Σ is the discrete σ-algebra of $\mathcal{S}$, and ν is the counting measure, an MP-DAG is reduced to a standard DAG; if $\mathcal{S} = \{s_1, \dots, s_n\}$, its adjacency matrix $\mathbf{A} \in \{1, 0\}^{n \times n}$ is $\mathbf{A}_{ij} = \mathbb{1}_{\{\kappa_f(s_i, \{s_j\}) > 0\}}$ and condition (5) implies acyclicity. As their discrete equivalent, a *continuous GFlowNet* is a parametric family composed by a MP-DAG, a flow function, and a pair of Markovian kernels.

Definition 3.1.2 (GFlowNets (LAHLOU et al., 2023))**.** *Continuous GFlowNets* are a parametric family $\{(\mathcal{G}, P_F^\theta, P_B^\theta, \mu^\theta) \colon \theta \in \Theta\}$ composed of a MP-DAG $\mathcal{G}$, a σ-finite measure $\mu^\theta \ll \nu$, σ-finite Markovian kernels $P_F^\theta \ll \kappa_f$ and $P_B^\theta \ll \kappa_b$, called *forward* and *backward* policies, and a parameter space Θ. We indistinguishably use the term *continuous GFlowNet* to refer to this family and to its individual members.

Recall that $P_F^\theta \ll \kappa_f$ (resp. $P_B^\theta \ll \kappa_b$) means that $P_F^\theta(s, \cdot) \ll \kappa_f(s, \cdot)$ (resp. $P_B^\theta(s, \cdot) \ll \kappa_b(s, \cdot)$) for all $s \in \bar{\mathcal{S}}$. Henceforth, we will omit θ when ambiguity is not an issue.

Training continuous GFlowNets. We first let $\mathcal{P}_{\mathcal{S}} = \bigcup_{n=1}^{N+1} \bar{\mathcal{S}}^{\otimes n}$ be the space of *trajectories* in the MP-DAG and $\Sigma_P = \sigma\left(\bigcup_{n=1}^{N+1} \Sigma^{\otimes n}\right)$ be the corresponding σ-algebra; P_F, P_B, κ_f, and κ_b are uniquely extended to this space in the usual way via Caratheodory's extension theorem (WILLIAMS, D., 1991). For instance, if $B \in \bigcup_{n=1}^{N+1} \Sigma^{\otimes n}$, then $B = \bigcup_{n=1}^{N+1} B_n$ with $B_n \in \Sigma^{\otimes n}$ and we define $P_F(s, B) = \sum_{n=1}^{N+1} P_F^{\otimes n-1}(s, B^{\otimes n})$ for each $s \in \bar{\mathcal{S}}$ as the pre-measure in $\bigcup_{n=1}^{N+1} \Sigma^{\otimes n}$. Consistently with Chapter 2, we interchangeably use P_F and P_B to denote the forward and backward policies in Σ and in Σ_P, and let $\{\tau \rightsquigarrow B\}$ denote the event in which the (a.s. unique) terminal state of the trajectory τ is in $B \in \Sigma|_{\mathcal{X}}$. In this context, given a *reward measure* $R \colon \Sigma|_{\mathcal{X}} \to \mathbb{R}_+$ such that $R \ll \nu$, the learning objective of a continuous GFlowNet is to find a P_F such that the marginal distribution of $P_F(s_o, \cdot)$ over $\mathcal{X}$ matches R up to a normalizing constant, i.e., for every $B \in \Sigma|_{\mathcal{X}}$,

$$\int_{\mathcal{P}_S} P_F(s_o, \mathrm{d}\tau) \mathbb{1}_{\{\tau \rightsquigarrow B\}} = \frac{R(B)}{R(\mathcal{X})}. \tag{3.1}$$

In the search for a P_F satisfying Equation (3.1), we parameterize the density p_F of P_F relatively to κ_f and proceed in the fashion described in Chapter 2 by minimizing a surrogate learning objective enforcing a balance condition. Illustratively, the trajectory and detailed balance are defined below. There, r (resp. u) is the density of R (resp. μ) relatively to ν.

Definition 3.1.3 (TB)**.** $u(s_o) p_F(s_o, \tau) = p_B(x, \tau) r(x)$ $\kappa_f(s_o, \cdot)$-a.s. on τ.

Definition 3.1.4 (DB)**.** $u(s) p_F(s, s') = p_B(s', s) u(s')$ ν-a.s. on s and $\kappa_f(s, \cdot)$-a.s. in s'.

Lahlou et al. demonstrated that satisfying either Definition 3.1.3 or Definition 3.1.4 is a sufficient condition for the sampling correctness of GFlowNets, i.e., for Equation (3.1) to be satisfied (LAHLOU et al., 2023, Propositions 1 and 2). Both the TB and DB conditions induce a learning objective corresponding to the expected log-squared difference between their left- and right-hand sides with respect to an exploratory policy P_E, as in Chapter 2. For example, the continuous version of the TB loss is

$$\mathcal{L}_{\text{TB}}(u, p_F, p_B) = \mathop{\mathbb{E}}_{\tau \sim P_E} \left[\left(\log \frac{u(s_o) \log p_F(s_o, \tau)}{p_B(x, \tau) r(x)} \right)^2 \right], \tag{3.2}$$

in which x denotes the (a.s.) unique terminal state of τ. Naturally, the SubTB loss can be analogously exteded to the context of continuous GFlowNets. Also, as proved by Bengio et al. (BENGIO; LAHLOU, et al., 2023), p_B can be either fixed or learned jointly with p_F when learning GFlowNets. In this sense, our early experiments suggested that the training process is more stable when only p_F is learned and p_B is fixed as the density of an uniform kernel P_B (SHEN et al., 2023); this is a recurring lesson throghout this thesis.

3.2 GFlowNets and Variational Inference

Clearly, the TB condition is satisfied when $p_F(s_o, \tau) \propto p_B(x, \tau) r(x)$, i.e., when $P_F(s_o, \cdot)$ and $Q(\cdot) \propto R(\cdot) \otimes P_B(\cdot, \cdot)$ induce the same distribution on the support[1] of $\kappa_f(s_o, \cdot)$. Drawing on this, Malkin et al. (MALKIN; JAIN, et al., 2022) suggested the learning of discrete GFlowNets by minimizing a statistical divergence D between $P_F^\theta(s_o, \cdot)$ and Q, namely,

$$\hat{\theta} = \min_{\theta \in \Theta} D(P_F^\theta, Q). \tag{3.3}$$

To sample from Q, we may draw $x \sim R$ and then $\tau \sim P_B(x, \cdot)$. Hence, we will often use $P_B(s_f, \cdot)$ to refer to Q. From a variational Bayes perspective, this corresponds to the problem of finding a variational distribution P_F^θ that properly approximates the unnormalized

[1] More specifically, let $B \in \Sigma_P$ such that $P_F(s_o, B) > 0$. Then, $Q(B) \propto \int_{\tau \in B} R(\mathrm{d}x) P_B(x, \tau)$.

target $R \otimes P_B$ in the family Θ. Consequently, any VI technique can be adapted to the training of GFlowNets, as explored in detail in Section 3.3. Notably, we show below that the TB loss in Equation (3.2) corresponds to the reverse KL in Equation (3.3) when the exploratory policy is the forward policy (referred to as *on-policy* learning).

Proposition 3.2.1 (TB loss and reverse KL)**.** *Let* $\mathcal{L}_{\text{TB}}(\tau; \theta) = \left(\log \frac{u(s_o) p_F^{\theta}(s_o, \tau)}{r(x) p_B(x, \tau)}\right)^2$. *Then,*

$$\nabla_{\theta} \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\mathcal{L}_{\text{TB}}(\tau; \theta)\right] = 2 \nabla_{\theta} \mathcal{D}_{KL}[p_F || p_B]; \tag{3.4}$$

$\mathcal{D}_{KL}[p_F^{\theta} || p_B] = \underset{\tau \sim P_F(s_o, \cdot)}{\mathbb{E}} \left[\log \frac{p_F^{\theta}(s_o, \tau)}{p_B(\tau)}\right]$ *is the KL divergence between* p_F *and* $p_B(\tau) \propto r(x) p_B(x, \tau)$.

Proposition 3.2.1 extends Proposition 1 of (MALKIN; JAIN, et al., 2022) far beyond discrete distributions. This reveals a deeper connection between the conventional approach for GFlowNet training by minimizing $\mathcal{L}_{\text{TB}}$ and traditional techniques for carrying out VI.

3.3 Low-variance gradient estimators for divergences

Interestingly, minimizing statistical divergences has been shown to underperform when compared against methods that enforce a balance condition. This prompts the question of *why a consolidated approach in the statistical literature fails in the context of GFlowNet learning*? Our experiments in Section 3.4 suggest that the reason for this is the large variance of naively estimated gradients via the REINFORCE method (WILLIAMS, R. J., 1992). To see this, we start by revisiting gradient-based algorithms for minimizing the Kullback-Leibler and the family of α-divergences. Then, we introduce simply implementable and efficiently computable techniques for low-variance gradient estimation of these divergences, which we show to drastically speed up learning convergence.

Kullback-Leibler divergence. The KL divergence (KULLBACK; LEIBLER, 1951) is a widely deployed divergence measure in statistics and machine learning. To conduct variational inference, both the *forward* and *reverse* KL are considered; see Definition 3.3.1.

Definition 3.3.1 (KL divergences)**.** The *forward* KL between a target P_B and a proposal P_F is $\text{KL}(P_B || P_F) = \mathbb{E}_{\tau \sim P_B(s_f, \cdot)} \left[\log \frac{p_B(\tau)}{p_F^{\theta}(s_o, \tau)}\right]$. Also, $\text{KL}(P_F || P_B) = \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\log \frac{p_F^{\theta}(s_o, \tau)}{p_B(\tau)}\right]$ is the *reverse* KL divergence. We omit s_f from the density p_B of the target $P_B(s_f, \cdot)$.

To estimate the forward KL divergence, which depends on sampling from the intractable target distribution P_B, we implement an importance sampling scheme with a tractable proposal, e.g., an exploratory policy P_E. As only the gradients of the learning objectives are needed for training, we apply the importance weights directly to the REINFORCE-based gradient estimator of $\text{KL}(P_B || P_F)$. Lemma 3.3.1 summarizes this approach. Remarkably, the adaptive gradient techniques used for practical stochastic optimization, such as Adagrad (DUCHI; HAZAN; SINGER, 2011) and Adam (KINGMA; BA, 2014), only require the value of the gradients up to a multiplicative constant during training.

Lemma 3.3.1 (Gradients for the KL divergence)**.** *Let θ be the parameters of P_F and $s(\tau;\theta) = \log p_F^\theta(s_o, \tau)$. Then, the gradient of* $\mathrm{KL}(P_F || P_B)$ *relatively to θ satisfies*

$$\nabla_\theta \mathrm{KL}(P_F^\theta || P_B) = \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\nabla_\theta s(\tau;\theta) + \log \frac{p_F^\theta(s_o, \tau)}{p_B(x, \tau) r(x)} \nabla_\theta s(\tau;\theta) \right]$$

Correspondingly, the gradient of $\mathrm{KL}(P_B || P_F^\theta)$ *with respect to θ is*

$$\nabla_\theta \mathrm{KL}(P_B || P_F^\theta) \overset{C}{=} -\mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\frac{p_F^\theta(s_o, \tau)}{p_B(x, \tau) r(x)} \nabla_\theta s(\tau;\theta) \right],$$

in which $\overset{C}{=}$ denotes equality up to a multiplicative constant.

Crucially, the REINFORCE estimator has a high variance and we have empirically found that a naive implementation of Lemma 3.3.1 leads to a relatively slow training convergence — in accordance with (MALKIN; JAIN, et al., 2022). However, as we later demonstrate, this inefficiency can be drastically mitigated by variance-reduced gradient estimators.

Renyi-α and Tsallis-α divergences. Renyi-α (RÉNYI, 1961) and Tsallis-α (TSALLIS, 1988) are families of statistical divergences including the KL divergence as limiting cases (MINKA, 2005); see Definition 3.3.2. Albeit exhibiting a moderate success in the fields of variational inference (LI; TURNER, 2016) and model-based reinforcement learning (DEPEWEG et al., 2016), they have not been previously considered in the realm of GFlowNets. In this context, Definition 3.3.2 reviews their functional form.

Definition 3.3.2 (Renyi-α and Tsallis-α divergences)**.** Let $\alpha \in \mathbb{R}$. Also, let p_F^θ and p_B be GFlowNet's policies, respectively. Then, the *Renyi-α divergence* between P_F and P_B is

$$\mathcal{R}_\alpha(P_F || P_B) = \frac{1}{\alpha - 1} \log \int_{\mathcal{P}_\mathcal{S}} p_F^\theta(s_o, \tau)^\alpha p_B(\tau)^{1-\alpha} \kappa_f(s_o, \mathrm{d}\tau).$$

Similarly, the *Tsallis-α divergence* between P_F and P_B is

$$\mathcal{T}_\alpha(P_F || P_B) = \frac{1}{\alpha - 1} \left(\int_{\mathcal{P}_\mathcal{S}} p_{F_\theta}(s_o, \tau)^\alpha p_B(\tau)^{1-\alpha} \kappa_f(s_o, \mathrm{d}\tau) - 1 \right).$$

Definition 3.3.2 highlights that both the Renyi-α and Tsallis-α divergences transition from a mass-covering to a mode-seeking behavior as α ranges from $-\infty$ to ∞. From the perspective of GFlowNet training, the choice of α may be seen as a trade-off between diversity ($\alpha \to -\infty$) and optimality ($\alpha \to \infty$) of the generated samples in terms of the reward function. Hence, the modulation of α controls which trajectories are preferentially sampled during training, in the fashion of ϵ-greedy (MALKIN; JAIN, et al., 2022), Thompson sampling (RECTOR-BROOKS et al., 2023a), local search (KIM; YUN; BENGIO; ZHANG, et al., 2023), and forward looking (PAN; MALKIN, et al., 2023b) heuristics for selecting highly informative states during training to enhance the off-policy learning of GFlowNets.

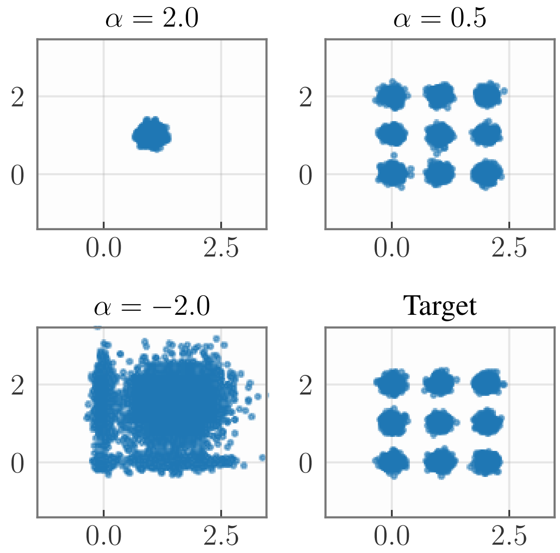

Figure 1 – Mode-seeking ($\alpha = 2$) versus mass-covering ($\alpha = -2$) in α-divergences.

To illustrate the effect of α on the learning dynamics of GFlowNets when minimizing the α-Renyi divergence, Figure 1 shows an early training stage when r represents the density of a mixture of bidimensional Gaussian distributions (bottom-right); see Section 3.4 for further details. As expected, the model exhibits a mode-seeking behavior when α is large ($\alpha = 2$, top-left) and a mass-covering behavior otherwise ($\alpha = -2$, bottom-left). The intermediate choice of $\alpha = 0.5$ provides an appropriate balance between these extremes and consistently performed well in our experiments. A similar pattern is observed when minimizing the Tsallis-α divergence (details omitted).

As discussed earlier, only the gradients of $\mathcal{R}_\alpha$ and $\mathcal{T}_\alpha$ are needed for learning. Lemma 3.3.2 formulates the REINFORCE-based estimators of both $\nabla_\theta \mathcal{R}_\alpha$ and $\nabla_\theta \mathcal{T}_\alpha$.

Lemma 3.3.2 ($\nabla_\theta \mathcal{R}_\alpha$ and $\nabla_\theta \mathcal{T}_\alpha$). *For $\tau \in \mathcal{P}_\mathcal{S}$, let $g(\tau, \theta) = \left(\frac{p_B(\tau|x)r(x)}{p_F^\theta(s_o,\tau)}\right)^{1-\alpha}$. Then,*

$$\nabla_\theta \mathcal{R}_\alpha(P_F^\theta || P_B) = \frac{\mathbb{E}[\nabla_\theta g(\tau,\theta) + g(\tau,\theta)\nabla_\theta \log p_F^\theta(s_o,\tau)]}{(\alpha-1)\mathbb{E}[g(\tau,\theta)]};$$

the expectations are computed under P_F. Analogously,

$$\nabla_\theta \mathcal{T}_\alpha(P_F^\theta || P_B) \stackrel{C}{=} \frac{\mathbb{E}[\nabla_\theta g(\tau,\theta) + g(\tau,\theta)\nabla_\theta \log p_F^\theta(s_o,\tau)]}{(\alpha-1)}.$$

We use a batch-based Monte Carlo estimate for both the numerator and denominator of the gradients in Lemma 3.3.2 (MALKIN; JAIN, et al., 2022; LAHLOU et al., 2023). Additionally, the function g is computed outside of the logarithmic domain and special care should be taken to ensure the algorithm's numerical stability. In our implementation, we sample an initial batch $\{\tau_1, \dots, \tau_K\}$ of trajectories with and compute the maximum value of the the terminal states's reward $\{r(x_1), \dots, r(x_K)\}$: $\log \bar{r} = \max_i \log r(x_i)$. Then, we re-defined the target log-density as $\log r^\star(x) = \log r(x) - \log \bar{r}$, which greatly contributed to enhance the stability of the floating-point arithmetic operations.

Control variates for low-variance gradient estimation. We first review the concept of a control variate. Let $f\colon \mathcal{P}_\mathcal{S} \to \mathbb{R}$ be a real-valued measurable function and assume that our goal is to estimate $\mathbb{E}_{\tau\sim\pi}[f(\tau)]$ according to a probability measure π on Σ_P. The variance of a naive Monte Carlo estimator based on n independent samples for this quantity is $\frac{\mathrm{Var}_\pi(f(\tau))}{n}$. On the other hand, consider a random variable (RV) $g\colon \mathcal{P}_\mathcal{S} \to \mathbb{R}$

positively correlated with f and with known expectation $\mathbb{E}_{\pi}[g(\tau)]$. Then, the variance of a naive Monte Carlo for $\mathbb{E}_{\pi}\left[f(\tau) - a(g(\tau) - \mathbb{E}_{\pi}[g(\tau)])\right]$ for a *baseline* $a \in \mathbb{R}$ is

$$\frac{1}{n}\left[\mathrm{Var}_{\pi}(f(\tau)) {-} 2a\mathrm{Cov}_{\pi}(f(\tau), g(\tau)) + a^2\mathrm{Var}_{\pi}(g(\tau))\right], \tag{3.5}$$

which is potentially smaller than $\frac{1}{n}\mathrm{Var}_{\pi}(f(\tau))$ if the covariance between f and g is sufficiently large; a is chosen to minimize Equation (3.5), i.e., $a^{\star} = \frac{\mathrm{Cov}_{\pi}(f(\tau), g(\tau))}{\mathrm{Var}_{\pi}(g(\tau))}$ (WEAVER; TAO, 2013). The function g is called a *control variate* (OWEN, 2013). As $a^{\star}$ is often unavailable in close form due to its dependence on the intractable covariance between f and g, we use a batch-based estimated of it; the incurred bias is negligible compared to the reduced variance (RANGANATH; GERRISH; BLEI, 2014; SHI et al., 2022). For vector-valued functions f and g, we provide in Proposition 3.3.1 the baseline value minimizing the trace of the covariance matrix of the corresponding Monte Carlo estimator.

Proposition 3.3.1 (Control variate for gradients)**.** *Let $f, g \colon \mathcal{P}_{\mathcal{S}} \to \mathbb{R}^d$ be vector-valued functions and π be a probability measure on $\mathcal{P}_{\mathcal{S}}$. Assume that $\mathbb{E}_{\pi}[g(\tau)] = 0$. Then,*

$$a^{\star} \coloneqq \operatorname*{arg\,min}_{a \in \mathbb{R}} \mathrm{Tr}\ \mathrm{Cov}_{\pi}[f(\tau) - a \cdot g(\tau)] {=} \frac{\mathbb{E}_{\pi}[g(\tau)^T(f(\tau) - \mathbb{E}_{\pi}[f(\tau')])]}{\mathbb{E}_{\pi}[g(\tau)^T g(\tau)]}. \tag{3.6}$$

Our gradient estimators can be written as $\mathbb{E}_{P_F(s_o, \cdot)}\left[\nabla_{\theta} f(\tau, \theta) + f(\tau, \theta)\nabla_{\theta}\log p_F^{\theta}(s_o, \tau)\right]$ for some function $f \colon \mathcal{P}_S \to \mathbb{R}$. For the first term $\nabla_{\theta} f(\tau, \theta)$, we use $g(|\tau) = \log p_F^{\theta}(\tau)$ as a control variate, which has zero expectation under $P_F(s_o, \cdot)$, and follow the approach outlined in Proposition 3.3.1. For the second term $f(\tau, \theta)\nabla_{\theta}\log p_F^{\theta}(\tau)$, we adopt a leave-one-out estimator, as described below. Importantly, Equation (3.6) cannot be straightforwardly represented as a vector-Jacobian product, which is efficient to compute in reverse-mode automatic differentation (BAYDIN et al., 2018, *autodiff*). Instead, we estimate $a^{\star}$ by linearly approximating both the numerator and denominator of its definition,

$$\hat{a} = \frac{\left\langle \sum_{n=1}^{N} \nabla_{\theta} \log p_{F_{\theta}}(\tau_n), \sum_{n=1}^{N} \nabla_{\theta} f(\tau_n) \right\rangle}{\epsilon + \left\| \sum_{n=1}^{N} \nabla_{\theta} \log p_{F_{\theta}}(\tau_n) \right\|^2}, \tag{3.7}$$

in which $\{\tau_1, \dots, \tau_n\} \sim P_F(s_o, \cdot)$ is a batch of trajectories and ϵ is a small constant that ensures numerical stability. This may also be interpreted as an instantiation of the delta method (SCHERVISH, 2012, Sec. 7.1.3). We empirically observed that this approach frequently reduces the variance of the estimated gradients by a large margin (see Section 3.4).

Leave-one-out gradient estimator. We now focus on obtaining a low-variance estimate of $\mathbb{E}_{\tau \sim P_F(s_o, \cdot)}[f(\tau, \theta)\nabla_{\theta}\log p_F^{\theta}(\tau)]$. As an alternative to Proposition 3.3.1, (SALIMANS; KNOWLES, 2014) and (SHI et al., 2022) proposed a sample-dependent baseline of the form $a(\tau_i) = \frac{1}{N-1}\sum_{n=1, n\neq i}^{N} f(\tau_n)$ for $i \in \{1, \dots, N\}$. The resulting gradient estimator,

$$\delta(\{\tau_1, \dots, \tau_N\}, f, p_F^{\theta}) = \frac{1}{N}\sum_{n=1}^{N}\left(f(\tau_n) - \frac{1}{N-1}\sum_{j=1, j\neq i}^{N} f(\tau_j)\right)\nabla_{\theta}\log p_F^{\theta}(\tau_n),$$

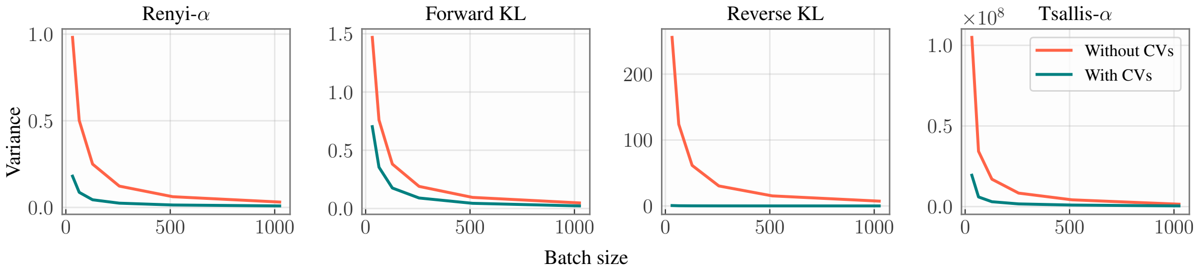

Figure 2 – **Variance of the estimated gradients as a function of the batch size.** Our CVs greatly reduce the estimator's variance even for small batch sizes.

is unbiased for $\mathbb{E}\left[f(\tau,\theta)\nabla_\theta \log p_F^\theta(\tau)\right]$ due to the independence between τ_i and τ_j for $i \neq j$. To efficiently compute δ with *autodiff*, let $\mathbf{f} = (f(\tau_n))_{n=1}^N$ and $\mathbf{p} = \left(\log p_F^\theta(\tau_n)\right)_{n=1}^N$; then

$$\delta(\mathbf{f}, \mathbf{p}) = \nabla_\theta \frac{1}{N}\left\langle \mathrm{sg}\left(\mathbf{f} - \frac{1}{N-1}(\mathbf{1} - \mathbf{I})\mathbf{f}\right), \mathbf{p}\right\rangle, \tag{3.8}$$

with sg as the stop-gradient operation (e.g., represented by `lax.stop_gradient` in JAX (BRADBURY et al., 2018) and by `torch.detach` in PyTorch (PASZKE et al., 2019)). Similarly to Proposition 3.3.1, the leave-one-out gradient estimator incurs a negligible computational overhead while significantly reducing the variance.

Relationship to past works. (MALKIN; JAIN, et al., 2022) used $\hat{a} = \frac{1}{N}\sum_{n=1}^N f(\tau_n)$ as a baseline, resulting in a biased gradient estimator. For small N, this bias may be considerable and negatively affect the optimization process. A learnable baseline independently trained to match $\hat{a}$ was also considered. We believe these design choices, along with a narrow experimental setup, entailed the inaccurate conclusion that both DB and TB losses are strictly better choices than divergence measures for training GFlowNets.

Illustraton of the control variate's effectiveness. We train the GFlowNets using increasingly larger batches of $\{2^i : i \in [[5, 10]]\}$ trajectories with and without CVs for the task of set generation (see Section 3.4 for details). Remarkably, Figure 2 shows that our CVs drastically reduce the estimator's variance — by up to three orders of magnitude. This statistical efficiency also improves training stability and convergence; see Figure 6. In contrast, our experiments with balance-based learning objectives suggested that their training dynamics is very stable and that gradient variance is not an issue; see Appendix D of Publication **III**. Also, it is mostly unclear which CVs would strongly correlate with the gradients of these objectives due to their log-squared form.

3.4 Training GFlowNets with statistical divergences

Our experiments seek to answer the following questions. First, *can the minimization of divergence-based objectives perform competitively with conventional balance-based approaches*? Second, *does the reduced variance promoted by our CVs increase convergence* *speed*? Our results positively support both questions and, more strongly, suggest that statistical divergences are often better learning objectives than their balance-based counterparts when suitable variance reduction techniques are utilized. Please refer to Publication **III** for further details regarding our experimental setup.

3.4.1 Generative tasks

We provide a high-level overview of the generative tasks used for evaluating our method. The first five tasks are based on distributions with finite support and fit in the framework of Chapter 2, while the remaining two are concerned with continuous distributions. We will often come back to this section when discussing the experimental setup in later chapters.

Set generation. (BENGIO; JAIN, et al., 2021; PAN; MALKIN, et al., 2023b; PAN; ZHANG, et al., 2023; JANG; KIM; AHN, 2024) A state s corresponds to a set of size up to a given S and the terminal states $\mathcal{X}$ are sets of size S; a transition corresponds to adding an element from a deposit $\mathcal{D}$ to s. The generative process starts at an empty set, and the log-reward of a $x \in \mathcal{X}$ is $\sum_{d \in x} f(d)$ for a fixed $f \colon \mathcal{D} \rightarrow \mathbb{R}$ defined prior to training.

Autoregressive sequence design. (JAIN et al., 2022; MALKIN; JAIN, et al., 2022) Similarly, a state is a sequence s of size at most S and a terminal state is a sequence ended by an end-of-sequence token; a transition appends $d \in \mathcal{D}$ to s. The generative process starts at an empty sequence and, for $x \in \mathcal{X}$, $\log r(x) = \sum_{i=1\dots|x|} g(i) f(x_i)$ for functions f, g.

Bayesian phylogenetic inference (BPI). (ZHOU et al., 2024) A state s is a forest composed of binary trees with labeled leaves and unlabelled internal nodes; a transition joins the roots of two trees to a newly added node. Biologically, each node corresponds to a species; the leaves represent observed species and the internal nodes characterize their long extintic ancestrals (YANG, 2014b). Then, s is terminal when it is a single connected tree — called a *phylogenetic tree*. Finally, given a dataset of nucleotide sequences, the reward function is the unnormalized posterior over trees induced by J&C69 mutation model (JUKES; CANTOR, 1969) with fixed branch lengths and an uniform prior. We use Felsenstein's algorithm (FELSENSTEIN, 1981) to efficiently compute the likelihood function.

Hypergrid navigation. (MALKIN; LAHLOU, et al., 2023; MALKIN; JAIN, et al., 2022) A state $s \in \{0, \dots, H-1\}^2$ is an element of the 2-dimensional grid with $H = 12$; a transition corresponds to adding 1 to a coordinate of s or interrupting the generative process and returning s. To assess the performance of the proposed algorithms in sampling from a highly sparse distribution, the reward function for a state s is defined as

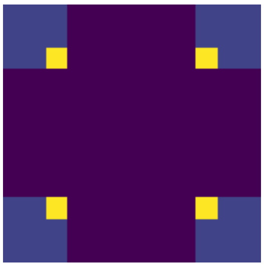

Figure 3 – Hypergrid.

$$r(s) = 10^{-3} + 0.5 \prod_{1 \leq i \leq 2} \mathbb{1}\left[\left|\frac{s_i}{H-1} - 0.5\right| \in (0.25, 0.5]\right] + 2 \prod_{1 \leq i \leq 2} \mathbb{1}\left[\left|\frac{s_i}{H-1} - 0.5\right| \in (0.3, 0.4)\right]$$

We illustrate r in Figure 3.

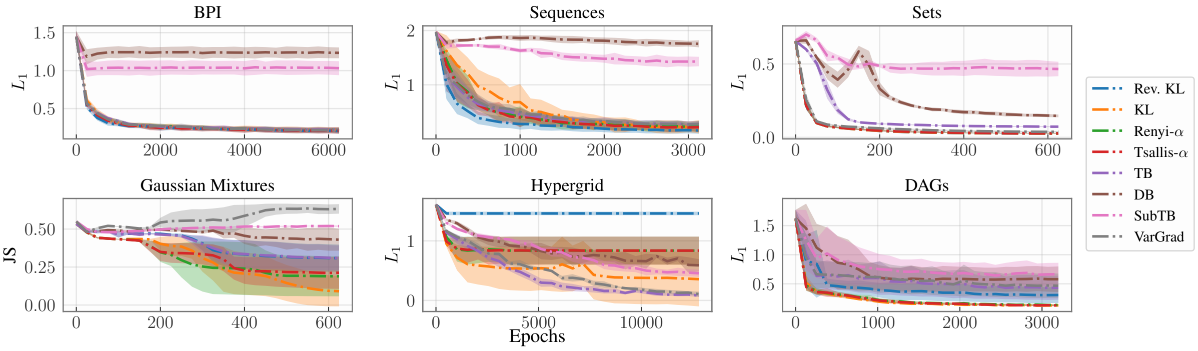

Figure 4 – **Divergence-based learning objectives often lead to faster training than TB loss.** Notably, contrasting with the experiments of (MALKIN; LAHLOU, et al., 2023), there is no single best loss function always conducting to the fastest convergence rate, and minimizing well-known divergence measures is often on par with or better than minimizing balance-based losses in terms of convergence speed. Results were averaged across three different seeds. Also, we fix $\alpha = 0.5$ for both Tsallis-α and Renyi-α divergences.

Structure learning (DAGs). (DELEU; GÓIS, et al., 2022; SILVA et al., 2024) We follow the generative process of (DELEU; GÓIS, et al., 2022, Appendix C) to generate DAGs representing a Bayesian network. Given a dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$ with n independent samples and d variables, the reward function is defined as the maximum likelihood of $\mathbf{X}$ under the structural Gaussian linear model induced by the Bayesian network. We set $n = 500$ and $d = 5$ in our experiments, similarly to (DELEU; GÓIS, et al., 2022, Figure 3).

Mixture of Gaussians (GMs) (LAHLOU et al., 2023; ZHANG; CHEN; LIU, et al., 2023)**.** The generative process starts at $\mathbf{0} \in \mathbb{R}^d$ and proceeds by sequentially substituting each coordinate with a sample from a real-valued distribution. For a K-component GM, the reward of $\mathbf{x} \in \mathbb{R}^d$ is defined as $\sum_{k=} \alpha_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$ with $\alpha_k \geq 0$ and $\sum_k \alpha_k = 1$.

Banana-shaped distribution. (RHODES; GUTMANN, 2019; MESQUITA; BLOMSTEDT; KASKI, 2019) We use the same generative process implemented for a bi-dimensional GM. For $\mathbf{z} \in \mathbb{R}^2$, we set $r(\mathbf{x})$ to a normal likelihood defined on a quadratic function of $\mathbf{x}$,

$$r(\mathbf{x}) = \mathcal{N}\left(\begin{bmatrix} x_1 \\ x_2 + x_1^2 + 1 \end{bmatrix} \middle| \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0.9 \\ 0.9 & 1 \end{bmatrix}\right). \tag{3.9}$$

We use HMC samples as ground truth to assess the performance of GFlowNet.

For the problems with finite state space, $p_F^{\theta}(\cdot|s)$ is parameterized as a softmax neural network. Otherwise, $P_F^{\theta}(s, \cdot)$ is defined as a Gaussian mixture whose mixing weights, localization, and scale are computed by an MLP (DELEU; NISHIKAWA-TOOMEY, et al., 2023).

3.4.2 Assessing convergence speed

We first show that divergence-based objectives often perform competitively with or better than conventional balance-based losses in terms of convergence speed.

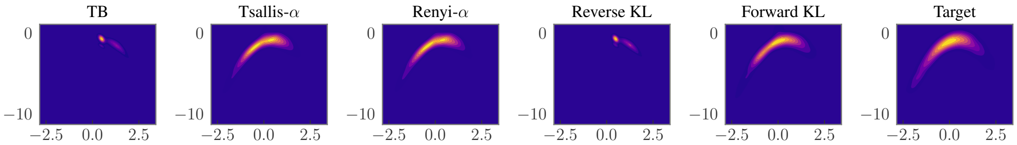

Figure 5 – **Learned distributions for the banana-shaped target.** Tsallis-α, Renyi-α, and forward KL lead to a better model than (on-policy) TB and reverse KL, which behave similarly — as predicted by Proposition 3.2.1.

Experimental setup. We compare the convergence speed in terms of the rate of decrease of a measure of distributional error when using different learning objectives. For discrete distributions, we follow (MADAN et al., 2022; BENGIO; JAIN, et al., 2021; MALKIN; JAIN, et al., 2022; PAN; ZHANG, et al., 2023) and compute the L_1 distance between the learned $p_{\top}^{\theta}(x)$ and target $r(x)$ distributions, i.e., $\sum_{x \in \mathcal{X}} |p_{\top}^{\theta}(x) - \frac{r(x)}{Z}|$ with $Z = \sum_{x \in \mathcal{X}} r(x)$ is the true r's partition function. To approximate $p_{\top}$, we use a Monte Carlo estimate of $p_{\top}(x;\theta) = \mathbb{E}_{\tau \sim P_B(x,\cdot)}\left[\frac{p_{F_\theta}(\tau|s_o;\theta)}{p_B(\tau|x)}\right]$. Otherwise, we echo (LAHLOU et al., 2023; ZHANG; CHEN; LIU, et al., 2023) and compute Jensen-Shannon's divergence between $P_{\top}^{\theta}$ and R,

$$\mathcal{D}_{JS}[P_{\top}||R] = \frac{1}{2}\left(\mathrm{KL}(P_{\top}||M) + \mathrm{KL}(R||M)\right) = \frac{1}{2}\cdot\left(\underset{x \sim P_{\top}}{\mathbb{E}}\left[\log \frac{p_{\top}(x)}{m(x)}\right] + \underset{x \sim R}{\mathbb{E}}\left[\log \frac{r(x)}{Zm(x)}\right]\right),$$

in which $M(B) = \frac{1}{2}\left(p_{\top}(B) + \frac{R(B)}{R(\mathcal{X})}\right)$ is the averaged measure of $p_{\top}$ and R and m its density relatively to the reference measure ν. For the mixture of Gaussian distributions, we can directly sample from the target to estimate $\mathrm{KL}(R||M)$. For the banana distribution, we use the HMC implementation of Stan (CARPENTER et al., 2017) to generate samples from the target and compute $\mathrm{KL}(R||M)$. In either case, we report an unbiased estimate of $\mathcal{D}_{JS}$.

Results. We compare our training algorithms against the TB, DB, SubTB, and VarGrad losses; the latter is defined as the variance of $\frac{p_F(s_o,\tau)}{p_B(x,\tau)r(x)}$ when $\tau \sim P_E(s_o,\cdot)$ (ZHANG, D. W. et al., 2023). Figure 4 highlights that the minimization of divergence-based measure with low-variance gradient estimators frequently outperforms the enforcement of balance conditions. Similarly, Table 1 and Figure 5 (for the banana-shaped distribution) reveal that divergence minimization also lead to more distributionally accurate models. A reason for this is that the optimization of both the KL- and α-divergences avoids, as opposed to TB, DB, SubTB, the estimation of the target distribution partition function, which is a hard problem (MA et al., 2013). Naturally, however, our proposed algorithms are not suitable for off-policy learning, which is a hallmark of GFlowNet training, and may not be appropriate for highly sparse target distributions as that in the hypergrid navigation problem. Similarly to the selection of MCMC's transition rules (GEYER, 1991), we believe the choice of a learning objective for GFLowNets should be done in a problem-by-problem basis.

Table 1 – Divergence minimization achieves better than or similar accuracy compared to enforcing TB.

	BPI	Sequences	Sets	GMs
TB	$0.22_{\pm 0.04}$	$0.28_{\pm 0.06}$	$0.07_{\pm 0.00}$	$0.31_{\pm 0.08}$
Rev. KL	$0.21_{\pm 0.04}$	$\mathbf{0.16}_{\pm 0.06}$	$\mathbf{0.03}_{\pm 0.00}$	$0.31_{\pm 0.09}$
For. KL	$0.22_{\pm 0.04}$	$0.23_{\pm 0.12}$	$\mathbf{0.03}_{\pm 0.00}$	$\mathbf{0.09}_{\pm 0.10}$
Renyi-α	$0.22_{\pm 0.03}$	$0.23_{\pm 0.10}$	$\mathbf{0.03}_{\pm 0.00}$	$0.19_{\pm 0.13}$
Tsallis-α	$0.21_{\pm 0.04}$	$0.22_{\pm 0.09}$	$\mathbf{0.03}_{\pm 0.00}$	$0.21_{\pm 0.11}$

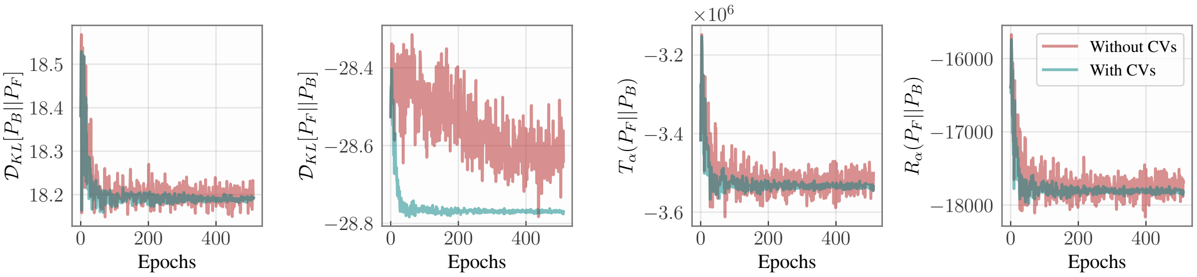

Figure 6 – **Learning curves** in the task of set generation. The reduced variance of the gradient estimates clearly increases training stability and speed. The y-axis represents the unnormalized loss, not to the exact numeric value of the divergence.

3.4.3 Reducing the variance of the estimated gradients

We have seen in Figure 2 that our CVs drastically reduce the variance of the estimated gradients. Importantly, they also significantly improve learning convergence and stability. To illustrate this, Figure 6 exhibits the learning curve of our divergence minimizing algorithms with and without CVs for the task of set generation. Remarkably, the use of CVs considerably stabilize the SGD-based optimization. This facilitates the search for a solution to the stochastic optimization problem since, intuitively, the gradient steps are less likely to steer the model away from an optimal configuration in the parameter space.

Finally, previous research has revealed similar benefits of low-variance gradient estimators in the context of simulating Langevin dynamics (HUANG; BECKER, 2021; DUBEY et al., 2016; KINOSHITA; SUZUKI, 2022) and policy gradient methods for reinforcement learning (XU; GAO; GU, 2020; PAPINI et al., 2018). Recently, Ahmadian et al. (AHMADIAN et al., 2024) demonstrated that combining the REINFORCE estimator with a leave-one-out control variate outperforms methods like DPO (RAFAILOV et al., 2024) and PPO (SCHULMAN et al., 2017) in the setting reinforcement learning with human feedback.

3.5 Chapter remarks

Our empirical results showed that the minimization of divergence measures is a sound and effective approach for training GFlowNets when adequate variance-reduction techniques are implemented to accelerate convergence (**RQI**). Also, we extended the connection between GFlowNets and VI beyond the context of finitely supported probability measures. These contributions re-open the once-dismissed research line focusing on algorithmic advances in GFlowNets inspired by principled approaches in the VI literature. For example, Burda et al. scheme for learning importance-weighted autoencoders (BURDA; GROSSE; SALAKHUTDINOV, 2016) and the mixed approach of Ruiz et al. for combining MCMC and VI (RUIZ; TITSIAS, 2019) may find fruitful applications in GFlowNet training.

3.6 Bibliographical notes

Approximate inference via variational inference (VI) methods (JORDAN et al., 1999; WAINWRIGHT; JORDAN, 2008; BISHOP, 2007; BLEI; AL., 2017) initially relied on message passing and coordinate ascent to minimize the KL divergence of an unnormalized distribution and a proposal in a parameterized tractable family of distributions. However, the development of algorithms and software for automatic differentiation (BAYDIN et al., 2018) and stochastic gradient estimators (MOHAMED et al., 2020) unlocked the potential application of generic gradient-based optimization algorithms in inference and learning tasks. Seminal works such as Black-Box VI (BBVI) (RANGANATH; GERRISH; BLEI, 2014), using the REINFORCE/score function estimator, and Automatic Differentiation VI (ADVI) (KUCUKELBIR et al., 2017), using reparameterization and change-of-variables, demonstrated practical algorithms for Bayesian inference in generic models. Overall, (MOHAMED et al., 2020) reviews the main gradient estimators: the score function (WILLIAMS, R. J., 1992; CARBONETTO; KING; HAMZE, 2009; RANGANATH; GERRISH; BLEI, 2014; YIN; ZHOU, 2018), and the pathwise gradient estimator, or the parametrization trick (REZENDE; MOHAMED, 2015; KINGMA, Durk P et al., 2014; KINGMA, Diederik P. et al., 2023). The vanilla REINFORCE/score function estimator has notoriously high variance (WILLIAMS, R. J., 1992; RICHTER et al., 2020a; CARBONETTO; KING; HAMZE, 2009; RANGANATH; GERRISH; BLEI, 2014), which prompted a body of work exploring variance reduction techniques (DOMKE, 2019, 2020; SHI et al., 2022; KIM; MA; GARDNER, 2024; WANG; GEFFNER; DOMKE, 2024).

4 Streaming Bayes GFlowNets

Recently, we have witnessed a remarkable increase in our capacity to collect and store data, which has become the trademark of the so-called *big data era*. During the COVID-19 pandemic, for example, advancements in genomic sequencing technologies led to the expansion of already extensive genetic sequence databases. This raised concerns regarding the maintenance of up-to-date models without having to repeatedly retrain them from scratch to adapt to an ever-changing environment. From a Bayesian viewpoint, this issue sparked the question of *how to update our current posterior distribution given a new batch of data*?

In this context, we refer to the problem of computing a Bayesian posterior when the data is observed in a sequence of batches as *streaming Bayesian inference* (BRODERICK et al., 2013b). Naturally, the Bayesian framework is inherently suited to a streaming setting since we can cast the current posterior as a prior for the newly observed data. As concisely formulated by Lindley (LINDLEY, 1972), “today’s posterior is tomorrow’s prior". When dealing with discrete parameter spaces, however, (approximate) Bayesian approaches either rely on intractable continuous relaxations (JANG; GU; POOLE, 2017; MADDISON; MNIH; TEH, 2017; HAN et al., 2020) or ad-hoc transition kernels for MCMC (DINH; DARLING; MATSEN IV, 2017), which are unfit for streaming Bayesian inference. In this chapter, we hence propose *Streaming Bayes GFlowNets* (SB-GFlowNets) as the first general-purpose streaming variational inference method for discrete parameter spaces.

In view of **RQII**, SB-GFlowNets also contribute to the *reusability* of GFlowNets through a *streaming update* scheme that enables adjusting the current posterior approximation to the full posterior using only the most recent model and the new data batch — and without ever needing to revisit past data. This allows for a more economical use of the available computational resources. In Chapter 5, we will present a similar approach to leverage GFlowNets for embarrassingly parallel Bayesian inference.

4.1 Streaming balance condition

Streaming Bayesian inference. We first review the problem of streaming Bayesian inference. In consistency with the terminology of Chapter 2, we assume that the data is i.i.d. drawn from a distribution $f(\cdot|x)$ indexed by a parameter $x \in \mathcal{X}$ and we define $\pi(x)$ as a prior distribution on $\mathcal{X}$. Given a sequence $(\mathcal{D}_t)_{t \geq 1}$ of datasets, we let the tth unnormalized posterior be $\tilde{\pi}_t(x) := f(\mathcal{D}_1, \ldots, \mathcal{D}_t|x)\pi(x)$. Importantly, the intrinsic coherence (BISSIRI; HOLMES; WALKER, 2016) of the Bayesian framework ensures that $\tilde{\pi}_t(x) = f(\mathcal{D}_t|x)\tilde{\pi}_{t-1}(x)$ for $t \geq 1$ with $\tilde{\pi}_o = \pi$. We also denote by $\pi_t \propto \tilde{\pi}_t$ the tth posterior. Our objective is to approximate π_t given an approximation to π_{t-1} and the new data $\mathcal{D}_t$.

Streaming Bayes GFlowNets. Our method address this issue by constructing a sequence of GFlowNets $\{G_t\}_{t \geq 1}$ such that the training of $G_t := (\mathcal{G}, p_F^{(t)}, p_B^{(t)}, F^{(t)})$ depends only on G_{t-1} and on $\mathcal{D}_t$. The first GFlowNet is standardly trained with $\tilde{\pi}_1$ as the reward function. For $t \geq 2$, we learn G_t by enforcing a *streaming balance condition*; see Definition 4.1.1. Henceforth, we will let $Z_t = F^{(t)}(s_o)$ be the G_t's estimated partition function.

Definition 4.1.1 (Streaming balance condition)**.** Let $G_t = (\mathcal{G}, p_F^{(t)}, p_B^{(t)}, Z_t)$ be a GFlowNet trained to sample proportionally to the posterior $\tilde{\pi}_t(x)$. The *streaming balance (SB) condition* for the GFlowNet $G_{t+1} = (\mathcal{G}, p_F^{(t+1)}, p_B^{(t+1)}, Z_{t+1})$, conditioned on G_t, is defined as

$$Z_{t+1} p_F^{(t+1)}(\tau) = \frac{f(\mathcal{D}_{t+1}|x) p_F^{(t)}(\tau) Z_t}{p_B^{(t)}(\tau|x)} p_B^{(t+1)}(\tau|x), \tag{4.1}$$

in which $f(\mathcal{D}_{t+1}|x)$ is the likelihood of the $(t+1)$th dataset.

Intuitively, Equation (4.1) uses G_t's estimate of $\tilde{\pi}_t$ instead of $\tilde{\pi}_t$ to train G_{t+1}. More specifically, we derive from G_t's and G_{t+1}'s TB conditions that

$$Z_t p_F^{(t)}(\tau) = \tilde{\pi}_t(x) p_B^{(t)}(\tau|x) \text{ and } Z_{t+1} p_F^{(t+1)}(\tau) = \tilde{\pi}_t(x) f(\mathcal{D}_{t+1}|x) p_B^{(t+1)}(\tau|x). \tag{4.2}$$

By re-arranging the terms to obtain $\tilde{\pi}_t(x) = {}^{Z_t p_F^{(t)}(\tau)}/_{p_B^{(t)}(\tau|x)}$ and substituting this in the latter equation, we arrive at Equation (4.1). Additionally, when $p_B^{(t)}$ is set to uniform for every t, which is the case for our experiments in Section 4.4, Equation (4.1) simplifies to

$$Z_{t+1} p_F^{(t+1)}(\tau) = Z_t f(\mathcal{D}_{t+1}|x) p_F^{(t)}(\tau). \tag{4.3}$$

Proposition 4.1.1 ensures that satisfying the SB condition up to time T is sufficient for the Tth GFlowNet to sample proportionally to the Tth posterior.

Proposition 4.1.1 (Soundness of SB-GFlowNets)**.** *Let $T \geq 2$. Assume that G_1 samples in proportion to $\tilde{\pi}_1$ and that $G_t = (p_F^{(t)}, p_B^{(t)}, Z_t)$ satisfies the SB condition in Equation (4.1) for $2 \leq t \leq T$. Then, G_T samples from $\mathcal{X}$ in proportion to $\tilde{\pi}_T(x) := f(\mathcal{D}_1, \ldots, \mathcal{D}_T|x)$.*

From a continual learning perspective, SB-GFlowNets can be interpreted as agents acting according to different rewards in a single environment and learning by focusing on one reward at a time (TIAPKIN et al., 2024a). To see this, let $\{R_t\}_{t \geq 1}$ be a sequence of reward functions $R_t \colon \mathcal{X} \to \mathbb{R}_+$ representing different tasks. (We recover the streaming Bayesian model when $R_1 = \tilde{\pi}_t$ for $t \geq 1$.) Proposition 4.1.1 implies that G_T samples $x \in \mathcal{X}$ in proportion to $\prod_{t=1}^{T} R_t(x)$ if the sequence $\{G_t\}_{t=1}^{T}$ of SB-GFlowNets satisfies the SB condition. In this case, high-scoring states according to all tasks are the most frequently sampled by G_T. This interpretation is briefly exemplified in Section 4.2, which discusses strategies for learning a GFlowNet that approximately satisfies Equation (4.1).

Algorithm 1 Training a SB-GFlowNet by minimizing $\mathcal{L}_{\text{SB}}$

```
Require: $(\mathcal{D}_t)_{T \geq t \geq 1}$ streaming data sets, $f(\cdot|x)$ a likelihood model parametrized by $x$,
    $\pi(x)$ a prior distribution over $\mathcal{X}$
Ensure: $G_T$ samples proportionally to $\left(\prod_{t=1}^{T} f(\mathcal{D}_t|x)\right)\pi(x)$
  $G_1 \leftarrow (p_F^{(1)}, p_B^{(1)}, Z_1) \coloneqq \arg\min_{p_F, p_B, Z} \mathbb{E}_{\tau \sim p_E}\left[\mathcal{L}_{\text{TB}}(\tau; p_F, p_B, Z)\right]$
  for $t$ in $\{2, \dots, T\}$ do
      $G_t \leftarrow (p_F^{(t)}, p_B^{(t)}, Z_t) \coloneqq \arg\min_{p_F, p_B, Z} \mathbb{E}_{\tau \sim p_F^{(t-1)}}\left[\mathcal{L}_{\text{SB}}(\tau; p_F, p_B, Z; G_{t-1})\right]$
  end for
```

4.2 Streaming updates of GFlowNets

Streaming balance loss. In the same fashion of GFlowNet's balance-based losses, the SB condition naturally gives rise to a learning objective by considering the expected log-squared difference between the left- and right-hand sides of Equation (4.1), namely,

$$\mathcal{L}_{\text{SB}}(G_{t+1}; G_t) = \mathop{\mathbb{E}}_{\tau \sim p_E}\left[\left(\log \frac{Z_{t+1} p_F^{(t+1)}(\tau)}{p_B^{(t+1)}(\tau|x)} \cdot \frac{p_B^{(t)}(\tau|x)}{Z_t p_F^{(t)}(\tau)} \cdot \frac{1}{f(\mathcal{D}_{t+1}|x)}\right)^2\right], \tag{4.4}$$

in which p_E is an exploratory policy. We call $\mathcal{L}_{\text{SB}}$ the *streaming balance* loss. Algorithm 1 illustrates the training of SB-GFlowNets when the first model is learned by minimizing the TB loss and the exploratory policy at the tth stage is the forward policy of the $(t-1)$th model $p_F^{(t-1)}$ for $t \geq 2$. We use SGD to approximately solve the minimization problems therein, and Section 4.3 assesses the extent to which error propagation due to imperfectly trained models affects the distributional accuracy of SB-GFlowNets.

Divergence-based streaming update. We showed in Chapter 3 that low-variance estimators of statistical divergences are effective learning objectives for GFlowNets. In this sense, we propose a KL divergence-based criterion for streaming updates of GFlowNets in Definition 4.2.1. Fundamentally, this objective trades off the advantages of off-policy learning promoted by $\mathcal{L}_{\text{SB}}$ with the exemption of estimating an intractable partition function.

Definition 4.2.1 (Divergence-based streaming update)**.** Let $G_t = (\mathcal{G}, p_F^{(t)}, p_B^{(t)}, Z_t)$ be a GFlowNet sampling proportional to $\tilde{\pi}_t$. Also, let $G_{t+1} = (\mathcal{G}, p_F^{(t+1)}, p_B^{(t+1)}, Z_{t+1})$ and define the unnormalized distribution $p(\tau) \propto p_F^{(t)}(\tau) f(\mathcal{D}_{t+1}|x)$ over trajectories. Then,

$$\mathcal{L}_{\text{KL}}(G_{t+1}; G_t) = \mathbb{E}_{\tau \sim p_F^{(t+1)}}\left[\log \frac{p_F^{(t+1)}(\tau)}{p_F^{(t)}(\tau)\pi_{t+1}(x)}\right] \overset{C}{=} \text{KL}\left(p_F^{(t+1)} || p\right), \tag{4.5}$$

is called the *KL's streaming criterion.*

To further understand the learning objective in Equation (4.5), observe that $\mathcal{L}_{\text{KL}}(G_{t+1}; G_t) = 0$ if and only if $p_F^{(t+1)}(\tau) = p(\tau) \propto p_F^{(t)}(\tau) f(\mathcal{D}_{t+1}|x)$. As a consequence,

$$p_\top^{(t+1)}(x) = \sum_{\tau \colon s_o \to x} p_F^{(t+1)}(\tau) \propto f(\mathcal{D}_{t+1}|x) \underbrace{\sum_{\tau \colon s_o \to x} p_F^{(t)}(\tau)}_{p_\top^{(t)}(x)} = p_\top^{(t)}(x) f(\mathcal{D}_{t+1}|x). \tag{4.6}$$

Hence, $p_{\top}^{(t+1)}(x) = \pi_{t+1}(x) \propto \pi_t(x) f(\mathcal{D}_{t+1}|x)$ when the tth GFlowNet correctly samples from the tth posterior π_t and the $(t+1)$th GFlowNet is a global minimizer of $\mathcal{L}_{\text{KL}}$. This shows that $\mathcal{L}_{\text{KL}}$ is a sound learning objective for streaming updates of GFlowNets. Nonetheless, as demonstrated in Chapter 3, a naive REINFORCE-based gradient estimator of $\mathcal{L}_{\text{KL}}$ has a high-variance that severely hampers the training convergence. To circumvent this issue, we rely on the REINFORCE leave-one-out (RLOO) gradient estimator (MNIH; REZENDE, 2016) discussed in Section 3.3. In this sense, define $\gamma(\tau) = \log {}^{p_F^{(t+1)}(\tau)}/_{p_F^{(t)}(\tau) f(\mathcal{D}_{t+1}|x)}$ for each trajectory τ and let $\{\tau_1, \dots, \tau_k\}$ be i.i.d. samples from $p_F^{(t+1)}$. Then, the RLOO estimator for the gradient of the KL's streaming criterion is

$$\frac{1}{k} \sum_{1 \le i \le k} \nabla_\theta \gamma(\tau_i) + \frac{1}{k} \sum_{1 \le i \le k} \left(\gamma(\tau_i) - \frac{1}{k-1} \sum_{1 \le j \le k, j \neq i} \gamma(\tau_j) \right) \nabla_\theta \log p_F^{(t+1)}(\tau), \tag{4.7}$$

in which θ represents the parameters of $p_F^{(t+1)}$. Importantly, Equation (4.7) adds a negligible computational overhead to the training process, as it can be efficiently computed in reverse-mode automatic differentation packages (BAYDIN et al., 2018).

Table 2 – TV between the single-step SB-GFlowNet and its target. TB outperforms KL for sparse targets (small α).

α	1.00	0.75	0.50
TB	0.21±0.06	0.28±0.10	**0.36**±0.24
KL	**0.13**±0.03	**0.17**±0.04	0.55±0.38

Empirical illustration. We consider the set generation task (see Section 3.4) to empirically validate SB-GFlowNets and highlight the differences between $\mathcal{L}_{\text{KL}}$ and $\mathcal{L}_{\text{SB}}$ for updating GFlowNets in a streaming fashion. Recall that this task consists of adding elements to an initially empty set until it reaches size S $(= 18)$; the state graph is illustrated in Figure 8. For a fixed utility function f, we define the family of reward functions $\left\{R_\alpha \colon \log R_\alpha(s) = \frac{1}{\alpha} \sum_{d \in s} f(d)\right\}$ parameterized by a temperature α; $f(d)$ is picked uniformly at random from $[-5, 5]$. Intuitively, R_α becomes increasingly sparse as α approaches 0, a circumstance that may favor the off-policy benefits of $\mathcal{L}_{\text{SB}}$ as opposed to $\mathcal{L}_{\text{KL}}$. Figure 7 then shows that SB-GFlowNets accurately approximate the evolving target distribution when minimizing $\mathcal{L}_{\text{SB}}$ with $\alpha = 1$. On the other hand (Table 2), while $\mathcal{L}_{\text{KL}}$ significantly accelerates the learning convergence for relatively large α (> 0.5), its benefits are hindered as we increase the target distribution's sparsity $(\alpha \to 0)$. The main reason for this is that the on-policy learning scheme implemented by $\mathcal{L}_{\text{KL}}$ potentially slows down training convergence and can suffer from mode collapse when the high-probability regions of the target are sparsely distributed (MALKIN; LAHLOU, et al., 2023). In these cases, the off-policy strategy enabled by $\mathcal{L}_{\text{SB}}$ excels at finding these regions and avoiding getting stuck in a subset of the state space.

In an analogy with Wolpert's no free-lunch theorems (WOLPERT; MACREADY, 1997), however, we emphasize that the choice of either $\mathcal{L}_{\text{SB}}$ or $\mathcal{L}_{\text{KL}}$ should be made in a problem-by-problem basis and is dependent on the prior knowledge related to the task at hand.

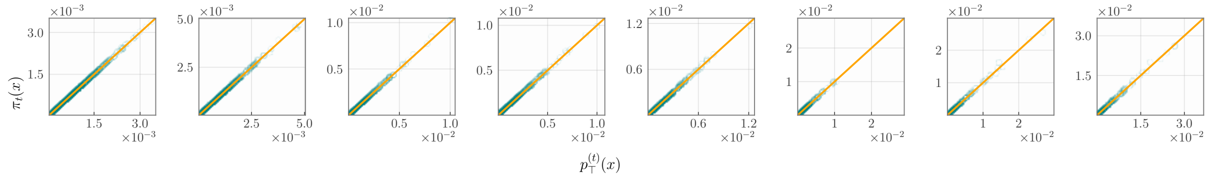

Figure 7 – **SB-GFlowNets accurately approximate a changing target** for the set generation task. Each plot shows the target and learned distributions from the first (left-most) to the last (right-most) streaming update.

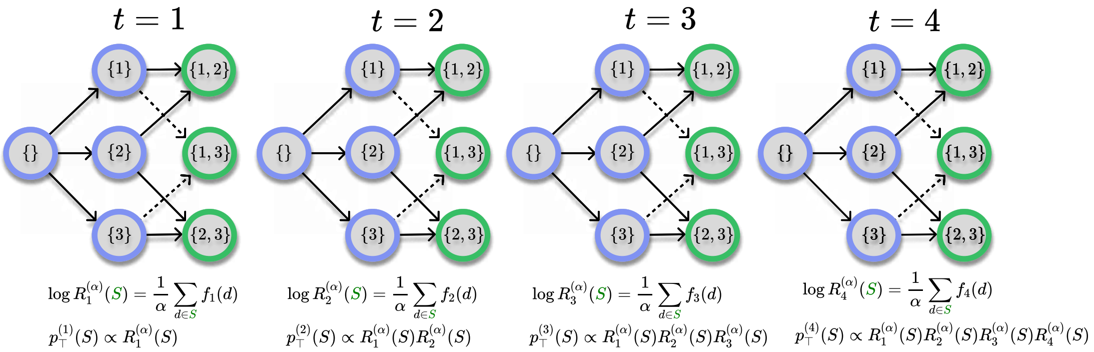
$t = 1$
$t = 2$
$t = 3$
$t = 4$
{}
{1}
{2}
{3}
{1, 2}
{1, 3}
{2, 3}
$\log R_1^{(\alpha)}(S) = \frac{1}{\alpha}\sum_{d \in S} f_1(d)$
$p_\top^{(1)}(S) \propto R_1^{(\alpha)}(S)$
$\log R_2^{(\alpha)}(S) = \frac{1}{\alpha}\sum_{d \in S} f_2(d)$
$p_\top^{(2)}(S) \propto R_1^{(\alpha)}(S)R_2^{(\alpha)}(S)$
$\log R_3^{(\alpha)}(S) = \frac{1}{\alpha}\sum_{d \in S} f_3(d)$
$p_\top^{(3)}(S) \propto R_1^{(\alpha)}(S)R_2^{(\alpha)}(S)R_3^{(\alpha)}(S)$
$\log R_4^{(\alpha)}(S) = \frac{1}{\alpha}\sum_{d \in S} f_4(d)$
$p_\top^{(4)}(S) \propto R_1^{(\alpha)}(S)R_2^{(\alpha)}(S)R_3^{(\alpha)}(S)R_4^{(\alpha)}(S)$

Figure 8 – **Illustration of the task of generating sets** of size $S = 2$ with elements in $\{1, 2, 3\}$. On each streaming update, a novel reward function $R_i^{(\alpha)}$ is observed. Terminal states $\mathcal{X}$ are illustrated in green and non-terminal states, in blue. At the tth iteration, we learn a $p_\top^{(t)}$ sampling $x \in \mathcal{X}$ in proportion to $\prod_{i=1}^{t} R_t^{(\alpha)}(x)$.

4.3 Theoretical analysis

While posterior propagation is computationally convenient, preventing training from scratch repeatedly, we should also expect errors to propagate through updates. To better understand the behavior of SB-GFlowNets, we analyze how choosing sub-optimal SB-GFlowNet at time t influences our approximation's quality at time $t+1$. We quantify goodness-of-fit of SB-GFlowNets' sampling distribution with respect to target both in terms of TV and of their expected distance in log-space. Since the SB loss and KL updates involve different parameterizations with distintic sources of errors, we analyze them separately.

Overall, we establish that the accuracy of a SB-GFlowNet's sampling distribution $p_\top^{(t+1)}$ at time $t+1$ depends on how close the new forward policy $p_F^{(t+1)}$ is from being optimal, on the size of the new data chunk $\mathcal{D}_{t+1}$, and on the goodness-of-fit of the previous estimate $p_\top^{(t)}$. As expected, our analysis suggests that the negative effects of poorly learned $p_\top^{(t)}$ are negligible when the size of a new data chunk is relatively large. Also, when G_t inadequately approximates the true posterior π_t, we discuss the benefits of using an earlier and potentially more accurate checkpoint G_s with $s < t$ as a reference for training G_{t+1}.

Although we constraint our discussion to $\mathcal{L}_{\text{SB}}$ and $\mathcal{L}_{\text{KL}}$, we also note that any surrogate objective (e.g., SubTB (MADAN et al., 2022), CB in Chapter 5, and the statistical divergences in Chapter 3) could accommodate streaming updates of GFlowNets by interpreting the $(t+1)$th policy $p_F^{(t+1)}$ as a proposal to $p_F^{(t)} f(\mathcal{D}_{t+1}|x)$ in the light of Definition 4.2.1.

4.3.1 Analysis for SB loss-based training

To start with, recall that the SB loss implies learning both a partition function Z_t and a forward policy $p_F^{(t)}$. Hence, we must account for inaccuracies in both these quantities when assessing the extent to which error propagation disrupts SB-GFlowNets. From a Markovian perspective, Proposition 4.3.1 quantifies the distributional inaccuracy of the $(t+1)$th SB-GFlowNet as a function of the estimation error at the $(t+1)$th and tth stages.

Proposition 4.3.1. *Let $G_t = (\mathcal{G}, p_F^{(t)}, p_B^{(t)}, Z_t)$ and $G_{t+1} = (\mathcal{G}, p_F^{(t+1)}, p_B^{(t+1)}, Z_{t+1})$ be subsequent GFlowNets. Let also $\pi_{t+1}(x) \propto \pi_t(x) f(\mathcal{D}_{t+1}|x)$, with $\pi_t(x) \propto \tilde{\pi}_t(x)$ and $\tilde{\pi}_t$ being an unnormalized distribution, and let $Z_k^\star := \sum_x \pi(x) \prod_{i=1}^k f(\mathcal{D}_i|x)$ for any $k > 0$. Define $(\hat{p}_F^{(t+1)}, \hat{p}_B^{(t+1)}, \hat{Z}_{t+1})$ as the optimal solution to Equation (3.7). Then,*

$$\delta_{\text{LS}}^{\pi_{t+1}}\left(p_\top^{(t+1)}, \pi_{t+1}\right) \le \underbrace{\delta_{\text{LS}}^{\pi_{t+1}}\left(p_\top^{(t+1)}, \hat{p}_\top^{(t+1)}\right) + \left|\log \frac{\hat{Z}_{t+1}}{Z_{t+1}^\star}\right|}_{\textit{Estimation error}} + \underbrace{\delta_{\text{LS}}^{\pi_{t+1}}\left(p_\top^{(t)}, \pi_t\right) + \left|\log \frac{Z_t}{Z_t^\star}\right|}_{\textit{Accuracy of } p_\top^{(t)}},$$

in which $\delta_{\text{LS}}^{\xi}(p, q) := \left(\mathbb{E}_{x \sim \xi}[\log p(x) - \log q(x)]^2\right)^{1/2}$.

Proposition 4.3.1 clarifies the importance of achieving a good approximation to both the optimal forward policy and the true partition function to ensure the resulting model is accurate. When it's reasonable to believe that properly estimating Z_t is beyond the reach of the GFlowNet, the KL streaming update might be a better surrogate objective; see Section 4.3.2. Proposition 4.3.2 complements this stance by highlighting the role of the new data batch $\mathcal{D}_{t+1}$ on the $(t+1)$th streaming update.

Proposition 4.3.2. *Let $TV(p, q) := \frac{1}{2}\sum_{x \in \mathcal{X}} |p(x) - q(x)|$ be the TV distance between probability distributions p and q. Then, with the notations of Proposition 4.3.1,*

$$TV\left(p_\top^{(t+1)}, \pi_{t+1}\right) \le TV\left(p_\top^{(t+1)}, \hat{p}_\top^{(t+1)}\right) + \frac{1}{2} \cdot f(\mathcal{D}_{t+1}|\hat{x}) \cdot \sum_{x \in \mathcal{X}} \left| \frac{Z_t}{Z_{t+1}} p_\top^{(t)}(x) - \frac{Z_t^\star}{Z_{t+1}^\star} \pi_t(x) \right|,$$

in which $\hat{x} \in \arg\max_{x \in \mathcal{X}} f(\mathcal{D}_{t+1}|x)$ is the maximum likelihood object in $\mathcal{X}$ for $\mathcal{D}_{t+1}$.

Since $f(\mathcal{D}_{t+1}|\hat{x}) \to 0$ as the size of new data batch grows and, for many models, $\frac{f(\mathcal{D}_{t+1}|\hat{x})}{f(\mathcal{D}_t|y)} \to 0$ if $y \notin \arg\max_x f(\mathcal{D}_{t+1}|x)$, the second term in Proposition 4.3.2 vanishes when the the newly observed data set $\mathcal{D}_{t+1}$ is relatively large. In this case, the distributional accuracy of $p_\top^{(t)}$ has a negligible effect on the sampling correctness of the $(t+1)$th SB-GFlowNet.

4.3.2 Analysis for KL streaming criterion-based training

Proposition 4.3.3 presents a result similar to Proposition 4.3.1 when learning with $\mathcal{L}_{\text{KL}}$.

Proposition 4.3.3. *Recall $p(\tau) \propto p_F^{(t)}(\tau) f(\mathcal{D}_{t+1}|\cdot)$. In the terminology of Proposition 4.3.1,*

$$TV\left(p_\top^{(t+1)}, \pi_{t+1}\right) \le \underbrace{\frac{1}{2}\sqrt{\mathcal{D}_{KL}\left[p_F^{(t+1)}||p\right]}}_{\textit{Estimation error}} + \underbrace{TV\left(\frac{p_\top^{(t)}(\cdot) f(\mathcal{D}_{t+1}|\cdot)}{\mathbb{E}_{y\sim p_\top^{(t)}}\left[f(\mathcal{D}_t|y)\right]}, \frac{\pi_t(\cdot) f(\mathcal{D}_{t+1}|\cdot)}{\mathbb{E}_{y\sim \pi_t}\left[f(\mathcal{D}_t|y)\right]}\right)}_{\textit{Accuracy of } p_\top^t}. \quad (4.8)$$

Proposition 4.3.3 reinforces Propositions 4.3.1 and 4.3.2 regarding the relevance of properly estimating $p_\top^{(t)}$ to achieve a good approximation to π_{t+1}. On the other hand, when $f(\mathcal{D}_{t+1}|\cdot)$ dominates the shape of $f(\mathcal{D}_{t+1}|\cdot)\pi_t(\cdot)$ and of $f(\mathcal{D}_{t+1}|\cdot)p_\top^{(t)}(\cdot)$, the accuracy of the tth model $p_\top^{(t)}$ negligibly impacts the approximation to the $(t+1)$th posterior π_{t+1}.

4.4 Case studies

We evaluate the performance of SB-GFlowNets in three realistic problems: linear preference learning with integer-valued features (COLE, 1993; HORNBERGER; HABRAKEN; BLOCH, 1995), Bayesian phylogenetic inference (YANG, 2014a; ZHOU et al., 2024), and structure learning (DELEU; NISHIKAWA-TOOMEY, et al., 2023). In each application, we measure both the distributional accuracy of the resulting SB-GFlowNet after a few streaming updates and the probability of the true parameter under the learned posterior.

4.4.1 Linear preference learning with integer-valued features

Problem description. Consider instances $\{y_i\}$ endowed with a transitive and complete preference relation $\succeq$; we assume that each $y_i \in \{1, 0\}^d$ is a binary feature vector. Naturally, the preference relation $\succeq$ is represented as a mapping u such that $y \succ y'$ if and only if $u(y) > u(y')$, and uncovering u is the major goal of preference learning methods. Similarly to (COLE, 1993; HORNBERGER; HABRAKEN; BLOCH, 1995), we here assume that u is a linear function, $u(y) = x^T y$, for a integer-valued vector x and that we have access to a data set $\mathbf{y} = \{(y_{i1}, y_{i2}, p_i)\}_i$ denoting whether y_{i1} is preferred to y_{i2} ($p_i = 1$) or otherwise ($p_i = 0$) for a fixed individual. Subsequently, we define the probabilistic model

$$p(p_i = 1|x, (y_{i1}, y_{i2})) \coloneqq \sigma(u(y_{i1}) - u(y_{i2})) = \sigma\left(x^T(y_{i1} - y_{i2})\right), \quad (4.9)$$

in which σ is the sigmoid function; the intuition is that a larger difference between y_{i1} and y_{i2}'s utilities makes the event in which y_{i1} is preferred over y_{i2} more likely. Our goal is to infer the individual's preferences based on the posterior $\pi(x|\mathbf{y})$ for some unseen pair $(\tilde{y}_1, \tilde{y}_j)$, i.e., to estimate the predictive distribution $p(\tilde{y}|(\tilde{y}_1, \tilde{y}_2), \mathbf{y})$.

Experimental setup. We assume $x \in [[0, 4]]^d$ and $d = 24$. At each streaming update, we sample a novel subset of the $5^d \cdot (5^d - 1)$ pairs of d-dimensional feature vectors and use them to update the GFlowNet. The prior on x is a factorized truncated Poisson with $\lambda = 3$.

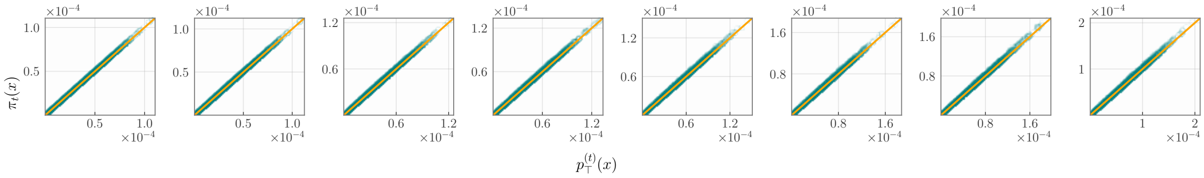

Figure 9 – **SB-GFlowNet accurately learns the posterior over the utility's parameters in a streaming setting.** Each plot compares the marginal distribution learned by SB-GFlowNet (horizontal axis) against the targeted posterior distribution (vertical axis) at increasingly advanced stages of the streaming process, i.e., from $\pi_1(\cdot|\mathcal{D}_1)$ (left-most) to $\pi_8(\cdot|\mathcal{D}_{1:8})$ (right-most).

Results. Figure 9 shows that SB-GFlowNet maintains an accurate distributional approximation throughout the streaming iterations, while the predictive negative log-likelihood (NLL) of a fixed held out data set and the mean squared error (MSE) to the true parameter decrease as a function of the amount of data consumed by the SB-GFlowNet; see Figure 10. This behavior, also exhibited by the true posterior (WALKER, 1969), further emphasizes the similarity between SB-GFlowNet's learned and targeted distributions.

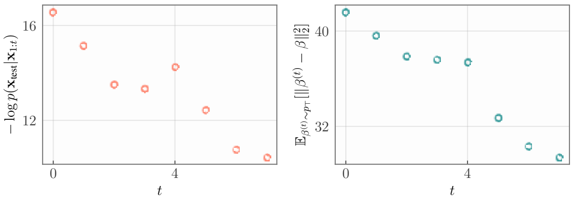

Figure 10 – Predictive performance of SB-GFlowNets in terms of predictive NLL and MSE.

4.4.2 Bayesian phylogenetic inference (BPI)

Problem description. We refer the reader to Section 3.4 for a description of the BPI task. Remarkably, the development of new sequencing technologies has led to a drastic increase in genetic sequence databases (DINH; DARLING; MATSEN IV, 2017). Thus, maintaining an up-to-date estimate of the posterior by repeatedly re-estimating the full posterior from scratch became an increasingly difficult task. In the following, we show that GFlowNets, which have recently shown SOTA performance in BPI (ZHOU et al., 2024), can also accurately update the posterior on phylogenetic trees given additional sequences.

Experimental setup. We generate the data by simulating JC69's model for a collection of $N = 7$ species and a substitution rate of $\lambda = 5 \cdot 10^{-1}$ (see (YANG, 2014a), Chapter 1). We sample a new DNA subsequence of size 10^2 at each streaming update. For Table 3, $|\mathcal{D}_1| = 10^3$ and $|\mathcal{D}_2| = 10^2$. The prior is an uniform distibution over phylogenies.

Results. Figure 11 (left) shows that the learned posterior distribution gets increasingly concentrated on the true phylogenetic tree; this behavior, which is inherent to posteriors over phylogenies under uniform priors (ROYCHOUDHURY; WILLIS; BUNGE, 2015), underlines the similarity of SB-GFlowNets to the true posterior. On the other hand, Figure 11 (right) suggets that the model's accuracy decreases as a function of the number of streaming updates. We believe this is predominantly due to the posterior's increasing spar-

Table 3 – **SB-GFlowNet significantly accelerates the training of GFlowNets** in a streaming setting while performing comparably to a GFlowNet trained from scratch on the entire dataset $\mathcal{D}_1 \cup \mathcal{D}_2$ (time measured in seconds per 20k epochs).

	Number of leaves		
Model	7	9	11
GFlowNet	2846.88 *s*	3779.11 *s*	4821.74 *s*
SB-GFlowNet (*ours*)	**1279.68** *s*	**1714.49** *s*	**2303.99** *s*
Relative accuracy gain (TV)	$0.00_{\pm 0.04}$	$-0.02_{\pm 0.04}$	$0.00_{\pm 0.01}$

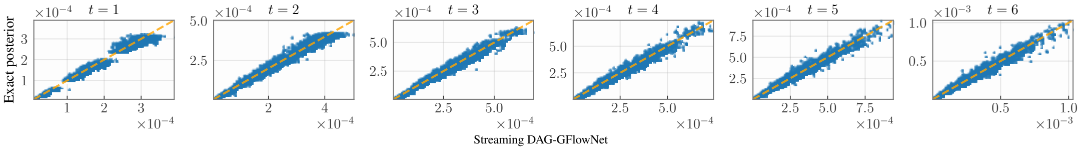

Figure 12 – **SB-GFlowNets accurately learn a distribution over DAGs** in each time step. We implemented a DAG-GFlowNet and considered the same setup of (DELEU; GÓIS, et al., 2022, Figure 3) for these experiments.

sity, which hampers GFlowNet's ability to learn a good approximation (DELEU; GÓIS, et al., 2022), instead of by an intrinsic limitation of SB-GFlowNets. Investigating this phenomenon and when to re-train the SB-GFlowNet based on an earlier checkpoint due to the accumulated unreliability of the streaming updates (Proposition 4.3.3) is application-dependent and is thereby left to future endeavors. Finally, as mentioned, SB-GFlowNets allow reusing past models to avoid expensive computations on the entire data set. In this sense, Table 3 highlights SB-GFlowNets are twice as fast as re-training a standard GFlowNet from scratch — while achieving a similar distributional approximation.

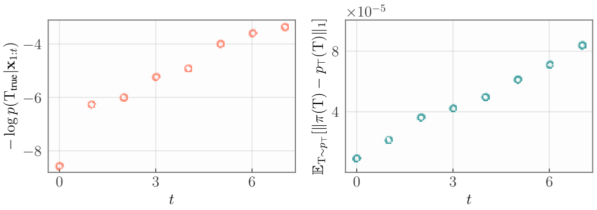

Figure 11 – SB-GFlowNet's fit to the true posterior in terms of the probability of the true phylogeny (left) and of the learned model's accuracy (right).

4.4.3 Structure learning

Problem description. Section 3.4 outlines the Bayesian structure learning problem. Likewise, we also adopt DAG-GFlowNet to learn a distribution over Bayesian networks (DELEU; GÓIS, et al., 2022). In this experiment we consider that a sequence $\{\mathcal{D}_t\}_{t \geq 1}$ of i.i.d. samples from a fixed linear Gaussian structural equation model (SEM) and define $R_t(s) = \max_\gamma \ell(\mathcal{D}_t|\gamma, s)$ as the maximum log-likelihood[1] of $\mathcal{D}_t$ with respect to the SEM's parameters γ. The resulting target distribution at the Tth streaming update is $\prod_{t=1}^T R_t(s)$.

[1] In principle, we could adopt Deleu et al.'s (DELEU; NISHIKAWA-TOOMEY, et al., 2023) scheme to learn a GFlowNet over both the structure and parameters of the underlying SEM in an fully Bayesian fashion. For illustration purposes, however, we stick to this admittedly simpler setting.

Experimental setup. A randomly parameterized Gaussian SEM over 5 variables is chosen as the data generating process. At each streaming update, we sample an additional sample of 200 points from this model. We train the SB-GFlowNet for 6 stages.

Results. Figure 12 shows SB-GFlowNets provide an accurate approximation to the belief distribution over DAGs across multiple time steps. Correspondingly, Figure 13 demonstrates that the probability of the true DAG representing the data generating process increases under SB-GFlowNet's learned distribution as more data points are incorporated into the model via streaming updates. These results, along with Section 4.4.1 and Section 4.4.2, attest the effectiveness of SB-GFlowNets in approximating a distribution in a streaming setting.

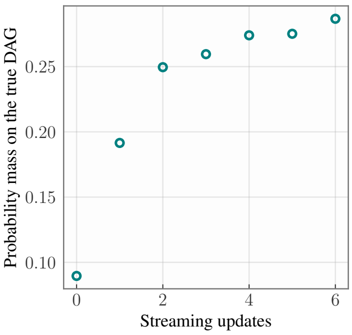

Figure 13 – True DAG probability.

4.5 Chapter remarks

We proposed *SB-GFlowNets*: the first method for streaming variational Bayesian inference over distributions on discrete parameter spaces. For this, we developed two training and update schemes for SB-GFlowNets, as well as a theoretical analysis accounting for the compounding effect of errors due to posterior propagation. Our experimental results highlight SB-GFlowNet's effectiveness in accurately sampling from the target posterior — while still achieving a significant reduction in training time with respect to a standard GFlowNet when the newly observed data is small relatively to the entire data set.

Nonetheless, Proposition 4.3.1 and Proposition 4.3.3 suggest that an inappropriate approximation to π_t may be propagated through time and lead to increasingly inaccurate models. This phenomenon, known as *catasthropic forgetting* in the online learning literature (MCCLOSKEY; COHEN, 1989; GOODFELLOW et al., 2014), may eventually demand retraining of the current SB-GFlowNet based on an earlier checkpoint with a larger batch of data. We also note that, unlike traditional variational methods for which the expressiveness of the approximation family is explicit, the expressiveness implied by different parametrizations of GFlowNets is not clear due to the intricate structure of neural networks. We believe these issues are a fruitful avenue for future investigations.

Finally, an interesting problem — that was not addressed in this thesis — consists in updating a GFlowNet when the *dimension* of the target's support changes (**RQII**). This may occur when a new species is included in our phylogenetic model or a novel variable is added to our structure learning problem (YU et al., 2010). From the flow network viewpoint, an additional dimension entails an increase in the state graph's size. However, it remains unclear how the terminal flows would be re-distributed throghout the expanded network.

4.6 Bibliographical notes

Broderick et al. proposed the Streaming, Distributed, and Asynchronous variational inference algorithm (SDA Bayes) for carrying out approximate Bayesian inference when the data is non-locally distributed in both space (e.g., in different machines) and time (BRODERICK et al., 2013a). Similarly to our work, Broderick et al.'s method was based on the principle that, if the variational distribution $q^{(t)}(x)$ is a good approximation to the unnormalized posterior $\tilde{\pi}_t(x)$, then $q^{(t+1)}(x)$ may be optimized to approximate $q^{(t)}(x)f(\mathcal{D}_t|x)$ instead of $f(\mathcal{D}_t|x)\tilde{\pi}_t(x)$. This framework was instantiated to accommodate, for example, the learning of Gaussian processes (BUI; NGUYEN; TURNER, 2017), of tensor factorization (FANG et al., 2021), of feature models (SCHAEFFER et al., 2022) and, jointly with SMC, nonlinear state-space models (ZHAO et al., 2023). In this context, SB-GFlowNets may be viewed as an instance of SDA Bayes for inferential problems on combinatorial supports.

5 Embarrassingly Parallel GFlowNets

Chapters 3 and 4 demonstrated the potential of GFlowNets in efficiently sampling from distributions over compositional spaces. In large-scale Bayesian inference, however, the repeated evaluation of an unnormalized posterior may be prohibitively expensive due to the size of the conditioning data set. In the realm of MCMC, this issue is addressed through a divide-and-conquer approach that, given an N-partition $\{\mathcal{D}_1, \dots, \mathcal{D}_N\}$ of a dataset $\mathcal{D}$ and an observational model $f(\cdot|x)$ indexed by $x \in \mathcal{X}$, approximates each *subpoterior*

$$R_n(x) \coloneqq p(\mathcal{D}_n|x)p(x)^{1/N} \tag{5.1}$$

with sample-based continuous surrogates (e.g., Gaussian processes (DE SOUZA et al., 2022) or normalizing flows (MESQUITA; BLOMSTEDT; KASKI, 2019)). The resulting models are combined in a server to sample from the *full posterior* $p(x|\mathcal{D}) \propto \prod_{n=1}^{N} R_n(x)$ (NEISWANGER; WANG; XING, 2014) — in single communication step between each client and the server. While these approaches have greatly contributed to scaling Bayesian methods to larger data sets, they are unsuited for discrete parameter spaces.

In this context, we propose *embarrassingly parallel GFlowNets* (EP-GFlowNets) as the first parallel sampling method for discrete distributions with minimal communication requirements. Similarly to their MCMC counterpart (NEISWANGER; WANG; XING, 2014), EP-GFlowNets employ a divide-and-conquer algorithm. First, a GFlowNet approximating each subposterior in (5.1) is learned in parallel. Then, these models are combined by training a centralized GFlowNet via the minimization of the newly proposed *aggregating balance* (AB) loss. Notably, the AB loss is based on the novel *contrastive balance* (CB) condition and a corresponding CB loss, which might be of independent interest and performs competitively with conventional techniques for the training of GFlowNets. In particcular, in contrast to the divergence-based learning objectives discussed in Chapter 3, the CB loss is amenable to arbitrarily complex off-policy sampling strategies.

Our developments in this chapter are in alignment with **RQI** and **RQII**. On the one hand, the CB loss allows for a minimal parameterization of the GFlowNet, requiring only the specification of a model for the forward policy, which is likely the reason for its efficiency. On the other hand, EP-GFlowNets can be seamlessly integrated into modern computer clusters to drastically reduce training time when the full posterior is expensive to evaluate.

5.1 Contrastive balance condition

Contrastive balance condition. In a distributed setting, each client GFlowNet is approximating a different target distribution. In view of our results in Chapter 3, each problem may benefit from a specific learning objective that involves a particular parameterization of the GFlowNet. For example, client 1 (resp. 2) might use the TB loss (resp. KL divergence) for training its own GFlowNet. Consequently, an ideal loss function for the aggregation of independently trained GFlowNets would not depend on the particular details of how these models were trained. As a stepstone towards achieving this objective, we develop the *contrastive balance* (CB) condition in Lemma 5.1.1.

Lemma 5.1.1 (Constrastive balance condition)**.** *Let p_F and p_B be the forward and backward policies of a GFlowNet sampling proportionally to some arbitrary reward function $R : \mathcal{X} \rightarrow \mathbb{R}^+$, then, for any pair of complete trajectories τ, τ' with $\tau \rightsquigarrow x$ and $\tau \rightsquigarrow x'$,*

$$R(x') \prod_{s \rightarrow s' \in \tau} \frac{p_F(s, s')}{p_B(s', s)} = R(x) \prod_{s \rightarrow s' \in \tau'} \frac{p_F(s, s')}{p_B(s', s)}. \tag{5.2}$$

Conversely, if a GFlowNet with forward and backward policies p_F and p_B abide by Equation (5.2), it induces a marginal distribution over $x \in \mathcal{X}$ proportional to R.

To obtain a finer understand Lemma 5.1.1, recall from the TB condition that $Zp_F(\tau) = R(x)p_B(\tau|x)$ and $Zp_F(\tau') = R(x')p_B(\tau'|x')$ for every pair of trajectories τ and τ' finishing at x and x'. By comparing these equations and rearranging the terms to isolate the partition function Z, we can easily derive Equation (5.2). Then, we get a learning objective from Equation (5.2) by taking the log-squared difference between its left- and right-hand sides.

Corollary 5.1.1 (Contrastive balance loss)**.** *Let p_F and p_B denote forward and backward policies, and let $\mathcal{T}$ be the space of complete trajectories. Also, define $\nu : \mathcal{T}^2 \rightarrow \mathbb{R}^+$ as a full-support probability distribution over pairs of trajectories in $\mathcal{T}$. Then, $p_{\top}(x) \propto R(x)$ $\forall x \in \mathcal{X}$ if and only if $\mathbb{E}_{(\tau, \tau') \sim \nu} \left[\mathcal{L}_{\text{CB}}(\tau, \tau')\right] = 0$, in which*

$$\mathcal{L}_{\text{CB}}(\tau, \tau') = \left(\log \frac{p_F(\tau)}{p_B(\tau|x)} - \log \frac{p_F(\tau')}{p_B(\tau'|x')} + \log \frac{R(x')}{R(x)} \right)^2. \tag{5.3}$$

$\mathcal{L}_{\text{CB}}$ and VI. Theorem 5.1.1 shows that the CB loss stated in Corollary 5.1.1 is equivalent to the KL divergence in terms of expected gradients. This extends the variational characterization of GFlowNets started by (MALKIN; JAIN, et al., 2022, Proposition 1) and expanded in Chapter 3. A similar result was obtained by (ZHANG; CHEN; MALKIN, et al., 2022, Proposition 14) in the context of GFlowNet-based generative modelling.

Theorem 5.1.1 (VI & CB)**.** *Let $p_F \otimes p_F$ be the outer product distribution assigning probability $p_F(\tau)p_F(\tau')$ to each trajectory pair (τ, τ'). The criterion in Equation (5.3) satisfies*

$$\nabla_\theta \text{KL}\left(p_F || p_B\right) = \frac{1}{4} \underset{(\tau, \tau') \sim p_F \otimes p_F}{\mathbb{E}} \left[\nabla_\theta \mathcal{L}_{CB}(\tau, \tau')\right].$$

Connection to other balance conditions. The CB loss is clearly connected to the TB loss (MALKIN; JAIN, et al., 2022), as discussed above. Similarly, recall that the expecta-

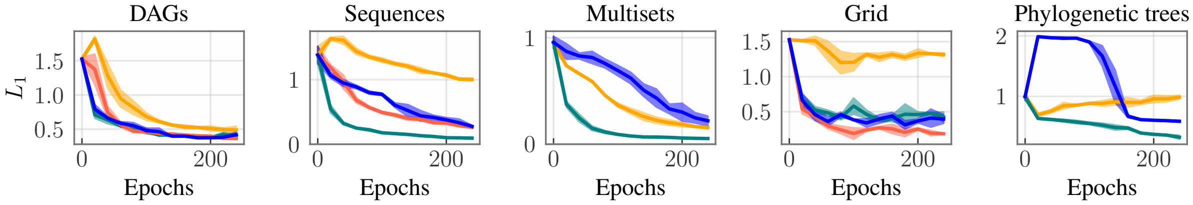

Figure 14 – $\mathcal{L}_{\text{CB}}$ performs competitively with $\mathcal{L}_{\text{TB}}$, $\mathcal{L}_{\text{DB}}$, and $\mathcal{L}_{\text{DBC}}$ in the training of conventional GFlowNets in terms of convergence speed.

tion of the squared difference of two i.i.d. random variables equals their variance. Considering this fact, $\mathcal{L}_{\text{CB}}$ may be seen as an alternative interpretation of VarGrad (RICHTER et al., 2020b; ZHANG, D. W. et al., 2023) under the hermeneutics of flow networks.

Empirical illustration. To illustrate the effectiveness of $\mathcal{L}_{\text{CB}}$ as a learning objective for training conventional GFlowNets, we assess its convergence speed in terms of reduction of the L_1 norm between the sampling and target distributions for the tasks of structure learning (DAGs), sequence design, hypergrid navigation (Grid), phylogenetic inference, and multiset generation. The latter is defined as the set generation task without the uniqueness constraint on the generated set's elements; the former were described in Section 3.4. Please refer to Publication **I** for further details regarding these experiments. In addition to the TB and DB losses, we also consider the modified DB loss proposed by (DELEU; GÓIS, et al., 2022) as an additional baseline for the tasks in which there is a terminal state associated to each non-terminal state, i.e., there is an isomorphism between $\mathcal{X}$ and $\mathcal{S} \setminus \mathcal{X}$. More specifically, if $p_F(s_f|s)$ is the probability of going from the state s to its corresponding terminal state with reward $R(s)$, Deleu et al. (DELEU; GÓIS, et al., 2022) showed that minimizing

$$\mathcal{L}_{\text{DCB}}(p_F, p_B) = \mathbb{E}_{\tau \sim p_E}\left[\frac{1}{\#\tau} \sum_{(s,s') \in \tau} \left(\log \frac{R(s) p_F(s'|s) p_F(s_f|s')}{p_F(s_f|s) p_B(s|s') R(s')}\right)^2\right] \tag{5.4}$$

is a sound approach for GFlowNet training. Likewise $\mathcal{L}_{\text{CB}}$, $\mathcal{L}_{\text{DBC}}$ also entails using a minimal parameterization of the GFlowNet. Under these circumstances, Figure 14 shows that $\mathcal{L}_{\text{CB}}$ often leads to faster convergence speed than the TB and DB losses for the tasks, while being comparable to the modified DB loss, as expected.

5.2 Aggregating balance condition

Aggregating balance condition. We now turn to the embarrassingly parallel training of GFlowNets. To fix notations, let $\{R_n\}_{n=1}^N$ be N reward functions on $\mathcal{X}$, which might be the subposteriors of a Bayesian model as in Equation (5.1). Similarly to Chapter 4, we assume there are N GFlowNets $\{G_n\}_{n=1}^N$ with each G_n samping from $\mathcal{X}$ (approximately) in proportion to R_n. Our objective is to learn a GFlownet G_o based exclusively on $\{G_n\}_{n=1}^N$ (but not on $\{R_n\}_{n=1}^N$) with sampling distribution (approximately) proportional to $\prod_{n=1}^N R_n$. Theorem 5.2.1 introduces the aggregating balance condition as a sufficient criterion on both G_o and $\{G_n\}_{n=1}^N$ for achieving this result.

Algorithm 2 Training of EP-GFlowNets

```
Require: $\left(p_F^{(1)}, p_B^{(1)}\right), \dots, \left(p_F^{(N)}, p_B^{(N)}\right)$ clients' policies, $R_1, \dots, R_N$ clients' rewards,
    $(p_F, p_B)$ parameterized global policies, $E$ epochs for training, $u_F$ uniform policy
Ensure: $p_\top(x) \propto R(x) \coloneqq \prod_{1 \leq n \leq N} R_n(x)$
  parfor $n \in \{1, \dots, N\}$ do                        ▷ Train the clients' models in parallel
      train the policies $\left(p_F^{(n)}, p_B^{(n)}\right)$ to sample proportionally to $R_n$
  end parfor
  for $e \in \{1, \dots, E\}$ do                                    ▷ Train the global model
      $\mathcal{B} \leftarrow \{(\tau, \tau')\colon \tau, \tau' \sim 1/2 \cdot p_F + 1/2 \cdot u_F\}$              ▷ Sample a batch of trajectories
      $L \leftarrow \frac{1}{|\mathcal{B}|} \sum_{\tau, \tau' \in \mathcal{B}} \mathcal{L}_{\text{AB}}\left(\tau, \tau'; \left\{\left(p_F^{(1)}, p_B^{(1)}\right), \dots, \left(p_F^{(N)}, p_B^{(N)}\right)\right\}\right)$
      Update the parameters of $p_F$ and $p_B$ through gradient descent on $L$
  end for
```

Theorem 5.2.1 (Aggregating balance (AB) condition)**.** *Let* $\left(p_F^{(1)}, p_B^{(1)}\right), \dots, \left(p_F^{(N)}, p_B^{(N)}\right)$ *be pairs of policies from* N *GFlowNets sampling respectively proportionally to* $R_1, \dots, R_N : \mathcal{X} \to \mathbb{R}_+$. *Then, another GFlowNet* G_o *with policies* p_F, p_B *samples proportionally to* $R(x) \coloneqq \prod_{n=1}^N R_n(x)$ *if and only if the it holds for all terminal trajectories* $\tau, \tau' \in \mathcal{T}$ *that*

$$\prod_{n=1}^N \frac{\left(\frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)}\right)}{\left(\frac{p_F^{(n)}(\tau')}{p_B^{(n)}(\tau'|x')}\right)} = \frac{\left(\frac{p_F(\tau)}{p_B(\tau|x)}\right)}{\left(\frac{p_F(\tau')}{p_B(\tau'|x')}\right)}. \tag{5.5}$$

We refer to $\left\{G_n \coloneqq \left(p_F^{(n)}, p_B^{(n)}\right)\right\}_{n=1}^N$ *as the* clients *and to* G_o *as the* centralized GFlowNet*.*

From an algebraic standpoint, Equation (5.5) may be directly derived from the conjunction of CB conditions. Indeed, as we demonstrated in Lemma 5.1.1, the centralized GFlowNet samples $x \in \mathcal{X}$ in proportion to $\prod_{n=1}^N R_n(x)$ when it satisfies

$$\prod_{n=1}^N \frac{R_n(x)}{R_n(x')} = \frac{p_F(\tau)}{p_B(\tau|x)} \cdot \frac{p_B(\tau'|x')}{p_F(\tau')} \tag{5.6}$$

for every pair of trajectories (τ, τ') finishing respectively at (x, x'). On the other hand, each client must also abide by its own CB: $\frac{R_n(x)}{R_n(x')} = \frac{p_F^{(n)}(\tau')p_B^{(n)}(\tau'|x')}{p_F^{(n)}(\tau)p_B^{(n)}(\tau'|x')}$. Equation (5.5) thus follows from a simple rearrangement of terms. See Appendix B for a complete proof. Importantly, the AB condition depends exclusively on the forward and backward policies of the client GFlowNets — which are required by most (possibly all) practically used GFlowNet parameterizations (MALKIN; JAIN, et al., 2022; DELEU; GÓIS, et al., 2022; PAN; ZHANG, et al., 2023; BENGIO; LAHLOU, et al., 2023; TIAPKIN et al., 2024a; KIM; YUN; BENGIO; DINGHUAI ZHANG, et al., 2024; ZHOU et al., 2024). Consequently, a practioner may freely select which learning algorithm is more suitable for each client GFlowNet and subsequently enforce Equation (5.5) by minimizing the loss function stated in Corollary 5.2.1. Algorithm 2 summarizes the training of EP-GFlowNets.

Corollary 5.2.1 (Aggregating balance loss)**.** *Let* $p_F^{(n)}$ *and* $p_B^{(n)}$ *be forward and backward transition functions such that* $p_\top^{(n)}(x) \propto R_n(x)$ *for arbitrary reward functions* R_n *over*

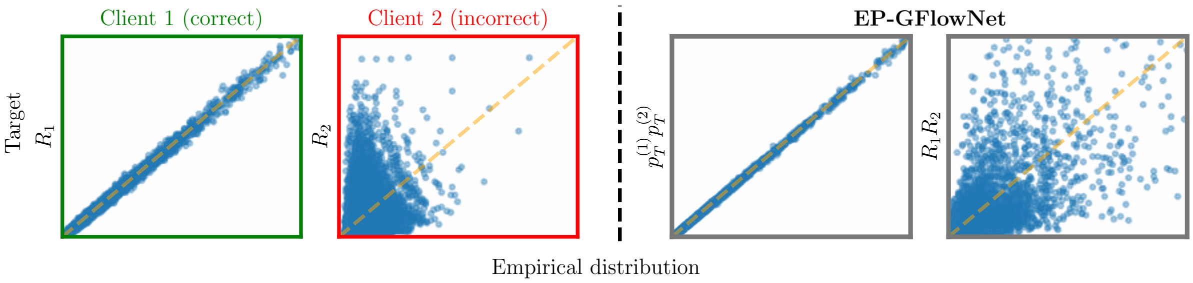

Figure 15 – **EP-GFlowNet samples proportionally to a pool of locally trained GFlowNets.** If a client correctly trains their local model (green) and another client trains theirs incorrectly (red), the distribution inferred by EP-GFlowNet (mid-right) differs from the target product distribution (right).

terminal states $x \in \mathcal{X}$. Also, let $\nu : \mathcal{T}^2 \to \mathbb{R}_+$ be a full support distribution over pairs of complete trajectories, and let p_F and p_B denote the forward and backward policies of a GFlowNet. Hence, $p_\top(x) \propto R(x)$ if and only if $\mathbb{E}_\nu \left[\mathcal{L}_{\text{AB}}\left(\tau', \tau, \{(p_F^{(n)}, p_B^{(n)})\}_{n=1}^N\right)\right] = 0$, with

$$\mathcal{L}_{\text{AB}}(\tau, \tau') = \left(\log \frac{p_F(\tau) p_B(\tau'|x')}{p_B(\tau|x) p_F(\tau')} - \sum_{1 \le i \le N} \log \frac{p_F^{(n)}(\tau) p_B^{(n)}(\tau'|x')}{p_B^{(n)}(\tau|x) p_F^{(n)}(\tau')}\right)^2. \tag{5.7}$$

Imperfect local inference. In practice, the local balance conditions often cannot be completely fulfilled by the local GFlowNets and the distributions $p_T^{(1)}, \ldots, p_T^{(N)}$ over terminal states are not proportional to the rewards $R_1, \ldots, R_N$. In this context, Theorem 5.2.2 quantifies the extent to which these local errors impact the overall result.

Theorem 5.2.2 (Influence of local failures)**.** *Let $Z_n \coloneqq \sum_{x \in \mathcal{X}} R_n(x)$, $\pi_n \coloneqq R_n/Z_n$, and $p_F^{(n)}$ and $p_B^{(n)}$ be the forward and backward policies of the n-th client. Suppose that the local balance conditions are lower- and upper-bounded for all $n \in \{1, \ldots, N\}$ as*

$$1 - \alpha_n \le \min_{x \in \mathcal{X}, \tau \rightsquigarrow x} \frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x) \pi_n(x)} \le \max_{x \in \mathcal{X}, \tau \rightsquigarrow x} \frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x) \pi_n(x)} \le 1 + \beta_n \tag{5.8}$$

where $\alpha_n \in (0, 1)$ and $\beta_n > 0$. The Jeffrey divergence J between the global model $\hat{\pi}(x)$ that fulfills the aggregating balance condition and $\pi(x) \propto \prod_{n=1}^N \pi_n(x)$ then satisfies

$$\mathrm{J}\left(\pi, \hat{\pi}\right) \le \sum_{n=1}^N \log\left(\frac{1 + \beta_n}{1 - \alpha_n}\right). \tag{5.9}$$

We make few remarks regarding the upper bound in Theorem 5.2.2. First, when the client GFlowNets are accurately learned (i.e., $\beta_n = \alpha_n = 0 \, \forall n$), the right-hand side in (5.9) vanishes. In this case, EP-GFlowNet's sampling distribution is correct: $\pi = \hat{\pi}$. Second, if either $\beta_n \to \infty$ or $\alpha_n \to 1$ for some n, the bound in Equation (5.9) goes to infinity, i.e., it degenerates if one of the local GFlowNets is inadequately trained. This is well-aligned with the *catastrophic failure* phenomenon (SOUZA et al., 2022), which was originally observed in the literature of parallel MCMC (NEISWANGER; WANG; XING, 2014; NEMETH;

SHERLOCK, 2018; MESQUITA; BLOMSTEDT; KASKI, 2019) and refers to the incorrectness of the global model due to inadequately estimated local parameters that results in, e.g., missing modes. Illustratively, Figure 15 shows a case in which one of the client GFlowNets is poorly trained (Client 2). In this case, minimizing the AB loss implies a good approximation to the product of marginal distributions over terminal states (encoded by the local GFlowNets), but the result is far from the product target distribution $R \propto R_1 R_2$.

Federated GFlowNets. In a federated setting, the data is scattered among different clients, each of whom is unwillingly to disclose their potentially sensitive information (MCMAHAN et al., 2017; NG; ZHANG, 2022). Clearly, EP-GFlowNets are versatile enough to allow for federated Bayesian inference. To avoid informational learkage, the policy networks could be trained with differential privacy guarantees (ABADI et al., 2016). Although not addressed in this thesis, we find these issues interesting for future works.

5.3 Parallel inference with GFlowNets

The main purpose of our experiments is to verify the empirical performance of EP-GFlowNets, i.e., their capacity to accurately sample from the combination of local distributions. To that extent, we consider five diverse tasks: sampling states from a *grid world*, *generation of multisets* (BENGIO; LAHLOU, et al., 2023; PAN; MALKIN, et al., 2023a), *design of sequences* (JAIN et al., 2022), *distributed Bayesian phylogenetic inference* (ZHANG; MATSEN IV, 2018; ZHOU et al., 2024), and *federated Bayesian network structure learning* (BNSL; NG; ZHANG, 2022). Since EP-GFlowNet is the first of its kind, we propose two baselines to compare it against. First, a centralized GFlowNet, which requires clients to disclose their rewards in a centralizing server. Second, a divide-and-conquer algorithm in which each client approximates its local GFlowNet with a product of categorical distributions by minimizing a KL divergence. These categorial distributions are then aggregated in the server. We call the latter approach parallel categorical VI (PCVI), which may be also viewed as an implementation of the SDA-Bayes framework for decentralized approximate Bayesian inference (BRODERICK et al., 2013b).

5.3.1 Experimental setup

The generative processes for the considered tasks are described in Section 3.4. For the grid world, the clients' individual reward functions are illustrated in Figure 16. For the multiset and sequence generation, each client has its own (randomly initialized) utility function of the available items. For the Bayesian phylogenetic inference, the clients have different genetic sequence data sets. Finally, for the BNSL problem, each client has access to i.i.d. samples from a fixed randomly parameterized linear Gaussian SEM. We only consider the PCVI baseline for the former three tasks; the latter two are not amenable to a mean-field categorical approximation to their highly structured nature.

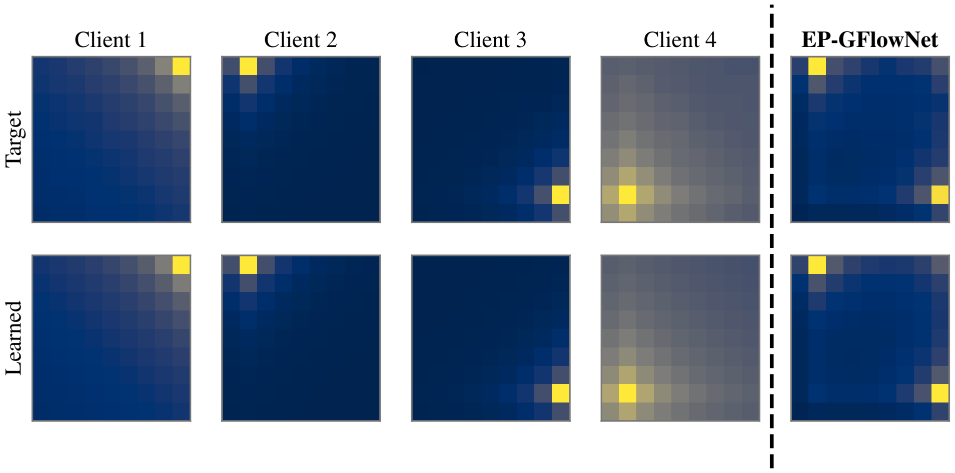

Figure 16 – **Grid world.** Each heatmap represents the target (first row), based on the normalized reward, and the local GFlowNets (second row). Results for EP-GFlowNet are in the rightmost panels. As established by Theorem 5.2.1, the good fit of the local models results in an accurate global fit.

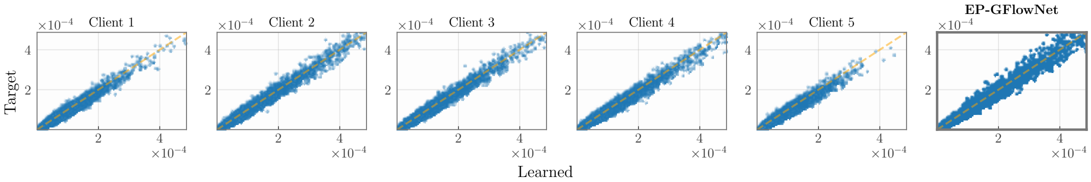

Figure 17 – **Multisets: learned × ground truth distributions.** Plots compare target vs. distributions learned by GFlowNets. The five plots to the left show local models were accurately trained. Thus, a well-trained EP-GFlowNet (right) approximates well the combined reward.

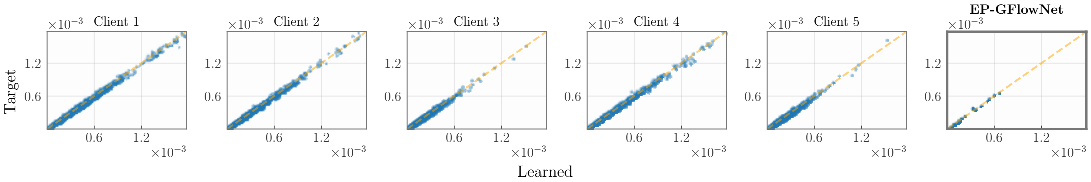

Figure 18 – **Sequences: learned × ground truth distributions.** . Plots compare target to distributions learned. The five leftmost plots show local GFlowNets were well trained. Hence, as implied by Theorem 5.2.1, EP-GFlowNet approximates well the combined reward.

5.3.2 Evaluation of EP-GFlowNets

Similarly to Chapter 4, we evaluate the accuracy of EP-GFlowNets by measuring the L_1 distance between the target and learned distributions. In this regard, Figures 16, 17, 18, and 19 show that the centralized GFlowNet's sampling distribution closely matches the target product distribution when the client GFlowNets are accurately learned. Following Deleu et al. (Figure 3; DELEU; GÓIS, et al., 2022), we compare the learned empirical averages of indicator functions for the existence of paths between node pairs against their expected value in the BNSL task. These positive results are expected due to Theorem 5.2.2.

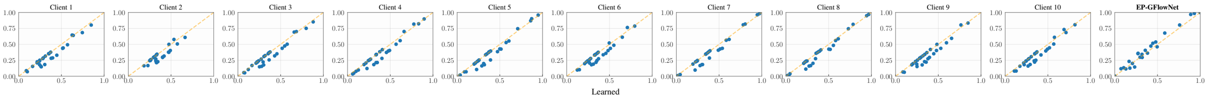

Figure 19 – **Federated Bayesian network structure learning.** Each plot shows, for each pair of nodes (U, V), the expected (vertical) and learned (horizontal) probabilities of U and V being connected by an edge, $\mathbb{P}[U \to V]$, and of existing a directed path between U and V, $\mathbb{P}[U \rightsquigarrow V]$. Notably, EP-GFlowNet accurately matches the target distribution over such edges' features.

Table 4 – **Quality of the parallel approximation** to the combined rewards. Values are the average and standard deviation over three repetitions.

	Grid World		**Multisets**		**Sequences**	
	$L_1 \downarrow$	Top-800 $\uparrow$	$L_1 \downarrow$	Top-800 $\uparrow$	$L_1 \downarrow$	Top-800 $\uparrow$
Centralized	0.027 (±0.016)	−6.355 (±0.000)	0.100 (±0.001)	27.422 (±0.000)	0.003 (±0.001)	−1.535 (±0.000)
EP-GFlowNet (**ours**)	**0.038** (±0.016)	**−6.355** (±0.000)	**0.130** (±0.004)	**27.422** (±0.000)	**0.005** (±0.002)	**−1.535** (±0.000)
PCVI	0.189 (±0.006)	**−6.355** (±0.000)	0.834 (±0.005)	26.804 (±0.018)	1.872 (±0.011)	−16.473 (±0.007)

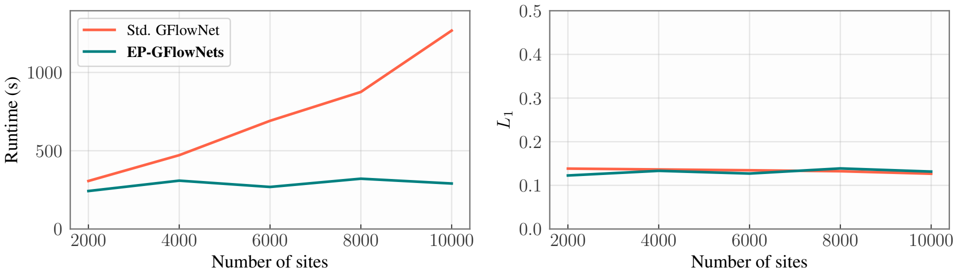

Figure 20 – **EP-GFlowNets achieve a significant reduction in runtime relatively to a standard GFlowNet** in the task of BPI when the size of the observed genomic sequences increases. Runtime for EP-GFlowNet is measured as the time for the longest client plus the time for the aggregation step.

Additionally, Table 4 shows the L_1 distance between the distribution induced by each method and the ground truth, as well as the average log reward of the top-800 scoring samples. Importantly, our EP-GFlowNet is consistently better than the PCVI baseline regarding L_1 distance, showing approximately the same performance as a centralized GFlowNet. Furthermore, EP-GFlowNet's average reward of the top-800 scoring samples, which are selected from a 10^4-sized batch drawn from the learned distribution, perfectly matches the centralized model — while PCVI's differ drastically. These results certify the effectiveness of EP-GFlowNets relatively to a naive parallel mean-field categorical approximation.

In conclusion, Figure 20 exhibits the significant gain in runtime enacted by EP-GFlowNets relatively to a standard GFlowNet when the target (reward function) is expensive to evaluate. To emulate this scenario, we simulated increasingly larger genetic sequence datasets for the BPI task. These data sets were equally partitioned among an also increasing set of clients. Importantly, the accuracy of the resulting model is roughly the same for both parallel and centralized approaches (Figure 20, right). Hence, as hypothetized in **RQII**, a distributed approach for GFlowNet appears to enhance training efficiency.

5.4 Chapter remarks

This chapter proposes the first embarrassingly parallel algorithm for GFlowNet learning, which we call EP-GFlowNet. In the same fashion of SB-GFlowNets – discussed in Chapter 4 –, EP-GFlowNets use the policies of pretrained GFlowNets as surrogates to (potentially expensive) reward functions. Remarkably, both methods are designed for non-localized training of GFlowNets with reward-level parallelization, i.e., the state graph is shared among the clients. In Chapter 7, however, we will demonstrate that an analogous reasoning enables developing a distributed algorithm with *network-level* parallelization.

From an practioner's perspective, our empirical results for the proposed CB loss suggest it leads to a significant speed up in learning convergence relatively to other balance-based objectives (**RQI**). Correspondingly, we demonstrated that EP-GFlowNets promote a significant reduction in training time when the target distribution is costly to evaluate in a single machine but can be easily parallelized across a computer cluster. Overall, we believe EP-GFlowNets will be useful for scaling up Bayesian inference with GFlowNets by amortizing the cost of expensive likelihood computations over different clients.

Finally, we notice that the *composition* of pre-trained models is an active research topic in machine learning (GARIPOV et al., 2023; DU; KAELBLING, 2024). In this sense, the modular nature of EP-GFlowNets allows for the reuse – via composition – of expensively trained GFlowNets. This might have a positive environmental impact on reducing carbon emissions emanating from the training of large-scale generative models (**RQII**).

5.5 Bibliographical notes

Hinton (HINTON, 2002) proposed a contrastive divergence-based scheme for learning a product of energy-based models, referred to as a *product of experts*. From this point of view, an EP-GFlowNet may be seen as a distributed procedure for sampling from a product of experts defined on discrete spaces (ZHANG; MALKIN, et al., 2022). In the Bayesian realm, there is a notable family of algorithms under the label of *embarrassingly parallel MCMC* (NEISWANGER; WANG; XING, 2014). For instance, (MESQUITA; BLOMSTEDT; KASKI, 2019) apply normalizing flows, (NEMETH; SHERLOCK, 2018) model the subposteriors using Gaussian processes, and (WANG, X. et al., 2015) use hyper-histograms. In contrast to EP-GFlowNets, however, these works are mostly geared towards approximating posteriors distributions over continuous random variables.

6 On the assessment and limitations of GFlowNets

Chapters 3, 4, and 5 outlined strategies for efficiently learning GFlowNets in a possibly non-localized setting. These methodological contributions, however, only slightly touched on the fundamental question of what makes the problem tackled by a GFlowNet — i.e., finding a specific flow assignment for a flow network — *hard*. In particular, can we always find a (approximate) solution to this problem within the chosen hypothesis space of policy networks? How do we know whether the learned model is close to its learning objective? In the remainder of this thesis, we consider these issues from the perspective of GNN expressivity (this chapter) and statistical learning theory (Chapter 7).

Towards this objective, we first measure the extent to which a violation to the detailed balance in a flow network might hamper the correctness of a GFlowNet's sampling distribution. In this sense, we demonstrate that the impact of an imbalanced edge on the model's accuracy depends on the amount of flow passing through it and, consequently, is unevenly distributed across the network. We also argue that, depending on the parameterization of the forward policy, imbalance is inevitable. In particular, we construct a family of graph generation problems for which a GFlowNet, due to the representation limits of its GNN-based policy networks, cannot find a flow assignment compatible with the reward function. To diagnose these limitations, we develop a tractable metric for assessing the accuracy of GFlowNets. This metric is based on comparing the learned and target distributions on small random subsets of the state space; we refer to it as *Flow-Consistency in Subgraphs* (FCS) and show it is a better proxy for correctness than popular evaluation protocols.

All in all, this is the first chapter addressing **RQIII**. The core issue we seek to explore is *when does a GFlowNet learn the correct distribution?* Although there are admittedly simple cases for which this question can be rigorously answered through a theoretical analysis, we acknowledge that an empirical evaluation is often the best we can do. In our view, however, the literature is filled up with bad advice on how to properly execute this evaluation. This is the reason we also develop FCS in Section 6.3.

6.1 Error propagation in flow networks

Our investigation starts with the main source of errors in a GFlowNet: the lack of balance in the underlying flow network. In this pursuit, the primary question we wish to address is: what is the impact of balance violations on the goodness-of-fit of GFlowNets?

6.1.1 Bounds on the total variation of GFlowNets

To build intuition, we first assess the extent to which a violation to the DB condition in a single node might affect the TV distance between the GFlowNet's sampling distribution and an uniform target for tree-structured SGs, which are often featured in applications, e.g., (JAIN et al., 2022; JIRALERSPONG et al., 2023; LIU; AL., 2023; HU et al., 2023).

Remark 6.1.1 (TV for tree-structured SGs). Let $(\mathcal{G}, p_F, p_B, F)$ be a GFlowNet balanced with respect to a reward R, in which $\mathcal{G}$ is a directed regular tree with branching factor g and depth h, and R is uniform. Also, consider $(\mathcal{G}, \tilde{p}_F, p_B, \tilde{F})$ such that i) $\tilde{F}(s_o) = F(s_o) + \delta$ and $\tilde{F}(s^\star) = F(s^\star) + \delta$ for some $s^\star \in \text{child}(s_o)$ and $\delta \geq 0$; ii) $\tilde{F}(s) = F(s)$ for all s not reachable from $s^\star$; and iii) $\tilde{F}(s) = \sum_{s' \in \text{child}(s)} \tilde{F}(s')$; see Figure 21 for when $g = 2$. Let $\tilde{p}_\top$ be the marginal of $\tilde{p}_F$ on the terminal states $\mathcal{X}$. Then, the TV between $\tilde{p}_\top$ and $\pi \propto R$ satisfies

$$\epsilon\left(\delta, g, F(s_0)\right) \leq \text{TV}\left(\tilde{p}_T, \pi\right) \leq \epsilon\left(\delta, g^h, F(s_0)\right), \text{ with } \epsilon(\delta, x, t) \coloneqq \left(1 - {}^{1}\!/\!_{x}\right) {}^{\delta}\!/\!_{t+\delta}. \tag{6.1}$$

Naturally, the upper and lower bounds are increasing functions of δ. Importantly, these bounds are tight, i.e., there is a corresponding flow function for which the TV equals the stated bounds. Also, the upper bound $\epsilon(\delta, g^h, F(s_o))$ increases monotonically with the number of leaves g^h, i.e., the further the imbalanced edge $(s_o, s^\star)$ is from the leaves, the greater the potential damage to accuracy. This demonstrates that the effect of balance violations on the distributional approximation is heterogeneously spread among the state graph's (SG) edges. We experimentally validate these findings in Figure 22.

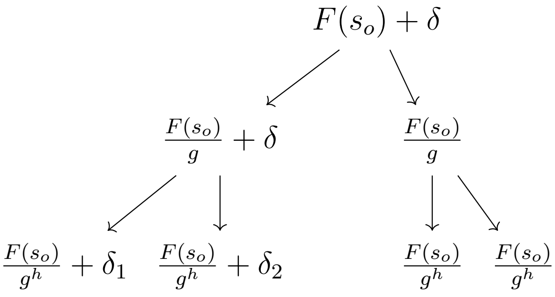
$F(s_o) + \delta$
$\frac{F(s_o)}{g} + \delta$
$\frac{F(s_o)}{g}$
$\frac{F(s_o)}{g^h} + \delta_1$
$\frac{F(s_o)}{g^h} + \delta_2$
$\frac{F(s_o)}{g^h}$
$\frac{F(s_o)}{g^h}$

Figure 21 – Tree-structured SG with excess flow δ from s_0 to left child.

As we show in Theorem 6.1.1, these intuitive results can be extended to the context of arbitrarily shaped SGs labeled with any target probability measure π. In this broader setting, of which Equation (6.1) is a particular case, the tree-inherited concepts of *depth* and *branching factor* of a state s are replaced by the *total probability mass* accumulated by the terminal descendants of s. The reader is invited to observe that, under the assumptions of Remark 6.1.1, these properties are interchangeable. From a practitioner's perspective, however, the exact computation of the quantities appearing in Theorem 6.1.1 is unfeasible for most benchmark and realistic problems. Consequently, our empirical analysis in the following section leverages the insights from Theorem 6.1.1 and the computational tractability of Remark 6.1.1 to derive a *weighted detailed balance* (WDB) loss which, by weighting different transitions based on their distance to the initial state of the SG, aims at facilitating the search for a balanced flow assignment and speeding up the training convergence.

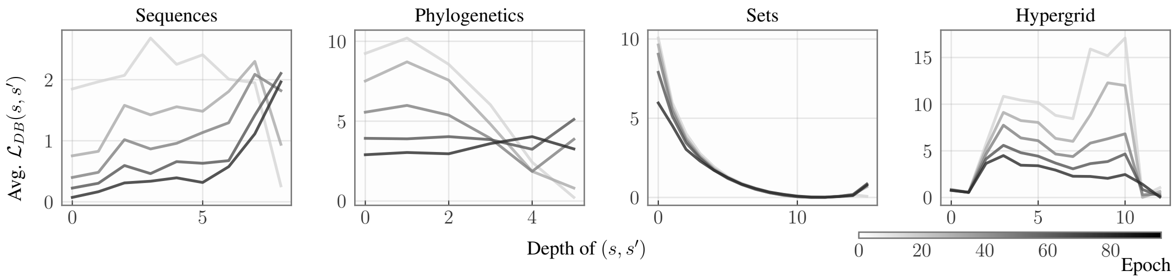

Figure 22 – **Average** $\mathcal{L}_{\text{DB}}(s, s') := (\log(F(s)p_F(s, s') - \log(F(s')p_B(s, s')))^2$ **along randomly sampled trajectories** during the early stages of training. As suggested by our analysis, the DB loss is unevenly distributed across a trajectory, with different transitions influencing the loss in diverse ways.

Theorem 6.1.1 (TV bounds for GFlowNets)**.** *Let $(\mathcal{G}, p_F, p_B, F)$ be a GFlowNet with arbitrary SG $\mathcal{G}$ satisfying the DB with respect to an arbitrary reward R. As in Remark 6.1.1, define $(\mathcal{G}, \tilde{p}_F, p_B, \tilde{F})$ by increasing the flow $F(s)$ in some node s by δ and redirecting the extra flow to a direct child $s^\star$. Likewise, $\tilde{F}$ is defined by propagating the extra flows to all states reachable from $s^\star$. Also, let $\mathcal{D}_{s^\star} \subseteq \mathcal{X}$ be the set of terminal descendants of $s^\star$. Then,*

$$\frac{\delta}{F(s_0) + \delta}\left(1 - \sum_{x \in \mathcal{D}_{s^\star}} \pi(x)\right) \leq \text{TV}\left(\tilde{p}_\top, \pi\right) \leq \frac{\delta}{F(s_0) + \delta}\left(1 - \min_{x \in \mathcal{D}_{s^\star}} \pi(x)\right). \tag{6.2}$$

6.1.2 Application to GFlowNet training

Weighted DB. We note that, by default, the DB los computes an arithmetic average of the transition-level errors (recall Chapter 2). Intrinsically, this design encodes that each transition has the same impact on our overall goal of approximating the target distribution. Nonetheless, as indicated by our theoretical analysis earlier in this section, this is not the case. Therefore, we construct a family of *weighted detailed balance* (WDB) losses,

$$\mathcal{F}_{\text{WDB}} = \left\{ \mathcal{L}_\gamma(s, s') \colon (s, s') \mapsto \gamma(s, s') \left(\log \frac{F(s)p_F(s'|s)}{F(s')p_B(s|s')} \right)^2 \middle| \gamma \colon \mathcal{S} \times \mathcal{S} \to \mathbb{R}_+ \right\}, \tag{6.3}$$

and train a GFlowNet by choosing a $\mathcal{L}_\gamma \in \mathcal{F}_{\text{WDB}}$ and minimizing the stochastic objective

$$\mathcal{L}^\gamma_{\text{WDB}}(p_F, p_B, F) \coloneqq \mathbb{E}_\tau \left[\frac{1}{\sum_{(s,s') \in \tau} \gamma(s, s')} \sum_{(s,s') \in \tau} \mathcal{L}_\gamma(s, s') \right]. \tag{6.4}$$

We are left with the task of choosing an appropriate γ. Inspired by recent advances in diffusion probabilistic models (KINGMA, Diederik P et al., 2021; KINGMA; GAO, 2023), we might choose a γ ensuring that no term in Equation (6.4) dominates the loss. In light of Remark 6.1.1, any monotonically decreasing function on $\#\mathcal{D}_{s'}$ (i.e., the number of terminal descendants of s') would be a principled choice for γ. Here, we use $\gamma(s, s') = {}^1/_{\#\mathcal{D}_{s'}}$, but acknowledge that other γ might be optimal for different tasks.

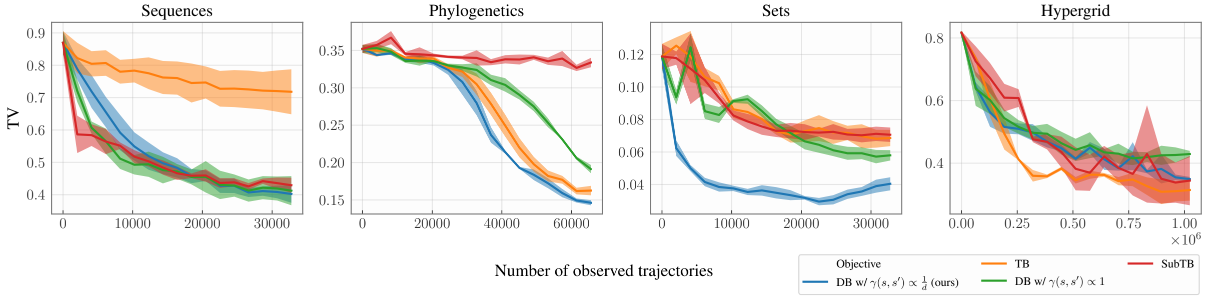

Figure 23 – **WDB performs competitively with or better than DB, SubTB, and TB**. By weighting each (s, s') in inverse proportion to the number of terminal descendants of s' (i.e., $\gamma(s, s') = {}^1\!/\#\mathcal{D}_{s'}$) in the DB loss, a faster convergence in terms of TV with respect to the standard objective is achieved.

Empirical illustration. We compare the performance of WDB against the TB, SubTB, and standard DB (with $\gamma \equiv 1$) objectives in terms of convergence speed for four benchmark of the benchmark tasks described in Section 3.4: set generation, BPI, autoregressive sequence design, and hypergrid navigation. In view of Figure 22, the initial transitions dominate the DB loss for the problems of set generation and BPI. As our weighting scheme dimishes the influence of earlier transitions in the DB loss, we would thus expect it to be more beneficial for these tasks than for the other two. Strikingly, this is exactly what Figure 22 indicates: WDB only outperforms other objectives for the set generation and BPI tasks. Broadly, our results suggest that the effectiveness of the DB loss might be drastically improved by adequately weighting the transition-wise ($\mathcal{L}_{\text{DB}}(s, s')$) terms therein.

6.2 On the limits of GNN-based GFlowNets

Our analysis so far has focused on the impact of imbalanced nodes on the GFlowNet's sampling distribution. We now step back and analyze a natural cause for this lack of balance: parametrization. Notably, some of the hottest applications of GFlowNets lie in graph domains and leverage GNNs to incorporate desirable inductive biases (e.g., BENGIO; JAIN, et al., 2021; NICA et al., 2022; ROY et al., 2023; ZHANG; DAI, et al., 2023; ZHU et al., 2023; PANDEY; SUBBARAJ; BENGIO, 2024b). In this scenario, this section explores the representational limits of GNN-based GFlowNets. Towards this goal, we show their universal capacity of approximating distributions over trees. Then, we construct a family of problems that a GFlowNet based on 1-WL GNNs, termed as *1-WL GFlowNet*, cannot solve, showing that balance violations may arise due to limited expressivity of the policy network.

Our first result (Theorem 6.2.1) demonstrates that, for any reward supported over trees, there is an 1-WL GFlowNet capable of sampling proportional to that reward. To achieve this result, we construct a simple generative process starting from a totally disconnected graph and adding one edge at a time, always yielding only one non-singleton component.

Theorem 6.2.1 (Universality of 1-WL GFlowNets for trees.)**.** *If $\mathcal{S}$ is a set of trees such that $(s, s') \in \mathcal{E}$ implies that $s \subset s'$ (s is a proper subtree of s') with $\#E(s') = \#E(s) + 1$, then there is a 1-WL GFlowNet that can approximate any distribution π over $\mathcal{X} \subseteq \mathcal{S}$.*

Theorem 6.2.1 certifies 1-WL GFlowNets can sample from arbitrary distributions over trees. However, the 1-WL test is not a perfect oracle for isomorphism. A natural question is: *are there limits to the representational power of 1-WL GFlowNets?* Theorem 6.2.2 shows a broad family of cases (i.e., combinations of SGs and rewards) for which 1-WL GFlowNets fail. This result rests on the fact that states must distribute flows evenly to children if the actions leading to them are 1-WL indistinguishable.

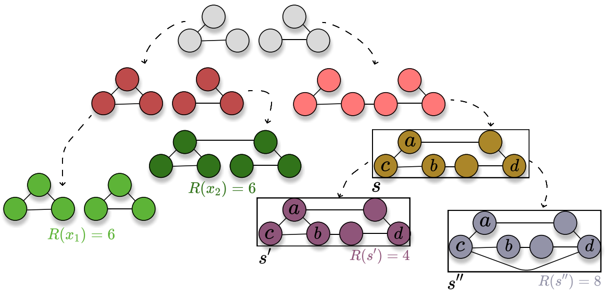

Figure 24 – A pair of state graph and reward function that causes 1-WL GFlowNets to fail.

Illustrating failure modes of 1-WL GFlowNets. To better understand Theorem 6.2.2, we first present a construction where 1-WL GFlowNets fail to achieve balance. In Figure 24, note that the actions leading to the children s' and s'' of s (enclosed by a box) are 1-WL indistinguishable. Hence, 1-WL policies distribute the flow in s equally among s' and s'', failing to match the distinct rewards $R(s') = 4$ and $R(s'') = 8$.

Theorem 6.2.2 (Limitations of GNN-based GFlowNets)**.** *Suppose the SG $\mathcal{G}$ is a directed tree with graph-structured states. Let $\mathcal{D}_s \subseteq \mathcal{X}$ for $s \in \mathcal{S}$ denote the set of terminal states reachable by a directed path starting at s. Also, assume there is a state $s = (V, E) \in \mathcal{S}$ and two pairs of nodes $(a, b) \neq (c, d) \in V^2 \setminus E$ that are not 1-WL distinguishable and $\sum_{x \in \mathcal{D}_{s'}} R(x) \neq \sum_{x \in \mathcal{D}_{s''}} R(x)$ with $s' = (V, E \cup \{(a, b)\})$ and $s'' = (V, E \cup \{(c, d)\})$ (please see Figure 24). Then there is no 1-WL GFlowNet capable of perfectly approximating $\pi \propto R$.*

We now leverage these insights to propose a more expressive GNN-based GFlowNet, which we call *Look-ahead GFlowNets* (LA-GFlowNets). The rationale of LA-GFlowNets is to incorporate children's graph embeddings as inputs to the forward policy. This allows LA-GFlowNets to disambiguate between children states obtained from 1-WL equivalent actions, enabling the assignment of uneven probabilities to non-distinguishable actions as long as the embeddings of corresponding children states differ.

More formally, let s' and s be two neighboring nodes in the SG, differing only by an edge (u, v) not in s — recall s and s' are graphs themselves. Let also $\phi_{v|G}$ be the 1-WL embedding of a node v within a graph G. Then, LA-GFlowNets' forward policy is defined as

$$p_F(s, s') \propto \exp\left\{\text{MLP}\left(\psi_1\left(\{\phi_{u|s}, \phi_{v|s}\}\right) \parallel \psi_2\left(\{\phi_{w|s'}\}_{w \in V(s')}\right)\right)\right\}, \tag{6.5}$$

in which ψ_1 and ψ_2 are order-invariant functions, e.g., DeepSets (ZAHEER et al., 2017). Since child embeddings are added (via concatenation) to the original action embedding, there is no loss of expressivity with respect to 1-WL GFlowNets. On the other hand, LA-GFlowNets can tackle cases like the one depicted in Figure 24. In this context, we formalize LA-GFlowNets' superior expressivity relatively to 1-WL GFlowNets in Theorem 6.2.3.

Theorem 6.2.3 (LA-GFlowNet $\succ$ 1-WL GFlowNet)**.** *If there is a 1-WL forward policy inducing a sampling distribution proportional to a reward R, there is a LA-GFlowNet forward policy over the same SG sampling in proportion to R. The converse does not hold.*

Empirical illustration. To demonstrate the limitations of 1-WL GFlowNets, we define next a group $\mathcal{G}$ of SGs for which there are actions that, despite leading to non-isomorphic states, cannot be distinguished by a GNN-based policy. In this scenario, let $\mathcal{R}_{n,k}$ be the set of regular graphs with n nodes of degree k. Then, consider SGs of the form $C_1 \leftarrow P \rightarrow C_2$ such that $P \in \mathcal{R}_{n,k}$ and C_1 and C_2 are non-isomorphic graphs differing from P by a single additional edge. Note that, due to the (graph-theoretic) regularity of P, $p_F(P, C_1) = p_F(P, C_2)$ for any GNN-based p_F. Thus, the corresponding GFlowNet is inherently unable to learn a non-uniform distribution on $\{C_1, C_2\}$. LA-GFlowNets, in contrast, are not constrained by such limited expressivity. As an example, we create four triples (C_1, P, C_2) with $n = 8$, $k = 3$, $R(C_1) = 0.1$ and $R(C_2) = 0.9$. In this case, Figure 25 shows LA-GFlowNet can accurately sample from the target, whereas a standard GNN-based GFlowNet cannot.

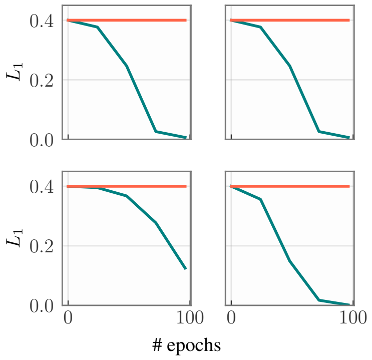

Figure 25 – Illustration of cases in which LA-GFlowNets succeed but standard GNN-based GFlowNet fail.

6.3 Flow Consistency in Subgraphs

Finally, with the understanding that there are distributions from which a GFlowNet cannot sample, we ask: how can we tractably assess the closeness of a GFlowNet's to its target? To answer this, we propose a provably correct and computationally amenable metric for probing the distributional incorrectness of GFlowNets (Section 6.3.1), termed *Flow Consistency in Subgraphs* (FCS). Strikingly, we compare FCS against three popular techniques for assessing the convergence of GFlowNets, namely, the number of modes, average reward of top-scoring samples, and Shen's accuracy, and show that FCS is often the only metric accurately reflecting a GFlowNet's goodness-of-fit (Section 6.3.2).

6.3.1 Probing GFlowNets' distributional incorrectness

Flow Consistency in Subgraphs (FCS). The basic principle of FCS is to estimate the discrepancy between ratios of probabilities (HYVÄRINEN, 2007) — instead of measuring the divergence between the intractable learned and target distributions. For this, we recall the marginal $p_\top$ of a GFlowNet $(\mathcal{G}, p_F, p_B)$ over the terminal states $\mathcal{X}$ can be computed as

$$p_\top(x) \coloneqq \sum_{\tau \colon s_o \rightsquigarrow x} p_F(\tau) = \mathbb{E}_{\tau \sim p_B(\cdot|x)}\left[\frac{p_F(\tau)}{p_B(\tau|x)}\right] \tag{6.6}$$

for each $x \in \mathcal{X}$. For most benchmark tasks, e.g., hypergrid navigation (MALKIN; LAHLOU, et al., 2023), set generation (SHEN et al., 2023), and sequence design (JAIN et al., 2022), we can exactly and efficiently compute $p_\top$. When exact computation of $p_\top$ is unfeasible, a Monte Carlo estimate of the expectation in Equation (6.6) offers an accurate approximation. Anyhow, FCS consists of comparing restrictions of $p_\top(x)$ and $R(x)$ to *random subsets* of $\mathcal{X}$. This procedure is formally described in the definition below.

Definition 6.3.1 (Flow Consistency in Sub-graphs)**.** Let P_S be a positive distribution on β-sized subsets of $\mathcal{X}$, $\beta \geq 2$. For $S \subseteq \mathcal{X}$, define the restrictions of $p_\top$ and R to S as

$$p_\top^{(S)}(x) = \frac{\mathbf{1}_{\{x \in S\}} p_\top(x)}{\sum_{y \in S} p_\top(y)} \text{ and } R^{(S)}(x) = \frac{\mathbf{1}_{\{x \in S\}} R(x)}{\sum_{y \in S} R(y)} \text{ for } x \in \mathcal{X}. \tag{6.7}$$

We define FCS as the expected TV between $p_\top^{(S)}$ and $R^{(S)}$:

$$\mathrm{FCS}(p_\top, R) \coloneqq \mathbb{E}_{S \sim P_S}[\mathrm{TV}(p_\top^{(S)}, R^{(S)})]. \tag{6.8}$$

Clearly, $\mathrm{FCS}(p_\top, R) \in [0, 1]$. Moreover, Theorem 6.3.1 shows that $\mathrm{FCS}(p_\top, R) = 0$ only if $p_\top(x) \propto R(x)$, asserting the FCS's conceptual correctness for diagnosing GFlowNets.

Theorem 6.3.1 (TV & FCS)**.** *Let P_S be any positive distribution on $\{S \subseteq \mathcal{X} \colon \#S = \beta\}$ for some $\beta \geq 2$. Also, let $\mathrm{TV}(p_\top, \pi) = \frac{1}{2}\sum_{x \in \mathcal{X}} |p_\top(x) - \pi(x)|$ be the TV distance between $p_\top$ and $\pi \coloneqq R/Z$, $Z = \sum_{x \in \mathcal{X}} R(x)$. Then, $\mathrm{TV}(p_\top, \pi) = 0$ if and only if $\mathrm{FCS}(p_\top, R) = 0$.*

Notably, β interpolates FCS between a ratio-matching-like metric (HYVÄRINEN, 2007) ($\beta = 2$) and the TV distance ($\beta = \#\mathcal{X}$). From a computational perspective, the choice of β is a trade-off between tractability (small β) and fidelity (large β). Here, we set β as the size of the batch of trajectories used in training. In this respect, Corollary 6.3.1 clarifies the role of β in FCS in terms of its proximity to the TV distance.

Corollary 6.3.1 (Role of β in FCS)**.** *Let $P_S(S; \beta) = \mathbf{1}_{\{\#S = \beta\}} \binom{n-1}{\beta-1}^{-1} \sum_{x \in S} p_\top(x)$ be a distribution over β-sized subsets of $\mathcal{X}$. Also, let $p_\top(S) = \sum_{x \in S} p_\top(x)$ and $\pi(S) = \sum_{x \in S} \pi(x)$ be the extensions of $p_\top$ and π to the discrete σ-algebra of $\mathcal{X}$. Then,*

$$\left|\mathrm{TV}\left(p_\top, \pi\right) - \mathbb{E}_{S \sim P_S(\cdot;\beta)}\left[\mathrm{TV}\left(p_\top^{(S)}, R^{(S)}\right)\right]\right| \leq \frac{1}{2} \cdot \frac{\#\mathcal{X}}{\beta} \cdot \max_{S \subseteq \mathcal{X},\, \#S = \beta} |p_\top(S) - \pi(S)|. \tag{6.9}$$

An implementation of FCS. First, we emphasize that FCS can be easily extended to accommodate variably sized subsets of $\mathcal{X}$. To see this, let P_S be any positive distribution in $\{S \subseteq \mathcal{X} \colon \#S \leq \beta\}$ and note $\mathbb{E}_{S \sim P_S}[TV(p_\top^{(S)}, R^{(S)})] = 0$ only if $\mathrm{FCS}(p_\top, R) \coloneqq$

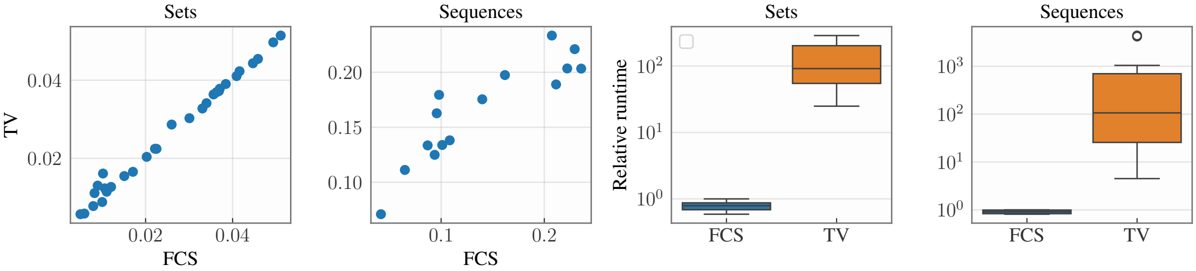

Figure 26 – **FCS is a computationally feasible surrogate for the TV distance.** (left) FCS accurately represents TV in the considered tasks (right) while being up to three orders of magnitude faster to compute.

$\mathbb{E}_{S \sim P_S(\cdot | \# S = \beta)}[TV(p_\top^{(S)}, R^{(S)})] = 0$. In this context, our implementation of FCS defines P_S as the distribution over at-most-β-sized subsets of $\mathcal{X}$ corresponding to the terminal states of a batch of trajectories sampled from a fixed policy, e.g., an uniform policy.

PAC statistical guarantees for FCS. From a statistical viewpoint, FCS approximates the distributional accuracy of a GFlowNet by probing the model on a relatively small fraction of the state graph. Under these conditions, it is natural to inquiry how this empirical estimate compares to a deterministic goodness-of-fit measure as the TV distance. Corollary 6.3.2 addresses this issue from a *probably approximately correct* (PAC) perspective, showing that an estimate of FCS closely approximates TV when a sufficiently large number of subsets is sampled (large m) and the model is relatively accurate (small error).

Corollary 6.3.2 (PAC bound for FCS). *Let P_S be as in Theorem 6.3.1 and $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$ over choosing β-sized subsets $S_1, \ldots, S_m \sim P_S$ of $\mathcal{X}$:*

$$\mathrm{TV}(p_\top, \pi) \leq \frac{1}{m} \sum_{1 \leq i \leq m} \mathrm{TV}\left(p_\top^{(S_m)}, R^{(S_m)}\right) + \frac{\#\mathcal{X}}{2\beta} \cdot \max_{S \subseteq \mathcal{X}, \#S = \beta} |p_\top(S) - \pi(S)| + \sqrt{2 \log \tfrac{1}{\delta} / m}.$$

Empirical illustration. Figure 26 (left) shows that FCS closely resembles the TV distance for the tasks of set and sequence generation; the Spearman correlation between these measures for these tasks is 0.99 and 0.90, respectively. Importantly, however, the estimation of FCS is up to three orders of magnitude faster than the computation of the TV distance (Figure 26 (right)). Remarkably, these results attest the usefulness of FCS as a general-purpose and computationally tractable goodness-of-fit metric for GFlowNets.

6.3.2 Case study: LED- and FL-GFlowNets with unrestricted flows

LED- and FL-GFlowNets. When training GFlowNets, the learning signal is *sparse*; it is only available at the end of each trajectory via the reward function. LED- (JANG; KIM; AHN, 2024) and FL- (PAN; MALKIN, et al., 2023a) GFlowNets aim at mitigating this issue by reparametrizing $\log F(s, s')$ as the residual of a potential function $\phi_\theta(s, s')$, i.e., $\log F(s, s') = \phi_\theta(s, s') + \log \tilde{F}(s, s')$, and subsequently minimizing

$$\mathcal{L}_{\mathrm{LED}}(s, s') = \left(\log \tilde{F}(s) + \log p_F(s, s') - \log p_B(s', s) - \log \tilde{F}(s') + \phi_\theta(s, s')\right)^2 \tag{6.10}$$

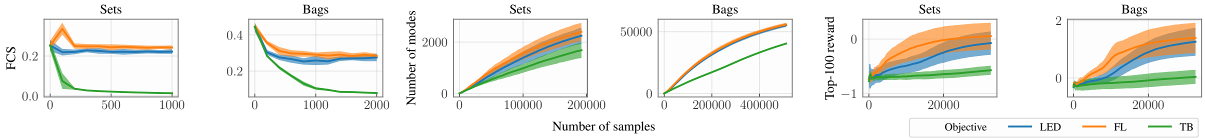

Figure 27 – **FCS is the only metric correctly reflecting GFlowNet's distributional accuracy**. On the one hand, the number of modes (columns 3-4) and the average reward (5-6) of the highest scoring samples found during training do not accurately reflect GFlowNet's goodness-of-fit. On the other hand, FCS is a sound proxy for correctness (1-2). We consider the *terminally unrestricted* variants of LED- and FL-GFlowNets (see Proposition 6.3.1).

for every (s, s'). For FL-GFlowNet, $\phi_\theta(s, s') = \xi(s') - \xi(s)$ is fixed as the gap between hand-crafted energy functions satisfying $\xi(x) = -\log R(x)$ for $x \in \mathcal{X}$ and $\xi(s_o) = 0$ (PAN; MALKIN, et al., 2023a, Equation (5)). For LED-GFlowNet, $\phi_\theta(s, s')$ is learned. Readers may consult Appendix A.2 and (JANG; KIM; AHN, 2024) for further details.

LED- and FL-GFlowNets with unrestricted flows. Our findings reveal that, even when the terminal flows $F(x)$ for $x \in \mathcal{X}$ are not constrained to equal $R(x)$, both LED- and FL-GFlowNets greatly outperform a standard GFlowNet according to conventional metrics. As we show both theoretically (Proposition 6.3.1) and empirically (Figure 27), however, constraining $F(x)$ to $R(x)$ is necessary to ensure GFlowNet's sampling correctness even under a FL-like parameterization. Importantly, we have significant reasons to believe that an unrestricted $F(x)$ was part of some experiments in the original works of (PAN; MALKIN, et al., 2023a) and (JANG; KIM; AHN, 2024), a fact that strengthens the need for a standard, easy-to-compute, and sound metric for GFlowNet assessment, such as FCS.

Proposition 6.3.1 (Unpredictability of GFlowNets with unrestricted terminal flows)**.** *Consider a FL- or LED-GFlowNet achieving $\mathcal{L}_{\text{LED}}(s, s') = 0$ for all transitions (s, s') and trajectories τ. Then, $p_\top(x) \propto R(x)\tilde{F}(x)$ for every terminal state $x \in \mathcal{X}$.*

We refer to GFlowNets that do not enforce $F(x) = R(x)$ as *terminally unrestricted* (TU).

Experimental setup. We empirically demonstrate that FCS is the *only metric* able to represent GFlowNet's accuracy when compared against three popular alternatives: number of modes, average reward of top-scoring samples, and Shen's accuracy (SHEN et al., 2023). The latter is defined by (KIM; YUN; BENGIO; DINGHUAI ZHANG, et al., 2024)

$$\texttt{ShenAccuracy}(p_\top, R) = \min\left\{\frac{\mathbb{E}_{x\sim p_\top}[R(x)]}{\mathbb{E}_{x\sim\pi}[R(x)]}, 1\right\}, \tag{6.11}$$

in which $\pi \propto R$ is a probability measure; in other words, `ShenAccuracy` measures the ratio between the GFlowNet and its target in terms of the expected reward, which is assumed to be tractably computable. For this, we consider the standard tasks of set and bag (multiset) generation, which also featured in (JANG; KIM; AHN, 2024)'s experimental campaign.

Results. Figure 27 and Table 5 teach us three facts. First, our baseline model (TB-GFlowNet) accurately learns to sample from the target distribution (Figure 27, left), whereas the (terminally unrestricted) LED- and FL-GFlowNets variants do not. Second, both LED- and FL-GFlowNets find a significantly more valuable portion of the state space — as measured by the reward function — than their standard counterpart during training (Figure 27, middle, right), but fail to sample correctly. Third, Shen's accuracy is not an appropriate proxy for goodness-of-fit. All in all, our experiments show that usual metrics should be used carefully when comparing the convergence speed of GFlowNets. In contrast to conventional wisdom in the literature, these quantities do not directly measure a GFlowNet's closeness to a global minimizer of its learning objective. Similarly, our analyses highlight the importance of a theoretically sound and computationally amenable metric for assessing GFlowNets to drive progress in the field. Strikingly, FCS is, as far as we know, the only alternative satisfying both of these constraints.

Table 5 – **(SHEN et al., 2023)'s accuracy metric** incorrectly assigns perfect score to the provably unsound TU-FL and TU-LED GFlowNet's variants.

	LED	FL	TB
Sets	**100.00**±0.00	**100.00**±0.00	93.74±0.98
Bags	**100.00**±0.00	**100.00**±0.00	81.38±6.86

6.4 Chapter remarks

We presented the first theoretical results regarding the limitations of GFlowNets in the literature. Although our analysis is far from exhaustive, we also introduced the first computationally amenable and general-purpose metric for probing the distributional accuracy of GFlowNets. On the whole, this chapter paves the road for many technical advances in this field. First, in the light of **RQI**, the development of more sophisticated weighting schemes for the DB loss might be beneficial for the training efficiency of GFlowNets. This improvement could likely be achieved by taking into account desirable properties of a learning objective in the weighting function's design, e.g., low gradient variance (recall Chapter 3). Second, we established the limits of 1-WL GNN-based GFlowNets. Albeit 1-WL GNNs are widespread in practice (KIPF; WELLING, 2016; VELIČKOVIĆ et al., 2017; XU; HU, et al., 2019; WANG; VELIČKOVIĆ, et al., 2024; CORSO et al., 2024), our analysis could be extended to more expressive GNN variants (MORRIS et al., 2021) (**RQIII**). Finally, the reason for the efficiency of the terminally unrestricted variants of FL- and LED-GFlowNets in finding high-probability regions of the target distribution remains elusive. We believe a principle understanding of this behavior would greatly benefit combinatorial optimization applications (ZHANG, D. W. et al., 2023; ZHANG; DAI, et al., 2023).

In a broader context, this work provides a cautionary tale for the adoption of invariant neural networks for parameterizing the policies of a GFlowNet and for the sloppy interpretation of surrogate metrics when assessing generative models (BLESSING et al., 2024).

6.5 Bibliographical notes

There are a few works seeking a better understanding of GFlowNets from an *experimental* point of view. Shen et al. (SHEN et al., 2023) empirically demonstrated that insufficiently trained GFlowNets tend to underestimate the expected reward under the target distribution, which motivated the use of Equation (6.11) as an evaluation metric for the correctness of GFlowNets. Similarly, Atanackovick and Bengio (ATANACKOVIC; TONG, et al., 2023) showed — through a series of experiments — that the learning of a generalizable policy networks crucially depends on the (left mostly unspecified) *structure* of the state graph. However, it is mostly unclear whether the experimental evidence therein is anecdotal. Finally, although Bengio and Lahlou et al. (BENGIO; LAHLOU, et al., 2023; LAHLOU et al., 2023) provide a thorough theoretical analysis regarding the correctness of a perfectly learned GFlowNet, they do not consider the feasibility of the GFlowNet's *learning problem*.

7 On the Generalization of GFlowNets

Chapter 6 was a first step towards a better understanding of when a GFlowNet does (not) learn the correct distribution. Nonetheless, the analysis therein was focused on how to identify and assess the limits of GFlowNets; the results were mostly on the models' negative side. From a positive standpoint, in contrast, we can only provide *probably approximately correct* (PAC) guarantees to the GFlowNet's accuracy due to the stochasticity of the training algorithm and the boundedness of computation time. In this regard, this final chapter develops PAC generalization bounds guided by McAllester's PAC-Bayesian framework (MCALLESTER, D. A., 1998) to shed light on the statistical properties of GFlowNet learning. In particular, we derive the first verifiably non-vacuous theoretical guarantees for the generalization of GFlowNets in the literature.

Epistemologically, our main conclusions are that the learning of a provably generalizable policy network becomes increasingly difficult for larger state spaces. This is intuitively plausible: the search for a compatible flow assignment for a large flow network should be harder than that for a small one. To sidestep this difficult, we propose a distributed algorithm that breaks up the flow network into subnetworks and parallely learns a different GFlowNet for each subnetwork. In a final phase, we train an additional GFlowNet to allocate the appropriate flow to each subnetwork. The resulting method, termed *Subgraph Asynchronous Learning* (SAL), relies on the principle that each GFlowNet must tackle a relatively easy task. As a consequence, SAL should find a better solution to the flow assignment problem than a monolithic GFlowNet according to usual metrics, e.g., coverage of high-probability regions and distributional accuracy. This is indeed what we observe.

In this context, our contributions enlighten the paths opened by **RQII** and **RQIII**. As a methodological advance, SAL enables the distributed training of GFlowNets with network-level parallelization in modern computer clusters. At a minimum, this achieves the objective of loosening the computational constraints of single-machine GFlowNet training. On the other hand, our theoretical analysis pioneers the development of non-vacuous generalization bounds for GFlowNets and promotes a clearer understanding of which factors hamper the learning of a provably generalizable flow assignment.

7.1 When do GFlowNets not generalize?

To start our detailed discussion on the generalization of GFlowNets, we provide a simple, but non-trivial, family of examples in which a GFlowNet may not learn a generalizable policy network even after observing an arbitrarily large portion of the state space. These negative examples impose a limit on the generality that our theoretical results can achieve.

A non-generalizable data distribution. To concretize our arguments, we recall the MDP for the set generation task (PAN; MALKIN, et al., 2023a; PAN; ZHANG, et al., 2023; BENGIO; LAHLOU, et al., 2023; JANG; KIM; AHN, 2024) described in Section 3.4.

For our purposes, we fix a function $u \colon \mathcal{D} \coloneqq \{1, \dots, W\} \rightarrow [0, 1]$ representing the *log-utility* of each $w \in \mathcal{D}$ and define the reward R associated to S as $R(s) = \mathbf{1}_{\{\#s = T\}} \exp\{\sum_{w \in s} u(w)\}$, i.e., the log-reward of a terminal set is the sum of its elements' log-utilities. Also, let $A(s)$ represent the allowed actions at state s, namely, $A(s) = \mathcal{D} \setminus s$. Then, let p_E be a forward policy such that $p_E(\cdot|s)$ is supported on $A(s) \setminus \{1\} \coloneqq \mathcal{D} \setminus (\{1\} \cup s)$ for every s; that is, the support of the marginal $p_{E,\top}$ of p_E on $\mathcal{X}$ is the set $\mathcal{X}'$ of subsets of $\{2, \dots, W\}$. We next show that $\mathcal{X}'$ may cover an arbitrarily large portion of $\mathcal{X}$ depending on T and W.

Lemma 7.1.1. *For each $\xi \in (0, 1)$, there exist T and W such that $|\mathcal{X}'| \geq \xi |\mathcal{X}|$.*

The (straightforward) proof of Lemma 7.1.1 Appendix B. Obviously, we cannot hope that a GFlowNet trained by minimizing an empirical risk defined on trajectories sampled from p_E would generalize to unseen states, as no information regarding $u(1)$ would be available during training. To empirically validate our reasoning, we show in Figure 28 that a GFlowNet trained on samples from p_E fails to learn the right distribution, whereas a standard ϵ-greedy strategy succeeds. It is remarkable, however, that a GFlowNet is unable to successfuly sample from the target distribution even after minimizing the empirical risk on samples covering over 90% of the state space. Statistically, this behavior can be understood via a change of measure inequality: preference over states is not properly captured by p_E. We formalize this intuition in the proposition below.

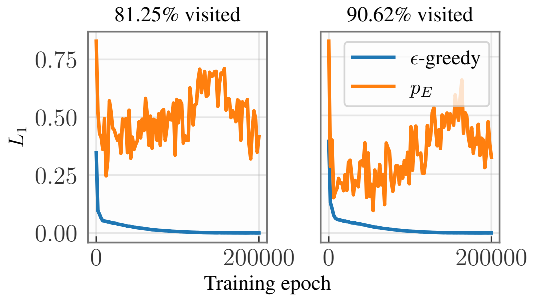

Figure 28 – Learning convergence when actions are masked (blue) or not (orange) for different state space sizes.

Proposition 7.1.1 (Generalization depends on the sampling distribution)**.** *Let (p_F, p_B) be a GFlowNet and $p_{E,\top}$ be any distribution over $\mathcal{X}$. Also, recall $\pi(x)$ is the normalized target and $p_\top$, the learned marginal. Define $q_{E,\top}$ as an uniform on $\mathcal{X}$: $q_{E,\top}(x) = |\mathcal{X}|^{-1}$. Then,*

$$\mathrm{TV}\left(p_\top, \pi\right) \lesssim \sqrt{(1 + \chi^2(q_{E,\top}||p_{E,\top}))\mathbb{E}_{x \sim p_{E,\top}}\left[\mathbb{E}_{\tau \sim p_B(\tau|x)}\left[\left(\log \frac{p_F(\tau)}{\pi(x) p_B(\tau|x)}\right)^2\right]\right]}, \quad (7.1)$$

in which $\chi^2(P||Q)$ represents the χ^2 divergence between P and Q.

We interpret Equation (7.1) in the following way: If the sampling policy $(p_{E,\top})$ greatly deviates from the uniform $(q_{E,\top})$, then a small empirical risk does not necessarily ensure an accurate distributional approximation. In contrast, Equation (7.1) does *not* entail that the uniform distribution is the optimal choice for sampling trajectories, as it does not address the algorithmic difficulty of minimizing the empirical risk via SGD. On a fundamental level, these examples underline the importance of taking into account the data distribution for understanding generalization performance. Section 7.3 elaborates on this problem through the lens of McAllester's PAC-Bayes framework, albeit with Dziugaite et al.'s data-dependent priors (DZIUGAITE; ROY, 2017, 2018; DZIUGAITE; HSU, et al., 2020).

7.2 Overview of our contributions

Before delving into the details, we briefly discuss the technical results under the light of the formalism presented in Chapter 2, alongside the main ideas they were built upon.

Non-vacuous generalization bounds for GFlowNets. A GFlowNet's learning objectives, due to their unboundedness, cannot be directly incorporated into standard PAC-Bayesian theorems (MCALLESTER, D., 2013), which assume that the risk function has at least bounded exponential moments (CASADO et al., 2024; RODRÍGUEZ-GÁLVEZ; THOBABEN; SKOGLUND, 2024). To circumvent this issue, our empirical analysis in Section 7.3.1 adopts the FCS metric introduced in Chapter 6 as the risk functional measuring the accuracy of a trained GFlowNet. Although FCS is restricted in $[0, 1]$ and relatively easy to compute, it is not an appropriate learning objective for GFlowNets. Instead, we minimize the TB loss $\mathcal{L}_{\text{TB}}$ as a surrogate objective for FCS during training and evaluate the generalization bound on FCS for inference in the inductive fashion mentioned in Chapter 2. Importantly, Figure 29 shows that the resulting bounds are remarkably tight.

Oracle generalization bounds for GFlowNets. As a complement, we also establish non-empirical high-probability upper bounds on the population risk of GFlowNets by assuming that a potentially intractable quantity bounds the corresponding loss function. In Section 7.3.2, we follow this rationale and demonstrate that there always is an $\alpha > 0$ for which the set of policy networks of the form $\alpha p_U + (1 - \alpha) p_F$ contains the solution to the flow assignment problem. Armed with such a family of models, which guarantee uniformly bounded log-probabilities, we consider the forward KL divergence risk (MALKIN; LAHLOU, et al., 2023) to avoid explicitly bounding the flow function F. Although insightful, the resulting Theorem 7.3.1 only considers the trajectories—and not transitions—as data points. As the number of observed transitions is significantly larger than that of trajectories, we enrich our results by constructing a martingale difference sequence based on the DB loss and adapting Azuma's inequality (AZUMA, 1967; SELDIN et al., 2012) to the context of independent martingales to derive a transition-level generalization bound for GFlowNets. Both approaches, encapsulated in Theorems 7.3.1 and 7.3.2, show that the population risk can be bounded with high-probability as, apart from technical nuances,

$$\mathbb{E}_{\theta \sim P}[\mathcal{L}(\theta)] \lesssim \mathbb{E}_{\theta \sim P}[\hat{\mathcal{L}}(\theta)] + \mathcal{O}\left(\frac{\log t_m}{n^{\alpha}}\right) \tag{7.2}$$

in which $\hat{\mathcal{L}}$ is an empirical measure of risk, t_m is the maximum trajectory length of the state graph, and n is the number of observed data points—either trajectories (Theorem 7.3.1, $\alpha = 0.5$) or transitions (Theorem 7.3.2, $\alpha = 1$). These results suggest that learning provably generalizable GFlowNets is harder for state spaces having longer trajectories.

Subgraph asynchronous learning (SAL). With this understanding in hand, the following strategy seems clearly beneficial: break up the state graph into small pointed DAGs to reduce the state space that the model needs to care about when learning a flow assignment. This is the central idea behind SAL: instead of learning a single monolithic GFlowNet, we divide the state space into smaller components and learn a different GFlowNet for each (possibly overlapping) component. In a final round, an aggregating GFlowNet is trained to sample these components by using the locally learned flow functions F as reward. To implement this principle, we introduce the concepts of a *fixed-horizon partition* (Definition 7.4.1) and of an *assigment function*. The former is composed of multi-source subnetworks (called *leaves*) connected by a small upper-level pointed DAG (called *root*); see Figure 30. The latter consists of a mapping between each source of the multi-source partition to an available computational unit. A typical implementation of this assignment function involves injectively encoding the leaves' sources as integers and assigning them to a resource based on the remainder after division by the total number of resources. That is, given T leaves and m resources, we would assign the tth leaf to the ($t \pmod{m}$)th resource. For inference, we choose the forward policy determined by this assignment function at each stage of the generative process. Please refer to Publication **IV** for a detailed discussion on how to implement SAL. In this chapter, we demonstrate the soundness of this approach (Theorem 7.4.1) and discuss potential extensions (Proporition 7.4.1) in Section 7.4. Also, we empirically show that SAL often substantially improves over a centralized approach in terms of both mode coverage and accuracy.

7.3 PAC-Bayesian generalization bounds for GFlowNets

In this context, Section 7.3.1 builds upon Dziugaite et al.'s data-dependent priors to derive the first non-vacuous PAC-Bayesian generalization bounds for GFlowNets. Then, Section 7.3.2 provides both trajectory- and transition-level oracle bounds by drawing upon the martingale-based PAC-Bayesian theory for non-i.i.d. data (BEYGELZIMER et al., 2011).

7.3.1 Non-vacuous empirical generalization bounds

GFlowNet learning as supervised learning. To rigorously address the generalization of GFlowNets, we firstly frame the training of these models as a supervised learning problem (SHALEV-SHWARTZ; BEN-DAVID, 2014; ATANACKOVIC; BENGIO, 2024). For this, we assume that a set of independently sampled complete trajectories, $\mathcal{T}_n = \{\tau_1, \dots, \tau_n\}$, is drawn from a fixed distribution and that each trajectory τ_i is annotated with a noise-free target, $y_i = p_B(\tau_i|x_i)R(x_i)$, with x_i representing τ_i's unique terminal state. In this context, minimizing $\mathcal{L}_{\text{TB}}$ is equivalent to the least-squares solution to the problem of finding a policy network p_F such that $\log Z + \log p_F(\tau) = \log p_B(\tau|x)R(x)$. Importantly, this setting differs from the standard ϵ-greedy approach to GFlowNet training, in which the sampling policy depends on the trajectories observed so far. In this case, the trajectories are not independently sampled. Nonetheless, the question of whether GFlowNets generalize remains relevant even under our simplified conditions, which may be seen as a single-iteration of their ϵ-greedy equivalent (KRICHEL et al., 2024).

A bounded risk functional for GFlowNets. PAC-Bayesian theory was originally built upon the assumption of bounded risk functions (MCALLESTER, D. A., 1998). Despite recent advances (CASADO et al., 2024; LOTFI; FINZI, et al., 2024; RODRÍGUEZ-GÁLVEZ; THOBABEN; SKOGLUND, 2024; HADDOUCHE; GUEDJ, et al., 2021; HADDOUCHE; GUEDJ, 2022), most generalization bounds still rely on technical and hardly verifiable assumptions such as bounded exponential moments. Consequently, our experiments will rely on the FCS metric presented in Equation (6.8), which is restricted to $[0, 1]$.

Empirical results. We follow the approach outlined in Proposition 2.4.1 in Chapter 2 to derive the first non-vacuous generalization bounds for GFlowNets in the literature. In particular, we substitute the risk functional $\mathcal{L}$ (resp. $\hat{\mathcal{L}}$) by $L_{\text{FCS}} \coloneqq \text{FCS}$ (resp. $\hat{L}_{\text{FCS}}$) and the data $\mathbf{X}$ by set of trajectories $\mathcal{T}_n$. Also, we disjointly partition the dataset $\mathcal{T}_n$ with $n = 3 \cdot 10^4$ into sets $\mathcal{T}_\alpha$ and $\mathcal{T}_{1-\alpha}$ with $\alpha = 0.6$. We learn an isotropic Gaussian prior Q on $\mathcal{T}_\alpha$ and then a diagonal Gaussian posterior P on $\mathcal{T}_\alpha \cup \mathcal{T}_{1-\alpha}$ by minimizing the bound in Equation (2.4). Finally, the bound is evaluated on $\mathcal{T}_{1-\alpha}$ to obtain the statistical certificate (PÉREZ-ORTIZ et al., 2021). Results in Figure 29 for the tasks of set generation (PAN; MALKIN, et al., 2023a; BENGIO; LAHLOU, et al., 2023), bag generation (SHEN et al., 2023), sequence design (MALKIN; JAIN, et al., 2022; JIRALERSPONG et al., 2023) with additive rewards, and (JAIN et al., 2022) highlight the non-vacuousness of our bound in Equation (2.4). On a fundamental level, this asserts the generalizability of GFlowNets.

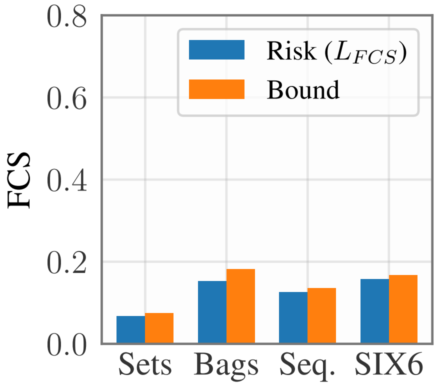

Figure 29 – Non-vacuous bounds for the FCS risk.

7.3.2 Oracle generalization bounds

Although the empirical results in the previous section certify the generalization of the learned policy network to novel trajectories, they do not necessarily shed light on which characteristics of the generative task are hindering the model's generalization capability. In the remaining of this section, we shift our focus to derive generalization bounds providing a better understanding of which factors play a role in the learning of a generalizable policy. As these results depend on quantities that cannot be tractably computed, we refer to them as *oracle bounds*. In particular, we observe that larger trajectories and peakier target distributions tend to make generalization harder when a fixed sampling budget is available. Later, we will see how a distributed algorithm may alleviate these issues (YAGLI; DYTSO; POOR, 2020; BARNES; DYTSO; POOR, 2022; SEFIDGARAN; CHOR; ZAIDI, 2022).

Trajectory-level bounds. We first consider the forward KL divergence between the forward and backward policies, i.e., $\mathrm{KL}(p_B || p_F)$, in which $p_B(\tau) \propto p_B(\tau|x)R(x)$ (recall Chapter 3), as the risk functional. As in Section 7.3.1, we assume that trajectories are i.i.d. sampled. In this setting, Lemma 7.3.1 shows that $\mathrm{KL}(p_B || p_F)$ can be bounded by sensibly reparameterizing p_F as a mixture between an uniform and a learnable policy.

Lemma 7.3.1 (Realizability of mixture policies)**.** *Let $p_U(\cdot|s)$ be the uniform policy. Then, there is an $\alpha \in (0, 1]$ such that the family $\{\tilde{p}_F \colon \tilde{p}_F(\cdot|s) = \alpha p_U(\cdot|s) + (1-\alpha)p_F(\cdot|s)\}$ contains a policy sampling $x \in \mathcal{X}$ in proportion to the reward function.*

In light of Lemma 7.3.1's reparameterization, Theorem 7.3.1 develops oracle generalization bounds for GFlowNets in the fashion of the tight results we derived in Section 7.3.1.

Theorem 7.3.1. *Assume p_F is parameterized as in Lemma 7.3.1. Also, let Q be a distribution over the parameters θ of p_F. Denote $H[p_B] = -\mathbb{E}_{\tau \sim p_B}[\log p_B(\tau)]$ for p_B's entropy and $M_T = \max_\tau (|\tau| \log(\alpha^{-1} \max_{s \in \tau} |\mathrm{Ch}(s)|))$, in which $\mathrm{Ch}(s)$ is the set of s's children. Then,*

$$\mathbb{E}_{\theta \sim P}\left[\mathrm{KL}(\pi || p_T)\right] \leq \mathbb{E}_{\theta \sim P}\left[\frac{1}{m}\sum_{1 \leq i \leq m} \log \frac{p_B(\tau_i)}{p_F(\tau_i)}\right] + \left(-H[p_B] + M_T\right)\eta(P, Q, n), \tag{7.3}$$

in which we recall that $\eta(P, Q, n) = \sqrt{\frac{\mathrm{KL}(P||Q) + \log 2\sqrt{n}/\delta}{n}}$ and $\pi(x) \propto R(x)$ is the target.

A few remarks on the excess risk upper bound of Theorem 7.3.1. Firstly, the assumption that trajectories are sampled according to $p_B(\tau) \propto p_B(\tau|x)R(x)$ is consistent with popular strategies for learning GFlowNets that focus on sampling trajectories leading to high-reward states more often than those leading to low-reward states, e.g., using a replay buffer (DELEU; GÓIS, et al., 2022). Secondly, in alignment with well-established practical knowledge, the result in Equation (7.3) shows it is harder to achieve tighter generalization bounds when the target distribution is peaky with a small entropy term $H[p_B]$, and when the generative task is composed of longer trajectories or larger action spaces.

Transition-level bounds. For many applications, the number of observed complete trajectories when training GFlowNets can be orders of magnitude smaller than the number of collected state transitions. In this context, one may obtain significantly tighter generalization bounds by interpreting the transitions — and not the complete trajectories — as data samples (LOTFI; FINZI, et al., 2024; LOTFI; KUANG, et al., 2024). Indeed, it is assumed that GFlowNets' outstanding potential emerges from its capacity to exploit the compositional structure of the space characterized by the state graph (BENGIO; JAIN, et al., 2021; NICA et al., 2022; SHEN et al., 2023; ATANACKOVIC; BENGIO, 2024). To incorporate this structure into our theoretical bounds, we design Azuma-Hoeffding-type concentration inequalities (AZUMA, 1967; MCDIARMID, 1998; BOUCHERON; LUGOSI; MASSART, 2013) applied to the stochastic process induced by the Markov Decision Process (MDP) governing the data-generating process. For this, we start defining a martingale difference sequence based on the DB loss (BENGIO; LAHLOU, et al., 2023, Example 5).

Definition 7.3.1 (A martingale difference sequence for $\mathcal{L}_{\text{DB}}$)**.** Recall the DB loss $\mathcal{L}_{\text{DB}}(s, s') = (\log F(s) p_F(s'|s) - \log F(s') p_B(s|s'))^2$. For a fixed sampling policy p_E, we let

$$M(S_i, S_{<i}) = \mathcal{L}_{\text{DB}}(S_i, S_{i-1}) - \mathbb{E}_{s_i \sim p_E(\cdot|S_{i-1})} \left[\mathcal{L}_{\text{DB}}(s_i, S_{i-1}) | S_{i-1} \right], \tag{7.4}$$

where $S_{<i} = \{S_1, \dots, S_{i-1}\}$. Also, we define the natural filtration $\mathcal{F}_t = \sigma(S_1, \dots, S_t)$ generated by the first t states of $\{S_i\}_{i \geq 1}$. Clearly, each $M(S_i, S_{<i})$ is $\mathcal{F}_i$-measurable and $\mathbb{E}_{S_i}[M(S_i, S_{<i}) | \mathcal{F}_{<i}] = 0$, i.e., $\{M(S_i, S_{<i})\}_{i \geq 1}$ is a martingale difference sequence.

From Definition 7.3.1, it is immediate that $M_t(\theta) \coloneqq \sum_{1 \leq i \leq t} M(S_i, S_{<i})$ is a martingale with respect to $\{\mathcal{F}_t\}_{t \geq 1}$ depending on p_F's parameters θ. Additionally, we define

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{\tau \sim p_E} \frac{1}{|\tau|} \sum_{1 \leq i \leq |\tau|} \mathbb{E} \left[\mathcal{L}_{\text{DB}}(S_i, S_{i-1}) | S_{i-1} \right] \text{ and } \hat{\mathcal{L}}(\theta) = \frac{1}{n} \sum_{1 \leq j \leq n} \frac{1}{t_j} \sum_{1 \leq i \leq t_j} \mathcal{L}_{\text{DB}}(S_i^{(j)}, S_{i-1}^{(j)})$$

as the population and empirical DB-based risk functionals for GFlowNets. Then, Theorem 7.3.2 complements Theorem 7.3.1 with a generalization bound based on the DB loss.

Theorem 7.3.2 (Transition-level generalization bounds for GFlowNets)**.** *Let $M_t(\theta)$ be the martingale arising from Definition 7.3.1. Also, let Q be a distribution on θ. Assume that $\mathcal{L}_{\text{DB}}(S_i, S_{i-1}) \leq U$ uniformly on (S_i, S_{i-1}) and that $M_{t_m}(\theta)^2 \leq K$, in which t_m is the maximum trajectory length. Similarly, define $\lambda \leq 1/2U$ and $\beta \in (0, 1)$, and let P be a data-dependent posterior distribution on θ. Then, with probability at least $1 - \delta$ over the set of independent martingales $\{S_o, S_1^{(j)}, \dots, S_{t_j}^{(j)}\}_{1 \leq j \leq n}$ such that $S_o = s_o$ almost surely,*

$$\mathbb{E}_{\theta \sim P} \left[\mathcal{L}(\theta) \right] \leq \frac{1}{\beta} \mathbb{E}_{\theta \sim P} \left[\hat{\mathcal{L}}(\theta) \right] + \alpha_{T,n} \left(\text{KL}(P || Q) + \log \frac{2}{\delta} \right) + \frac{\log t_m}{\beta T \lambda} + \gamma \frac{\lambda K}{\beta T},$$

in which T is the number of observed transitions, $\alpha_{T,n} = \left(\frac{U}{2\beta(1-\beta)n} + \frac{1}{\beta T \lambda} \right)$, and $\gamma = e - 2$.

Similarly to Theorem 7.3.1, Theorem 7.3.2 implies that obtaining tighter generalization guarantees is harder for larger state spaces with longer trajectories when the sampling process is limited by a maximum number of observable transitions (or states) T, which is an often imposed constraint for comparing the sample-efficiency of different learning objectives for GFlowNets in the literature (PAN; ZHANG, et al., 2023; PAN; MALKIN, et al., 2023a; MADAN et al., 2022; MALKIN; JAIN, et al., 2022; MALKIN; LAHLOU, et al., 2023). Below, we empirically show that our distributed learning scheme alleviates this effect.

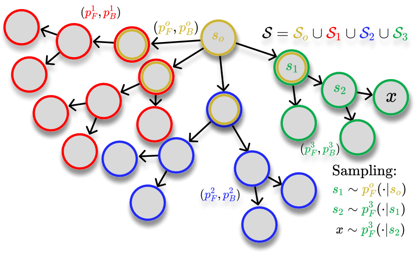
(p_F^1, p_B^1)
(p_F^o, p_B^o)
s_o
$\mathcal{S} = \mathcal{S}_o \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \mathcal{S}_3$
s_1
s_2
x
(p_F^3, p_B^3)
(p_F^2, p_B^2)
Sampling:
$s_1 \sim p_F^o(\cdot|s_o)$
$s_2 \sim p_F^3(\cdot|s_1)$
$x \sim p_F^3(\cdot|s_2)$

Figure 30 – A fixed-horizon DAG partition with three leaves ($\mathcal{S}_1$, $\mathcal{S}_2$, $\mathcal{S}_3$) and one root ($\mathcal{S}_o$). For inference, the sampling policy is chosen based on the current state.

Algorithm 3 Subgraph Asynchronous Learning

```
$\mathcal{S} = \mathcal{S}_o \cup \bigcup_{j=1}^{m} \mathcal{S}_j$    ▷ Fixed-horizon partition
$\mathcal{I}_j = \mathcal{S}_j \cap \mathcal{S}_o$ for $j \in \{1, \dots, m\}$
▷ Local training
parfor $j \in \{1, \dots, m\}$ do
    ▷ Minimize $\mathcal{L}_{\text{ATB}}^{j}$ in $\mathcal{S}_j$ with SGD
    $(p_F^j, F_j) = \arg\min_{p_F, F} \mathcal{L}_{\text{ATB}}^{j}(p_F, F)$
end parfor
$R^o \colon x \mapsto \mathbf{1}_{\{x \in \mathcal{X}\}} R(x) + \sum_{j=1}^{m} \mathbf{1}_{\{x \in \mathcal{I}_j\}} F_j(x)$
$(p_F^o, F_o) = \arg\min_{p_F, F} \mathcal{L}_{\text{TB}}(p_F, F, R^o)$
return $\{(p_F^o, F_o)\} \cup \bigcup_{1 \le j \le m} \{(p_F^j, F_j)\}$
```

7.4 Divide and Conquer: Distributed Learning of GFlowNets

Hitherto, our resulsts indicated that the diverse exploration of state graphs (Section 7.1) with smaller trajectory sizes (Section 7.3) is a desirable property for the successful training of GFlowNets. Henceforth, we show how to efficiently implement these features by recasting the GFlowNet training as an embarrassingly parallel divide-and-conquer algorithm, which we call *subgraph asynchronous learning* (SAL). This differs from Chapter 5's EP-GFlowNets, in which each client learns a distinct distribution over the *same* state graph.

7.4.1 Subgraph Asynchronous Learning

Overview. As briefed in Section 7.2, there are two ingredients making up SAL: a fixed-horizon partition (FHP) and an assignment function (AF). We formally introduce both concepts below. For a clearer understanding, it might be helpful to think about the state graph as a metro map with each state representing a station. Then, each component of the proposed FHP may be seen as a metro line; the AF indicates in which line the GFlowNet agent is in. As in SAL, we may arrive at the same station from different lines.

Convergence guarantees. We define a FHP below. Under the flow network hermeneutics, this partition represents a collection of possibly overlapping multi-source subnetworks (*leaves*) grouped together by a single-source network (*root*). We use the term *fixed-horizon* due to the fixed distance of the subnetworks' sources to s_o. The reader may note that a FHP is only a *partition* in the set-theoretical sense when the state graph is a tree.

Definition 7.4.1 (Fixed-horizon partition)**.** We say that $\mathcal{S} = \mathcal{S}_o \cup \left(\bigcup_{1 \le j \le m} \mathcal{S}_j \right)$ is a FHP of the state space $\mathcal{S}$, with *leaves* $\{S_j\}_{j=1}^{m}$ and *root* $\mathcal{S}_o$, when it satisfies the conditions below:

1. (Disjointness of sources) $s_o \in \mathcal{S}_o$ and the sets $\{\mathcal{I}_j \coloneqq \mathcal{S}_o \cap \mathcal{S}_j\}_{j=1}^{m}$ are pairwise disjoint.
2. (Completeness) If $s \in \mathcal{S}_j$ for a $j \ge 1$, then all descendants of s are in $\mathcal{S}_j$.
3. (Regularity) If d denotes the shortest-path distance, $d(s_o, \mathcal{I}_j) = d(s_o, \mathcal{I}_i)$ for all i, j.

Under Definition 7.4.1, we let $\mathcal{X}_j = \mathcal{S}_j \cap \mathcal{X}$ be the set of terminal states reachable from $\mathcal{I}_j$. For conciseness, we denote $\{\mathcal{S}_j\}_{j=0}^{m} = \text{FHP}(\mathcal{S}, m)$ when $\{\mathcal{S}_j\}_{j=0}^{m}$ is a FHP of $\mathcal{S}$ with m components. Figure 30 illustrates a FHP of a tree in which $m = 3$ and the $\mathcal{I}_j$, represented by the doubly-stroked circles, are singletons for the blue and green leaves. In this context, an AF is the *dual* of a FHP: any AF is uniquely associated to a FHP and vice-versa. Definition 7.4.2 formalizes this. In practice, an AF is a simple mechanism for defining a FHP.

Definition 7.4.2 (Assignment function)**.** Let $f \colon \mathcal{S} \to \{0, 1, \ldots, m\}$, in which m is the number of available computational units. Assume that f satisfies the following.

1. (Completeness). $f^{-1}(j) \neq \emptyset$ for each j, i.e., f assigns some states to every unit;
2. (Consistency). $\{\mathcal{S}_j = f^{-1}(j)\}_{j=0}^{m}$ is a fixed-horizon partition of $\mathcal{S}$.

Then, f is called an *assignment function* and $\{S_j\}_{j=0}^{m}$ is the FHP associated to f.

Definitions 7.4.1 and 7.4.2 form the building blocks of SAL, rigorously defined below.

Definition 7.4.3 (SAL)**.** Let $\{\mathcal{S}_j\}_{j=0}^{m} = \text{FHP}(\mathcal{S}, m)$. For each $1 \leq j \leq m$, let $\mathcal{G}_j = (p_F^j, p_B^j, F_j)$ be a GFlowNet and p_E^j be any forward policy over the $\mathcal{S}_j$-induced subgraph of the state graph. Finally, let q_j be any distribution with full support on $\mathcal{I}_j$. Then, define

$$\mathcal{L}_{\text{ATB}}^{j}(p_F^j, F_j) = \mathbb{E}_{s \sim q_j} \mathbb{E}_{\tau \sim p_E^j(\cdot|s)} \left[\left(\log \frac{F_j(s) p_F^j(\tau|s)}{R(x) p_B^j(\tau|x)} \right)^2 \right], \tag{7.5}$$

in which x represents τ's terminal state, as the *amortized trajectory balance* (ATB) objective. For the root, let p_E^o be a policy in $\mathcal{S}_o$ and $\mathcal{G}_o = (p_F^o, p_B^o, R^o)$. For each $x \in (\mathcal{X} \setminus \bigcup_{j=1}^{m} \mathcal{X}_j) \cup (\bigcup_{j=1}^{m} \mathcal{I}_j)$, let $R^o(x) = F_j(x)$ if $x \in \mathcal{I}_j$ for some j and $R^o(x) = R(x)$ otherwise. In this context, SAL follows a two-step procedure: first, m models are trained in parallel by minimizing Equation (7.5); then, a global model is estimated by optimizing the TB loss with reward R^o, which we denote by $\mathcal{L}_{\text{TB}}(p_F, F, R^o)$ for a GFlowNet (p_F, p_B, F).

At a high level, SAL receives as input a FHP and returns a collection of GFlowNets approximately minimizing the learning objective in Equation (7.5); see Algorithm 3. Clearly, any flow-based loss function (e.g., DB (BENGIO; LAHLOU, et al., 2023), SubTB (MADAN et al., 2022), Munchausen DQN (TIAPKIN et al., 2024b; DELEU; NOURI, et al., 2024), GAFlowNets (PAN; ZHANG, et al., 2023)), parametrizations (e.g., FL (PAN; MALKIN, et al., 2023a), LED (JANG; KIM; AHN, 2024), temperature-scaled (KIM; KO, et al., 2024)), and off-policy sampling strategies (e.g., replay buffer (VEMGAL; LAU; PRECUP, 2023) and local search (KIM; YUN; BENGIO; ZHANG, et al., 2024)) could be employed for estimating both the root and leaf GFlowNets in Definition 7.4.3. In Theorem 7.4.1, we demonstrate that the sampling distribution induced by SAL over the terminal states $\mathcal{X}$ matches the reward R (up to an normalizing constant) when both the leaf and root GFlowNets globally minimize their respective risk functionals. This ensures SAL is a sound GFlowNet training algorithm in the sense of (BENGIO; JAIN, et al., 2021, Proposition 3).

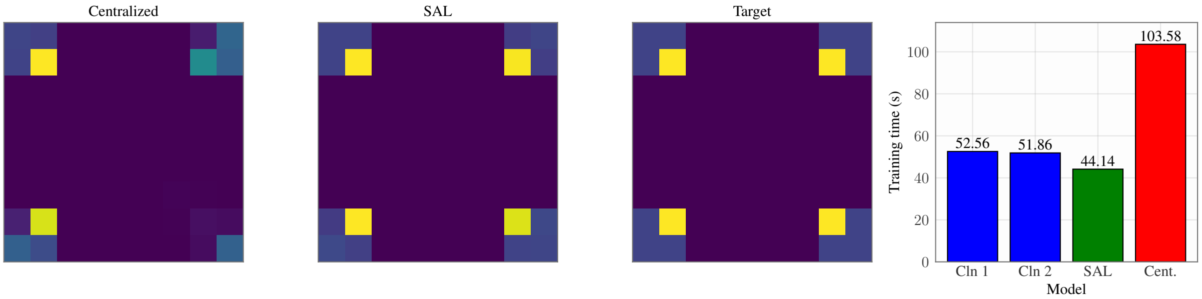

Figure 31 – **SAL improves mode discovery and the distributional accuracy**. Results for a centralized GFlowNet, for our algorithm (SAL), and the target reproduced from (MALKIN; JAIN, et al., 2022, Section 5.1) are shown from left to right. Running time for SAL equals the running time of the longest client plus that of the aggregation phase (right-most plot).

Theorem 7.4.1 (Sampling correctness of SAL). *Let $\{\mathcal{S}_j\}_{j=0}^m = \mathrm{FHP}(\mathcal{S}, m)$ and $\{\mathcal{G}_j\}_{j=0}^m$ be the corresponding GFlowNets. Let $p_F^{\star,o}$ and $\{p_F^{\star,j}\}_{j=1}^m$ be global minimizers of their respective learning objectives. Then, the marginal over $\mathcal{X}$ induced by $\{p_F^{\star,j}\}_{j=0}^m$ and the FHP,*

$$p_\top^\star(x) = \sum_{1\leq j\leq m}\sum_{s\in\mathcal{I}_j}\sum_{\tau'\colon s_o\rightsquigarrow s} p_F^{\star,o}(\tau'|s_o)\sum_{\tau''\colon s\rightsquigarrow x} p_F^{\star,j}(\tau''|s), \tag{7.6}$$

matches the target distribution $\pi(x) \coloneqq {}^{R(x)}/_{Z}$, with $Z = \sum_{x\in\mathcal{X}} R(x)$.

To understand Equation (7.6), recall the metro map viewpoint. We interpret $\mathcal{I}_j$ as a collection of stations (states) in which the GFlowNet agent must go from the root “line" ($\mathcal{S}_o$) to the “line" labelled with j ($\mathcal{S}_j$). In this context, to compute the probability $p_F(\tau|s_o)$ of a trajectory τ starting at s_o and finishing at the terminal state x requires knowing which line-changing station was traversed by the GFlowNet agent when following τ. Hence, if $\tau = [\tau', \tau'']$ with $\tau'\colon s_o \rightsquigarrow s$ and $\tau''\colon s \rightsquigarrow x$ and $s \in \mathcal{I}_j$, then $p_F(\tau|x) = p_F^o(\tau'|s_o)p_F^j(\tau''|s)$. Equation (7.6) is then the marginal of this p_F over $\mathcal{X}$. We prove Theorem 7.4.1 in Appendix B.

Recursive SAL. In the perspective of Definition 7.4.3, SAL has a *single layer*: each leaf is directly connected to the root in the underlying FHP. However, there is no obstacle preventing us from building SAL upon a *multi-layered partition* of the state graph, as illustrated in Figure 32. For this, we must first define a hierarchy of partitions. Then, we recursively learn a flow assignment for each partition by starting at the lowest levels of this hierarchy and moving upwards. Each learning step is based on minimizing the ATB loss via SGD (see Equation (7.5)). The fact that this is a sound GFlowNet training algorithm is ensured by Proposition 7.4.1, which follows from Theorem 7.4.1 and an inductive argument.

Proposition 7.4.1 (Recursive SAL). *Let $\mathcal{S}$ be the vertices of a state graph with diameter D. Then, for sequences $0 = d_o < d_1 < d_2 < \cdots < d_k \leq D$ and $\{m_o = 1, m_1, \ldots, m_k\}$, we let $\bigcup_{1\leq j\leq m_i} \mathcal{I}_{ij}$ be a disjoint m_i-partition of the states distanced d_i from s_o. Also, let $d(\cdot,\cdot)$ be the shortest-path distance, let $\mathcal{X}_k = \{x \in \mathcal{X}\colon d(x, s_o) \geq d_k\}$, and, for $i < k$, let $\mathcal{X}_i = \{s\colon s \in \mathcal{I}_{i+1,j} \vee (d(s, s_o) \leq d_i \wedge s \in \mathcal{X})\}$. Finally, we define $\mathcal{G}_i = \{(p_F^{i,j}, p_B^{i,j}, F_{ij})\colon 1 \leq$*

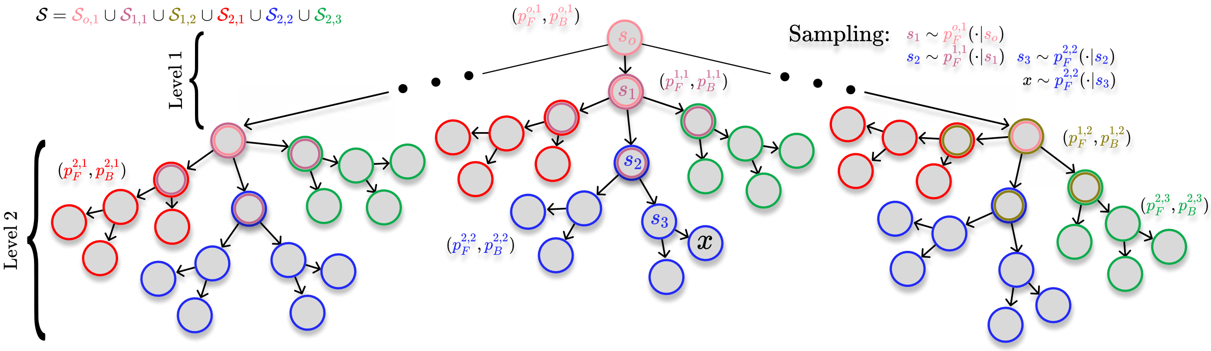
$\mathcal{S} = \mathcal{S}_{o,1} \cup \mathcal{S}_{1,1} \cup \mathcal{S}_{1,2} \cup \mathcal{S}_{2,1} \cup \mathcal{S}_{2,2} \cup \mathcal{S}_{2,3}$
$(p_F^{o,1}, p_B^{o,1})$
Sampling: $s_1 \sim p_F^{o,1}(\cdot|s_o)$
$s_2 \sim p_F^{1,1}(\cdot|s_1)$ $s_3 \sim p_F^{2,2}(\cdot|s_2)$
$x \sim p_F^{2,2}(\cdot|s_3)$
Level 1
Level 2
s_o
s_1
$(p_F^{1,1}, p_B^{1,1})$
$(p_F^{1,2}, p_B^{1,2})$
$(p_F^{2,1}, p_B^{2,1})$
s_2
s_3
x
$(p_F^{2,2}, p_B^{2,2})$
$(p_F^{2,3}, p_B^{2,3})$

Figure 32 – **Illustration of Recursive SAL.** We show a two-level partition with $m_1 = 2$ models within the first level and $m_2 = 3$ models within the second one. For training, we first train models at the bottommost layer (represented in blue, red, and green) and recursively proceed upwards towards the middle (magenta and yellow) and top (root partition, shown in pink) layers. For the non-root layers, learning is based on minimizing $\mathcal{L}_{\text{ATB}}$ with the reward defined as in Equation (7.7); for the root, we minimize $\mathcal{L}_{\text{TB}}$ instead. For inference, we start at s_o and iteratively select the policy based on the current state, as illustrated in the highlighted trajectory and in the annotated text on the top-right corner.

$j \leq m_i\}$ as a set of GFlowNets trained on a state graph starting at $\bigcup_j \mathcal{I}_{ij}$ and finishing at the terminal states $\mathcal{X}_i$ with reward function $R_i \colon \mathcal{X}_i \to \mathbb{R}_+$ such that

$$R_i(s) = \begin{cases} F_{i+1}(s), & \textit{if } s \in \mathcal{I}_{i+1} \textit{ and } i < k, \\ R(s), & \textit{if } s \in \mathcal{X}. \end{cases} \tag{7.7}$$

Then, when the GFlowNets $\cup_{i=0}^{k} \mathcal{G}_i$ satisfy their respective balance conditions, the generative process starting at s_o and recursively following $p_F^{i,b}$ until either reaching $\mathcal{I}_{i+1,a}$, at which point the guiding forward policy is changed to $p_F^{i+1,a}$, for $0 \leq i \leq k$, or reaching $\mathcal{X}$, at which point to stop the current object is returned, samples each $x \in \mathcal{X}$ in proportion to $R(x)$.

Although we do not provide an empirical evaluation of Recursive SAL in this thesis, we are optimistic about its potential applications in tasks with very large trajectory sizes. Also, readers may consult the supplement of Publication **IV** for an extensive discussion regarding the character of the AF, the properties of the local distributions learned by the leaf GFlowNets, and the sensibility of SAL to error propagation.

7.4.2 Empirical illustration

The previous subsection established the theoretical foundations of SAL as a sound algorithm for the non-localized training of GFlowNets. Nonetheless, the problem of whether a distributed approach outperforms a centralized GFlowNet according to conventional metrics is an empirical question — that we positively answer below.

Experimental setup. We evaluate the performance of SAL in six different generative tasks. Please refer to Section 3.4 for a description of the corresponding MDPs.

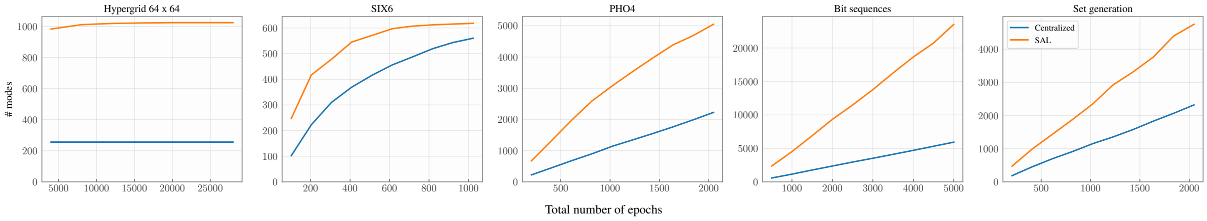

Figure 33 – **SAL enacts faster mode discovery** under varying time budgets. The horizontal axis represents the total number of epochs used by the centralized model, which is set to the sum of the number of epochs for each leaf GFlowNet and the root GFlowNet to ensure the approaches are fairly compared. We provide additional evidence for the enhance performance of SAL in Figure 34.

1. **Hypergrid** (BENGIO; JAIN, et al., 2021; MALKIN; JAIN, et al., 2022; MALKIN; LAHLOU, et al., 2023; PAN; ZHANG, et al., 2023; KRICHEL et al., 2024). We consider both a 8 x 8 and a 64 x 64 hypergrid environment with (MALKIN; JAIN, et al., 2022, Section 5.1)'s reward function, which is illustrated in Figure 31 for $H = 8$.

2. **SIX6** (JAIN et al., 2022; MALKIN; JAIN, et al., 2022; SHEN et al., 2023; CHEN; MAUCH, 2024; KIM; KO, et al., 2024). We generate 8-sized nucleotide strings. The reward represents wet-lab DNA binding measurements to a human transcription factor (BARRERA et al., 2016; TRABUCCO et al., 2022).

3. **PHO4** (JAIN et al., 2022; MALKIN; JAIN, et al., 2022; SHEN et al., 2023; CHEN; CHEWI, et al., 2023). Similarly, we construct 10-sized nucleotide strings; the reward reflects wet-lab measurements of DNA binding activities to a yeast transcription factor (BARRERA et al., 2016; TRABUCCO et al., 2022).

4. **Bit sequences** (MADAN et al., 2022; RECTOR-BROOKS et al., 2023b; TIAPKIN et al., 2024b). We produce 60-sized binary sequences. Given a subset M of such sequences, we define $R(x) = \exp\{-\min_{m \in \mathcal{M}} d_L(x, m)\}$, in which d_L is the edit distance.

5. **Sequence design** (JAIN et al., 2022; SILVA et al., 2024). We build 8-sized sequences of $\{1, \dots, 6\}$. Also, $R(x) = \sum_{i=1}^{8} g(i) f(x_i)$, with f, g being $[-1, 1]$-valued functions.

6. **Set generation** (BENGIO; LAHLOU, et al., 2023; PAN; MALKIN, et al., 2023a). We assemble 16-sized subsets of a 32-sized set with the reward function of Section 7.1.

Results. As expected, Figures 31, 33, and 34 show that SAL drastically speeds up the discovery of high-valued states for all considered generative problems under varying computational constraints. Complementarily, Table 6 suggests that SAL leads to a more accurate approximation for the set generation and sequence design tasks.

	Sets	Sequences
Centralized	$0.092_{\pm 0.001}$	$0.126_{\pm 0.012}$
SAL	$\mathbf{0.072}_{\pm 0.008}$	$\mathbf{0.094}_{\pm 0.005}$

Table 6 – TV distance between target and learned distributions for a centralized model and SAL.

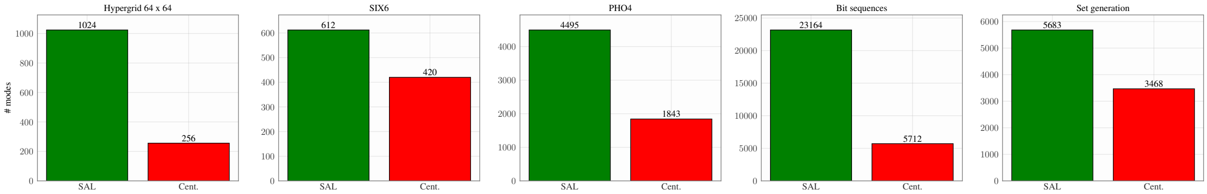

Figure 34 – **SAL improves the discovery of high-valued states** for all considered tasks. For a fair comparison, the centralized model is allowed to explore for twice the number of epochs permitted to each client, ensuring the training times are roughly the same for SAL and the standard GFlowNet.

7.5 Chapter remarks

Conventional wisdom has attributed the impressive performance of GFlowNets to their ability to exploit the compositional structure of the state space to learn a generalizable flow assignment. Despite the abundance of empirical evidence supporting this hypothesis, however, a theoretical understanding of the generalizability of GFlowNets has received far less attention from the literature. To fill this gap, this chapter developed the first PAC-Bayesian bounds and verifiably non-vacuous statistical guarantees for the generalization of GFlowNets. Additionally, our theoretical results provided a deeper understanding of the negative effect of the trajectory length on the proven learnability of a generalizable policy (**RQIII**). Inspired by these conclusions, our distributed algorithm SAL, which is also the first of its kind, exhibited promising results in both synthetic and real-world problems according to the usual metrics of mode coverage and distributional accuracy (**RQII**).

Importantly, these contributions pave the road for future investigations in both the fundamental properties and methodological improvements of GFlowNets. On the one hand, our theoretical analysis in Section 7.3 was restricted to the case of i.i.d. sampled trajectories. Nonetheless, practical approaches for GFlowNet learning often rely on sophisticated sampling schemes — such as ϵ-greedy (BENGIO; JAIN, et al., 2021) and replay buffer (DELEU; GÓIS, et al., 2022) — that do not fit in this setting. In this sense, a principled understanding of the effectiveness of these strategies for the learning of GFlowNets is still lacking, and it is natural to ask whether they facilitate generalization. On the other hand, the assignment functions defining our FHPs were heuristically derived. Albeit the resulting approach was demonstrably effective, we believe there is room for improvement. Theorem 7.3.1, for example, suggests that a low entropy distribution is easier to approximate; this information could be accounted for in the design of the assignment function.

Finally, it has not escaped our notice that SAL is related to (MANKOWITZ; MANN; MANNOR, 2016)'s Adaptive Skills, Adaptive Partitions (ASAP) framework for learning temporally extended actions in MDPs and may find fruitful applications in multi-task RL by interpreting each leaf (resp. root) GFlowNet as an intra- (resp. inter-) skill policy.

7.6 Bibliographical notes

The benefits of distributed approaches for generalization performance of learning algorithms were also pointed out by (YAGLI; DYTSO; POOR, 2020; BARNES; DYTSO; POOR, 2022; SEFIDGARAN; CHOR; ZAIDI, 2022); similarly to SAL, these authors consider the problem of training a set of models in parallel and subsequently aggregating them with a (possibly randomized) estimator in a central server. In a recent work, Eran Malach (MALACH, 2024) introduced the notion of *length complexity* for next-token autoregressive learning on Chain-of-Thought data, referring to the minimum number of iterations required by an AR learner to compute a target function, which seems to be (vaguely) connected to our results regarding the harmful effects of the maximum trajectory size on GFlowNet learning. In the GFlowNet realm, we are confident that problems such as language modelling (HU et al., 2023) and drug discovery (BENGIO; JAIN, et al., 2021; PANDEY; SUBBARAJ; BENGIO, 2024b) could greatly benefit from SAL if appropriate policy networks and fixed-horizon partitionings are designed. Nonetheless, given the open-endedness and specialized nature of these applications, we believe that they would be more suited for future, dedicated works and are not addressed in this text.

8 Conclusions and open problems

This thesis focused on the development of principled approaches for the non-localized training and assessment of GFlowNets. Our main argument is that progress in this field will be enabled by three primary ingredients: statistical efficiency (**RQI**), computational scalability (**RQII**), and theoretical groundedness (**RQIII**). To achieve this, we designed better learning objectives and effective streaming, distributed, and asynchronous algorithms for efficiently training GFlowNets in modern computer clusters on temporally evolving target distributions. Along the way, we pioneered a theory to understand the limitations and generalization of GFlowNets under the lens of a novel tractable and sound evaluation metric for probing the models' correctness.

From a methodological perspective, we introduced SB-GFlowNets (Publication **II**), EP-GFlowNets (Publication **I**), and SAL (Publication **IV**) as simple approaches for scaling GFlowNets beyond the context of centralized and static training. In essence, the proposed algorithms are based on the training of a GFlowNet on top of other GFlowNet(s). On the one hand, the demonstrated succcess of this strategy underlines the inherent *compositionality* of these models. This fact was also noticed by Garipov et al. (GARIPOV et al., 2023), who showed how to perform certain binary operations between GFlowNets. In this context, it is natural to ask: in which additional ways can we usefully compose a collection of pre-trained GFlowNets? A clear answer to this question remains largely elusive. On the other hand, EP-GFlowNets and SAL rely on a stochastic aggregation step operationalized in a centralized server. However, the flow assignment problem determined by the aggregating phase of each method can be deterministically solved under specific conditions, e.g., for tree-structured state graphs (for EP-GFlowNets) and for FHPs with a small root (for SAL), and a complete characterization of these conditions would be desirable.

At a fundamental level, we established the first non-vacuous statistical guarantees for the learning of GFlowNets in the literature (Publication **IV**). As generalization is thought to play a major role in the effectiveness of these models (BENGIO; JAIN, et al., 2021), this marks an important advancement in the understanding of GFlowNets. In a broader context, the search for tighter (PAC-Bayesian) generalization bounds under more relaxed assumptions on the risk functional and for richly structured data sets is an active research topic, and we think that GFlowNets provide a natural testbed for these developments.

Additionally, we demonstrated that a restrictive parameterization (Publication **V**) and a large gradient variance (Publication **III**) of the policy networks critically hinder the learning efficiency of GFlowNets. Although we mitigated these issues with the highly expressive LA-GFlowNet and statistically efficient gradient estimators, respectively, further improvements are necessary to promote the widespread adoption of these models as general-purpose tools for approximate Bayesian inference. For continuous distributions, in particular, GFlowNets seem to fall short compared to HMC. We believe that improving credit assignment (MALKIN; JAIN, et al., 2022) via reward shaping techniques (NG; HARADA; RUSSELL, 1999; MARTHI, 2007) and combining GFlowNets with MCMC (RUIZ; TITSIAS, 2019; DELEU; BENGIO, 2023) for problems having discrete and continuous components, such as Bayesian phylogenetic inference (ZHOU et al., 2024), are promising and underexplored directions for achieving this objective.

In conclusion, GFlowNets have consistently proven to be the best option for sampling from distributions over discrete spaces. As we have demonstrated, these models are also seamlessly composable and in alignment with the current trend in machine learning to build *generative systems* comprising diversely specialized models instead of a single large-scale generative neural network (DU; KAELBLING, 2024). With that said, we note GFlowNets have been used to approximately solve NP-hard problems (ZHANG; DAI, et al., 2023; ZHANG, D. W. et al., 2023), and a key unaddressed theoretical question is whether these models can be efficiently PAC learned using a neural network hypothesis class.

References

ABADI, Martin et al. Deep Learning with Differential Privacy. In: PROCEEDINGS of the 2016 ACM SIGSAC Conference on Computer and Communications Security. [S.l.]: ACM, Oct. 2016. (CCS'16). DOI: 10.1145/2976749.2978318. Available from: <http://dx.doi.org/10.1145/2976749.2978318>.

AHMADIAN, Arash et al. **Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs**. [S.l.: s.n.], 2024. arXiv: 2402.14740 [cs.LG]. Available from: <https://arxiv.org/abs/2402.14740>.

ALQUIER, Pierre. User-friendly introduction to PAC-Bayes bounds. **arXiv preprint arXiv:2110.11216**, 2021.

________. User-friendly introduction to PAC-Bayes bounds. **Foundations and Trends® in Machine Learning**, Now Publishers, Inc., v. 17, n. 2, p. 174–303, 2024. arXiv: 2110.11216 [stat.ML]. Available from: <https://arxiv.org/abs/2110.11216>.

ATANACKOVIC, Lazar; BENGIO, Emmanuel. **Investigating Generalization Behaviours of Generative Flow Networks**. [S.l.: s.n.], 2024. arXiv: 2402.05309 [cs.LG]. Available from: <https://arxiv.org/abs/2402.05309>.

ATANACKOVIC, Lazar; TONG, Alexander, et al. DynGFN: Towards Bayesian Inference of Gene Regulatory Networks with GFlowNets. In: THIRTY-SEVENTH Conference on Neural Information Processing Systems. [S.l.: s.n.], 2023. Available from: <https://openreview.net/forum?id=e7MK5Vq44Q>.

AXLER, Sheldon. **Measure, Integration &; Real Analysis**. [S.l.]: Springer International Publishing, 2020. ISBN 9783030331436. DOI: 10.1007/978-3-030-33143-6.

AZUMA, Kazuoki. Weighted sums of certain dependent random variables. **Tohoku Mathematical Journal, Second Series**, Mathematical Institute, Tohoku University, v. 19, n. 3, p. 357–367, 1967.

BARNES, Leighton Pate; DYTSO, Alex; POOR, Harold Vincent. Improved Information-Theoretic Generalization Bounds for Distributed, Federated, and Iterative Learning. **Entropy**, MDPI AG, v. 24, n. 9, p. 1178, Aug. 2022. ISSN 1099-4300. DOI: 10.3390/e24091178. Available from: <http://dx.doi.org/10.3390/e24091178>.

BARRERA, Luis A et al. Survey of variation in human transcription factors reveals prevalent DNA binding changes. **Science**, American Association for the Advancement of Science, v. 351, n. 6280, p. 1450–1454, 2016.

BAYDIN, Atilim Gunes et al. Automatic differentiation in machine learning: a survey. **Journal of Marchine Learning Research**, Microtome Publishing, 2018.

BENGIO, Emmanuel; JAIN, Moksh, et al. Flow Network based Generative Models for Non-Iterative Diverse Candidate Generation. In: NEURIPS (NeurIPS). [S.l.: s.n.], 2021.

BENGIO, Yoshua; LAHLOU, Salem, et al. GFlowNet Foundations. **Journal of Machine Learning Research (JMLR)**, 2023.

BEYGELZIMER, Alina et al. **Contextual Bandit Algorithms with Supervised Learning Guarantees**. [S.l.: s.n.], 2011. arXiv: `1002.4058 [cs.LG]`. Available from: <`https://arxiv.org/abs/1002.4058`>.

BISHOP, Christopher M. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2007.

BISSIRI, P. G.; HOLMES, C. C.; WALKER, S. G. A General Framework for Updating Belief Distributions. **Journal of the Royal Statistical Society Series B: Statistical Methodology**, v. 78, n. 5, p. 1103–1130, Feb. 2016. DOI: `10.1111/rssb.12158`.

BLEI, David M.; AL., et. Variational Inference: A Review for Statisticians. **Journal of the American Statistical Association**, 2017.

BLESSING, Denis et al. **Beyond ELBOs: A Large-Scale Evaluation of Variational Methods for Sampling**. [S.l.: s.n.], 2024. arXiv: `2406.07423 [cs.LG]`. Available from: <`https://arxiv.org/abs/2406.07423`>.

BOUCHERON, Stéphane; LUGOSI, Gábor; MASSART, Pascal. **Concentration inequalities: A nonasymptotic theory of independence**. [S.l.]: Oxford University Press, 2013.

BRADBURY, James et al. **JAX: composable transformations of Python+NumPy programs**. [S.l.: s.n.], 2018.

BRODERICK, Tamara et al. Streaming Variational Bayes. In: NEURIPS. [S.l.: s.n.], 2013.

________. ________. In________. **Advances in Neural Information Processing Systems**. [S.l.]: Curran Associates, Inc., 2013. v. 26.

BUESING, Lars; HEESS, Nicolas; WEBER, Theophane. Approximate inference in discrete distributions with monte carlo tree search and value functions. In: PMLR. AISTATS. [S.l.: s.n.], 2020. P. 624–634.

BUI, Thang D; NGUYEN, Cuong; TURNER, Richard E. Streaming sparse Gaussian process approximations. **NeurIPS**, 2017.

BURDA, Yuri; GROSSE, Roger B.; SALAKHUTDINOV, Ruslan. **Importance Weighted Autoencoders**. [S.l.: s.n.], 2016.

CARBONETTO, Peter; KING, Matthew; HAMZE, Firas. A stochastic approximation method for inference in probabilistic graphical models. **NeurIPS**, 2009.

CARPENTER, Bob et al. Stan: A probabilistic programming language. **Journal of statistical software**, Columbia Univ., New York, NY (United States); Harvard Univ., Cambridge, MA (United States), 2017.

CASADO, Ioar et al. **PAC-Bayes-Chernoff bounds for unbounded losses**. [S.l.: s.n.], 2024. arXiv: 2401.01148 [stat.ML]. Available from: <https://arxiv.org/abs/2401.01148>.

CATONI, Olivier. PAC-Bayesian supervised classification: the thermodynamics of statistical learning. **arXiv preprint arXiv:0712.0248**, 2007.

CHEN, Sitan; CHEWI, Sinho, et al. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In: THE Eleventh International Conference on Learning Representations. [S.l.: s.n.], 2023.

CHEN, Yihang; MAUCH, Lukas. Order-Preserving GFlowNets. In: THE Twelfth International Conference on Learning Representations. [S.l.: s.n.], 2024.

COLE, T. J. Algorithm AS 281: Scaling and Rounding Regression Coefficients to Integers. **Applied Statistics**, JSTOR, v. 42, n. 1, 1993. ISSN 0035-9254. DOI: 10.2307/2347432.

CORSO, Gabriele et al. Graph neural networks. **Nature Reviews Methods Primers**, Nature Publishing Group UK London, v. 4, n. 1, p. 17, 2024.

CSISZÁR, Imre; KÖRNER, János. **Information theory: coding theorems for discrete memoryless systems**. [S.l.]: Cambridge University Press, 2011.

DE SOUZA, Daniel A. et al. Parallel MCMC Without Embarrassing Failures. In: AISTATS. [S.l.: s.n.], 2022.

DELEU, Tristan; BENGIO, Yoshua. **Generative Flow Networks: a Markov Chain Perspective**. [S.l.: s.n.], 2023. arXiv: 2307.01422 [cs.LG]. Available from: <https://arxiv.org/abs/2307.01422>.

DELEU, Tristan; GÓIS, António, et al. Bayesian Structure Learning with Generative Flow Networks. In: UAI. [S.l.: s.n.], 2022.

DELEU, Tristan; NISHIKAWA-TOOMEY, Mizu, et al. Joint Bayesian Inference of Graphical Structure and Parameters with a Single Generative Flow Network. In: ADVANCES in Neural Processing Systems (NeurIPS). [S.l.: s.n.], 2023.

DELEU, Tristan; NOURI, Padideh, et al. **Discrete Probabilistic Inference as Control in Multi-path Environments**. [S.l.: s.n.], 2024. arXiv: 2402.10309 [cs.LG]. Available from: <https://arxiv.org/abs/2402.10309>.

DEPEWEG, Stefan et al. Learning and policy search in stochastic dynamical systems with bayesian neural networks. **arXiv preprint arXiv:1605.07127**, 2016.

DINH, Vu; DARLING, Aaron E; MATSEN IV, Frederick A. Online Bayesian Phylogenetic Inference: Theoretical Foundations via Sequential Monte Carlo. Ed. by Edward Susko. **Systematic Biology**, Oxford University Press (OUP), v. 67, n. 3, Dec. 2017. ISSN 1076-836X. DOI: 10.1093/sysbio/syx087.

DOMKE, Justin. Provable Gradient Variance Guarantees for Black-Box Variational Inference. In: NEURIPS. [S.l.: s.n.], 2019. P. 328–337.

________. Provable Smoothness Guarantees for Black-Box Variational Inference. In: ICML. [S.l.]: PMLR, 2020. v. 119. (Proceedings of Machine Learning Research), p. 2587–2596.

DU, Yilun; KAELBLING, Leslie. **Compositional Generative Modeling: A Single Model is Not All You Need**. [S.l.: s.n.], 2024. arXiv: 2402.01103 [cs.LG]. Available from: <https://arxiv.org/abs/2402.01103>.

DUBEY, Kumar Avinava et al. Variance Reduction in Stochastic Gradient Langevin Dynamics. In: NEURIPS. [S.l.: s.n.], 2016.

DUCHI, John; HAZAN, Elad; SINGER, Yoram. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of machine learning research**, v. 12, n. 7, 2011.

DZIUGAITE, Gintare Karolina; HSU, Kyle, et al. **On the role of data in PAC-Bayes bounds**. [S.l.: s.n.], 2020. arXiv: 2006.10929 [cs.LG]. Available from: <https://arxiv.org/abs/2006.10929>.

DZIUGAITE, Gintare Karolina; ROY, Daniel M. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. **arXiv preprint arXiv:1703.11008**, 2017.

________. Data-dependent PAC-Bayes priors via differential privacy. **Advances in Neural Information Processing Systems**, v. 31, 2018.

FANG, Shikai et al. Streaming Bayesian Deep Tensor Factorization. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2021.

FELSENSTEIN, Joseph. Evolutionary trees from DNA sequences: A maximum likelihood approach. **Journal of Molecular Evolution**, 1981.

GARIPOV, Timur et al. Compositional Sculpting of Iterative Generative Processes. In: THIRTY-SEVENTH Conference on Neural Information Processing Systems. [S.l.: s.n.], 2023.

GEYER, Charles J. Markov chain Monte Carlo maximum likelihood. Interface Foundation of North America, 1991.

GOODFELLOW, Ian J. et al. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. In: 2ND International Conference on Learning Representations, ICLR 2014. [S.l.: s.n.], 2014.

GRAZIANI, Caterina et al. The Expressive Power of Path-Based Graph Neural Networks. In: INTERNATIONAL Conference on Machine Learning (ICML). [S.l.: s.n.], 2024.

GUEDJ, Benjamin. **A Primer on PAC-Bayesian Learning**. [S.l.: s.n.], 2019. arXiv: 1901.05353 [stat.ML]. Available from: <https://arxiv.org/abs/1901.05353>.

HADDOUCHE, Maxime; GUEDJ, Benjamin. PAC-Bayes generalisation bounds for heavy-tailed losses through supermartingales. **arXiv preprint arXiv:2210.00928**, 2022.

HADDOUCHE, Maxime; GUEDJ, Benjamin, et al. PAC-Bayes Unleashed: Generalisation Bounds with Unbounded Losses. **Entropy**, MDPI AG, v. 23, n. 10, p. 1330, Oct. 2021. ISSN 1099-4300. DOI: 10.3390/e23101330. Available from: <http://dx.doi.org/10.3390/e23101330>.

HAMILTON, William L. **Graph representation learning**. [S.l.]: Morgan & Claypool Publishers, 2020.

HAN, Jun et al. Stein variational inference for discrete distributions. In: AISTATS. [S.l.: s.n.], 2020.

HINTON, Geoffrey; SRIVASTAVA, Nitish; SWERSKY, Kevin. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent, 2012.

HINTON, Geoffrey E. Training Products of Experts by Minimizing Contrastive Divergence. **Neural Computation**, v. 14, n. 8, Aug. 2002.

HORNBERGER, John C.; HABRAKEN, Hilde; BLOCH, Daniel A. Minimum Data Needed on Patient Preferences for Accurate, Efficient Medical Decision Making. **Medical Care**, Ovid Technologies (Wolters Kluwer Health), v. 33, n. 3, Mar. 1995. ISSN 0025-7079. DOI: 10.1097/00005650-199503000-00008.

HORNIK, Kurt; STINCHCOMBE, Maxwell; WHITE, Halbert. Multilayer feedforward networks are universal approximators. **Neural networks**, Elsevier, v. 2, n. 5, p. 359–366, 1989.

HU, Edward J. et al. **Amortizing intractable inference in large language models**. [S.l.: s.n.], 2023. arXiv: 2310.04363 [cs.LG].

HUANG, Zhishen; BECKER, Stephen. Stochastic Gradient Langevin Dynamics with Variance Reduction. **CoRR**, 2021.

HYVÄRINEN, Aapo. Some extensions of score matching. **Computational statistics & data analysis**, Elsevier, v. 51, n. 5, p. 2499–2512, 2007.

JAIN, Moksh et al. Biological Sequence Design with GFlowNets. In: INTERNATIONAL Conference on Machine Learning (ICML). [S.l.: s.n.], 2022.

JANG, Eric; GU, Shixiang; POOLE, Ben. Categorical Reparameterization with Gumbel-Softmax. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2017.

JANG, Hyosoon; KIM, Minsu; AHN, Sungsoo. Learning Energy Decompositions for Partial Inference in GFlowNets. In: THE Twelfth International Conference on Learning Representations. [S.l.: s.n.], 2024.

JIRALERSPONG, Marco et al. **Expected flow networks in stochastic environments and two-player zero-sum games**. [S.l.: s.n.], 2023. arXiv: 2310.02779 [cs.LG].

JORDAN, Michael I. et al. An Introduction to Variational Methods for Graphical Models. **Mach. Learn.**, v. 37, n. 2, p. 183–233, 1999.

JUKES, Thomas H; CANTOR, Charles R. Evolution of Protein Molecules. In: MAMMALIAN Protein Metabolism. [S.l.]: Elsevier, 1969.

KIM, Kyurae; MA, Yian; GARDNER, Jacob. Linear Convergence of Black-Box Variational Inference: Should We Stick the Landing? In: AISTATS. [S.l.]: PMLR, 2024. v. 238. (Proceedings of Machine Learning Research), p. 235–243.

KIM, Minsu; KO, Joohwan, et al. **Learning to Scale Logits for Temperature-Conditional GFlowNets**. [S.l.: s.n.], 2024. arXiv: 2310.02823 [cs.LG]. Available from: <https://arxiv.org/abs/2310.02823>.

KIM, Minsu; YUN, Taeyoung; BENGIO, Emmanuel;
DINGHUAI ZHANG, Yoshua Bengio, et al. Local search GFlowNets. In: INTERNATIONAL Conference on Learning Representations (ICLR). [S.l.: s.n.], 2024.

KIM, Minsu; YUN, Taeyoung; BENGIO, Emmanuel; ZHANG, Dinghuai, et al. **Local Search GFlowNets**. [S.l.: s.n.], 2024. arXiv: 2310.02710 [cs.LG]. Available from: <https://arxiv.org/abs/2310.02710>.

________. Local search gflownets. **arXiv preprint arXiv:2310.02710**, 2023.

KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

KINGMA, Diederik P; GAO, Ruiqi. Understanding Diffusion Objectives as the ELBO with Simple Data Augmentation. In: THIRTY-SEVENTH Conference on Neural Information Processing Systems. [S.l.: s.n.], 2023.

KINGMA, Diederik P et al. On Density Estimation with Diffusion Models. In________. **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2021.

KINGMA, Diederik P. et al. **Variational Diffusion Models**. [S.l.: s.n.], 2023. arXiv: 2107.00630 [cs.LG].

KINGMA, Durk P et al. Semi-supervised learning with deep generative models. **NeurIPS**, 2014.

KINOSHITA, Yuri; SUZUKI, Taiji. Improved Convergence Rate of Stochastic Gradient Langevin Dynamics with Variance Reduction and its Application to Optimization. In: NEURIPS. [S.l.: s.n.], 2022.

KIPF, Thomas N; WELLING, Max. Semi-supervised classification with graph convolutional networks. **arXiv preprint arXiv:1609.02907**, 2016.

KRICHEL, Anas et al. **On Generalization for Generative Flow Networks**. [S.l.: s.n.], 2024. arXiv: 2407.03105 [cs.LG]. Available from: <https://arxiv.org/abs/2407.03105>.

KUCUKELBIR, Alp et al. Automatic Differentiation Variational Inference. **J. Mach. Learn. Res.**, v. 18, 14:1–14:45, 2017.

KULLBACK, S.; LEIBLER, R. A. On Information and Sufficiency. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, 1951.

LAHLOU, Salem et al. A theory of continuous generative flow networks. In: ICML. [S.l.]: PMLR, 2023. v. 202. (Proceedings of Machine Learning Research), p. 18269–18300.

LAU, Elaine et al. **QGFN: Controllable Greediness with Action Values**. [S.l.: s.n.], 2024. arXiv: 2402.05234 [cs.LG]. Available from: <https://arxiv.org/abs/2402.05234>.

LI, Yingzhen; TURNER, Richard E. Rényi divergence variational inference. **NeurIPS**, v. 29, 2016.

LINDLEY, Dennis Victor. **Bayesian statistics: A review**. [S.l.]: SIAM, 1972.

LIU, Dianbo; AL., et. GFlowOut: Dropout with Generative Flow Networks. In: INTERNATIONAL Conference on Machine Learning. Honolulu, Hawaii, USA: JMLR.org, 2023. (ICML'23).

LIU, Jun S; LIU, Jun S. **Monte Carlo strategies in scientific computing**. [S.l.]: Springer, 2001. v. 10.

LOTFI, Sanae; FINZI, Marc, et al. **Non-Vacuous Generalization Bounds for Large Language Models**. [S.l.: s.n.], 2024. arXiv: 2312.17173 [stat.ML]. Available from: <https://arxiv.org/abs/2312.17173>.

LOTFI, Sanae; KUANG, Yilun, et al. **Unlocking Tokens as Data Points for Generalization Bounds on Larger Language Models**. [S.l.: s.n.], 2024. arXiv: 2407.18158 [stat.ML]. Available from: <https://arxiv.org/abs/2407.18158>.

MA, Jianzhu et al. Estimating the Partition Function of Graphical Models Using Langevin Importance Sampling. In: AISTATS. [S.l.]: PMLR, 2013.

MADAN, Kanika et al. Learning GFlowNets from partial episodes for improved convergence and stability. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2022.

MADDISON, Chris J.; MNIH, Andriy; TEH, Yee Whye. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2017.

MALACH, Eran. **Auto-Regressive Next-Token Predictors are Universal Learners**. [S.l.: s.n.], 2024. arXiv: 2309.06979 [cs.LG]. Available from: <https://arxiv.org/abs/2309.06979>.

MALKIN, Nikolay; JAIN, Moksh, et al. Trajectory balance: Improved credit assignment in GFlowNets. In: NEURIPS (NeurIPS). [S.l.: s.n.], 2022.

MALKIN, Nikolay; LAHLOU, Salem, et al. GFlowNets and variational inference. **International Conference on Learning Representations (ICLR)**, 2023.

MANKOWITZ, Daniel J.; MANN, Timothy A.; MANNOR, Shie. **Adaptive Skills, Adaptive Partitions (ASAP)**. [S.l.: s.n.], 2016. arXiv: 1602.03351 [cs.LG]. Available from: <https://arxiv.org/abs/1602.03351>.

MARTHI, Bhaskara. Automatic shaping and decomposition of reward functions. In: PROCEEDINGS of the 24th International Conference on Machine learning. [S.l.: s.n.], 2007. P. 601–608.

MAURER, Andreas. **A Note on the PAC Bayesian Theorem**. [S.l.: s.n.], 2004. arXiv: cs/0411099 [cs.LG]. Available from: <https://arxiv.org/abs/cs/0411099>.

MCALLESTER, David. **A PAC-Bayesian Tutorial with A Dropout Bound**. [S.l.: s.n.], 2013. arXiv: 1307.2118 [cs.LG]. Available from: <https://arxiv.org/abs/1307.2118>.

MCALLESTER, David A. PAC-Bayesian model averaging. In: PROCEEDINGS of the twelfth annual conference on Computational Learning Theory. [S.l.: s.n.], 1999. P. 164–170.

________. Some pac-bayesian theorems. In: PROCEEDINGS of the eleventh annual conference on Computational Learning Theory. [S.l.: s.n.], 1998. P. 230–234.

MCCLOSKEY, Michael; COHEN, Neal J. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In: BOWER, Gordon H. (Ed.). [S.l.]: Academic Press, 1989. v. 24. (Psychology of Learning and Motivation). P. 109–165. DOI: https://doi.org/10.1016/S0079-7421(08)60536-8.

MCDIARMID, Colin. Concentration. In: PROBABILISTIC methods for algorithmic discrete mathematics. [S.l.]: Springer, 1998. P. 195–248.

MCMAHAN, Brendan et al. Communication-efficient learning of deep networks from decentralized data. In: PMLR. ARTIFICIAL intelligence and statistics. [S.l.: s.n.], 2017. P. 1273–1282.

MESQUITA, D.; BLOMSTEDT, P.; KASKI, S. Embarrassingly parallel MCMC using deep invertible transformations. In: UAI. [S.l.: s.n.], 2019.

MINKA, Thomas. **Divergence measures and message passing**. Ed. by Microsoft Research. [S.l.], 2005. Available from: <https://www.seas.harvard.edu/courses/cs281/papers/minka-divergence.pdf>.

MNIH, Andriy; REZENDE, Danilo. Variational inference for monte carlo objectives. In: PMLR. INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2016. P. 2188–2196.

MOHAMED, Shakir et al. Monte Carlo Gradient Estimation in Machine Learning. **J. Mach. Learn. Res.**, v. 21, 132:1–132:62, 2020.

MORRIS, Christopher et al. **Weisfeiler and Leman Go Neural: Higher-order Graph Neural Networks**. [S.l.: s.n.], 2021. arXiv: 1810.02244 [cs.LG]. Available from: <https://arxiv.org/abs/1810.02244>.

NEAL, Radford M et al. MCMC using Hamiltonian dynamics. **Handbook of markov chain monte carlo**, Chapman and Hall/CRC, 2011.

NEISWANGER, W.; WANG, C.; XING, E. P. Asymptotically Exact, Embarrassingly Parallel MCMC. In: UAI. [S.l.: s.n.], 2014.

NEMETH, C.; SHERLOCK, C. Merging MCMC Subposteriors through Gaussian-Process Approximations. **Bayesian Analysis**, v. 13, n. 2, 2018.

NG, Andrew Y; HARADA, Daishi; RUSSELL, Stuart. Policy invariance under reward transformations: Theory and application to reward shaping. In: ICML. [S.l.: s.n.], 1999. v. 99, p. 278–287.

NG, Ignavier; ZHANG, Kun. Towards Federated Bayesian Network Structure Learning with Continuous Optimization. In: INTERNATIONAL Conference on Artificial Intelligence and Statistics. [S.l.: s.n.], 2022.

NICA, Andrei Cristian et al. Evaluating generalization in gflownets for molecule design. In: ICLR2022 Machine Learning for Drug Discovery. [S.l.: s.n.], 2022.

OWEN, Art B. **Monte Carlo theory, methods and examples**. [S.l.: s.n.], 2013.

PAN, Ling; MALKIN, Nikolay, et al. Better Training of GFlowNets with Local Credit and Incomplete Trajectories. In: INTERNATIONAL Conference on Machine Learning (ICML). [S.l.: s.n.], 2023.

________. Better training of gflownets with local credit and incomplete trajectories. **arXiv preprint arXiv:2302.01687**, 2023.

PAN, Ling; ZHANG, Dinghuai, et al. Generative Augmented Flow Networks. In: INTERNATIONAL Conference on Learning Representations (ICLR). [S.l.: s.n.], 2023.

PANDEY, Mohit; SUBBARAJ, Gopeshh; BENGIO, Emmanuel. GFlowNet Pretraining with Inexpensive Rewards. **arXiv preprint arXiv:2409.09702**, 2024.

________.________. [S.l.: s.n.], 2024. arXiv: 2409.09702 [cs.LG]. Available from: <https://arxiv.org/abs/2409.09702>.

PAPINI, Matteo et al. Stochastic Variance-Reduced Policy Gradient. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2018.

PASZKE, Adam et al. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. [S.l.: s.n.], 2019. arXiv: 1912.01703 [cs.LG].

PÉREZ-ORTIZ, María et al. Tighter risk certificates for neural networks. **Journal of Machine Learning Research**, v. 22, n. 227, p. 1–40, 2021. Available from: <http://jmlr.org/papers/v22/20-879.html>.

RAFAILOV, Rafael et al. **Direct Preference Optimization: Your Language Model is Secretly a Reward Model**. [S.l.: s.n.], 2024. arXiv: 2305.18290 [cs.LG]. Available from: <https://arxiv.org/abs/2305.18290>.

RANGANATH, Rajesh; GERRISH, Sean; BLEI, David M. Black Box Variational Inference. In: AISTATS. [S.l.]: JMLR.org, 2014. v. 33. (JMLR Workshop and Conference Proceedings), p. 814–822.

RECTOR-BROOKS, Jarrid et al. Thompson sampling for improved exploration in GFlowNets. **arXiv preprint arXiv:2306.17693**, 2023.

________.________. [S.l.: s.n.], 2023. arXiv: 2306.17693 [cs.LG]. Available from: <https://arxiv.org/abs/2306.17693>.

RÉNYI, Alfréd. On measures of entropy and information. In: UNIVERSITY OF CALIFORNIA PRESS. PROCEEDINGS of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. [S.l.: s.n.], 1961.

REZENDE, Danilo; MOHAMED, Shakir. Variational inference with normalizing flows. In: PMLR. INTERNATIONAL conference on machine learning. [S.l.: s.n.], 2015.

RHODES, Benjamin; GUTMANN, Michael U. Variational noise-contrastive estimation. In: AISTATS. [S.l.: s.n.], 2019.

RICHTER, Lorenz et al. VarGrad: A Low-Variance Gradient Estimator for Variational Inference, 2020.

RICHTER, Lorenz et al. VarGrad: A Low-Variance Gradient Estimator for Variational Inference. In: ADVANCES in Neural Information Processing Systems (NeurIPS). [S.l.: s.n.], 2020.

RODRÍGUEZ-GÁLVEZ, Borja; THOBABEN, Ragnar; SKOGLUND, Mikael. **More PAC-Bayes bounds: From bounded losses, to losses with general tail behaviors, to anytime validity**. [S.l.: s.n.], 2024. arXiv: 2306.12214 [stat.ML]. Available from: <https://arxiv.org/abs/2306.12214>.

ROY, Julien et al. **Goal-conditioned GFlowNets for Controllable Multi-Objective Molecular Design**. [S.l.: s.n.], 2023. arXiv: 2306.04620 [cs.LG]. Available from: <https://arxiv.org/abs/2306.04620>.

ROY, Vivekananda. Convergence diagnostics for markov chain monte carlo. **Annual Review of Statistics and Its Application**, Annual Reviews, v. 7, n. 1, p. 387–412, 2020.

ROYCHOUDHURY, Arindam; WILLIS, Amy; BUNGE, John. Consistency of a phylogenetic tree maximum likelihood estimator. **Journal of Statistical Planning and Inference**, 2015.

RUIZ, Francisco; TITSIAS, Michalis. A contrastive divergence for combining variational inference and mcmc. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2019.

SALIMANS, Tim; KNOWLES, David A. **On Using Control Variates with Stochastic Approximation for Variational Bayes and its Connection to Stochastic Linear Regression**. [S.l.: s.n.], 2014. arXiv: 1401.1022 [stat.CO].

SCHAEFFER, Rylan et al. Streaming Inference for Infinite Feature Models. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2022.

SCHERVISH, Mark J. **Theory of statistics**. [S.l.]: Springer Science & Business Media, 2012.

SCHULMAN, John et al. Proximal policy optimization algorithms. **arXiv preprint arXiv:1707.06347**, 2017.

SEFIDGARAN, Milad; CHOR, Romain; ZAIDI, Abdellatif. Rate-Distortion Theoretic Bounds on Generalization Error for Distributed Learning. In________. **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2022.

SELDIN, Yevgeny et al. **PAC-Bayesian Inequalities for Martingales**. [S.l.: s.n.], 2012. arXiv: 1110.6886 [cs.LG]. Available from: <https://arxiv.org/abs/1110.6886>.

SHALEV-SHWARTZ, Shai; BEN-DAVID, Shai. **Understanding machine learning: From theory to algorithms**. [S.l.]: Cambridge university press, 2014.

SHEN, Max W. et al. Towards Understanding and Improving GFlowNet Training. In: INTERNATIONAL Conference on Machine Learning. [S.l.: s.n.], 2023.

SHI, Jiaxin et al. Gradient estimation with discrete Stein operators. **NeurIPS**, v. 35, 2022.

SIEGELMANN, Hava T; SONTAG, Eduardo D. On the computational power of neural nets. In: PROCEEDINGS of the fifth annual workshop on Computational learning theory. [S.l.: s.n.], 1992. P. 440–449.

SILVA, Tiago da et al. **Embarrassingly Parallel GFlowNets**. [S.l.: s.n.], 2024. arXiv: 2406.03288 [cs.LG]. Available from: <https://arxiv.org/abs/2406.03288>.

SOUZA, D. de et al. Parallel MCMC without embarrassing failures. In: AISTATS. [S.l.: s.n.], 2022.

SUTTON, Richard. The bitter lesson. **Incomplete Ideas (blog)**, v. 13, n. 1, p. 38, 2019.

TIAPKIN, Daniil et al. **Generative Flow Networks as Entropy-Regularized RL**. [S.l.: s.n.], 2024. arXiv: 2310.12934 [cs.LG].

________.________. [S.l.: s.n.], 2024. arXiv: 2310.12934 [cs.LG]. Available from: <https://arxiv.org/abs/2310.12934>.

TRABUCCO, Brandon et al. **Design-Bench: Benchmarks for Data-Driven Offline Model-Based Optimization**. [S.l.: s.n.], 2022. arXiv: 2202.08450 [cs.LG]. Available from: <https://arxiv.org/abs/2202.08450>.

TSALLIS, Constantino. Possible generalization of Boltzmann-Gibbs statistics. **Journal of statistical physics**, Springer, v. 52, p. 479–487, 1988.

VAPNIK, Vladimir. **Statistical Learning Theory**. [S.l.]: John Wiley & Sons, 1998.

VAPNIK, Vladimir N; CHERVONENKIS, A Ya. On the uniform convergence of relative frequencies of events to their probabilities. In: MEASURES of complexity: festschrift for alexey chervonenkis. [S.l.]: Springer, 2015. P. 11–30.

VAPNIK, Vladimir N. **The Nature of Statistical Learning Theory**. [S.l.]: Springer New York, 2000. ISBN 9781475732641. DOI: 10.1007/978-1-4757-3264-1. Available from: <http://dx.doi.org/10.1007/978-1-4757-3264-1>.

VELIČKOVIĆ, Petar et al. Graph attention networks. **arXiv preprint arXiv:1710.10903**, 2017.

VEMGAL, Nikhil; LAU, Elaine; PRECUP, Doina. **An Empirical Study of the Effectiveness of Using a Replay Buffer on Mode Discovery in GFlowNets**. [S.l.: s.n.], 2023. arXiv: 2307.07674 [cs.LG]. Available from: <https://arxiv.org/abs/2307.07674>.

WAINWRIGHT, Martin J.; JORDAN, Michael I. Graphical Models, Exponential Families, and Variational Inference. **Found. Trends Mach. Learn.**, v. 1, n. 1-2, p. 1–305, 2008.

WALKER, A M. On the asymptotic behaviour of posterior distributions. en. **J. R. Stat. Soc.**, Wiley, v. 31, n. 1, p. 80–88, Jan. 1969.

WANG, Qing et al. $\mathscr{N}$-WL: A New Hierarchy of Expressivity for Graph Neural Networks. In: THE Eleventh International Conference on Learning Representations. [S.l.: s.n.], 2023. Available from: <https://openreview.net/forum?id=5cAI0qXxyv>.

WANG, Xi; GEFFNER, Tomas; DOMKE, Justin. Joint control variate for faster black-box variational inference. In: AISTATS. [S.l.]: PMLR, 2024. v. 238. (Proceedings of Machine Learning Research), p. 1639–1647.

WANG, Xiangyu et al. Parallelizing MCMC with Random Partition Trees. In: NEURIPS (NeurIPS). [S.l.: s.n.], 2015.

WANG, Zhe; VELIČKOVIĆ, Petar, et al. TacticAI: an AI assistant for football tactics. **Nature communications**, Nature Publishing Group UK London, v. 15, n. 1, p. 1906, 2024.

WEAVER, Lex; TAO, Nigel. The optimal reward baseline for gradient-based reinforcement learning. **arXiv preprint arXiv:1301.2315**, 2013.

WEISFEILER, B.; LEHMAN, A. A. A Reduction of a Graph to a Canonical Form and an Algebra Arising during This Reduction. **Nauchno-Technicheskaya Informatsia**, v. 2, n. 9, p. 12–16, 1968.

WILLIAMS, David. **Probability with Martingales.** [S.l.]: Cambridge University Press, 1991.

WILLIAMS, Ronald J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. **Machine Learning**, Springer Science and Business Media LLC, 1992.

WOLPERT, David H; MACREADY, William G. No free lunch theorems for optimization. **IEEE transactions on evolutionary computation**, IEEE, v. 1, n. 1, p. 67–82, 1997.

XU, Keyulu; HU, Weihua, et al. How powerful are graph neural networks? **International Conference on Learning Representations (ICLR)**, 2019.

XU, Pan; GAO, Felicia; GU, Quanquan. An Improved Convergence Analysis of Stochastic Variance-Reduced Policy Gradient. In: UAI. [S.l.: s.n.], 2020.

YAGLI, Semih; DYTSO, Alex; POOR, H. Vincent. **Information-Theoretic Bounds on the Generalization Error and Privacy Leakage in Federated Learning**. [S.l.: s.n.], 2020. arXiv: 2005.02503 [cs.LG]. Available from: <https://arxiv.org/abs/2005.02503>.

YANG, Ziheng. **Molecular Evolution: A Statistical Approach**. [S.l.]: Oxford University PressOxford, May 2014. ISBN 9780191782251. DOI: 10.1093/acprof:oso/9780199602605.001.0001.

YANG, Ziheng. **Molecular evolution: a statistical approach**. [S.l.]: Oxford University Press, 2014.

YIN, Mingzhang; ZHOU, Mingyuan. Semi-implicit variational inference. In: PMLR. INTERNATIONAL conference on machine learning. [S.l.: s.n.], 2018.

YU, Kui et al. Causal discovery from streaming features. In: IEEE. 2010 IEEE International Conference on Data Mining. [S.l.: s.n.], 2010. P. 1163–1168.

ZAHEER, M. et al. Deep Sets. In: ADVANCES in Neural Information Processing Systems (NeurIPS). [S.l.: s.n.], 2017.

ZHANG, Bohang et al. Rethinking the Expressive Power of GNNs via Graph Biconnectivity. In: THE Eleventh International Conference on Learning Representations. [S.l.: s.n.], 2023. Available from: <https://openreview.net/forum?id=r9hNv76KoT3>.

ZHANG, Cheng; MATSEN IV, Frederick A. Variational Bayesian phylogenetic inference. In: INTERNATIONAL Conference on Learning Representations (ICLR). [S.l.: s.n.], 2018.

ZHANG, David W et al. Robust scheduling with GFlowNets. In: INTERNATIONAL Conference on Learning Representations (ICLR). [S.l.: s.n.], 2023.

ZHANG, Dinghuai; CHEN, Ricky Tian Qi; LIU, Cheng-Hao, et al. **Diffusion Generative Flow Samplers: Improving learning signals through partial trajectory optimization**. [S.l.: s.n.], 2023. arXiv: 2310.02679 [cs.LG].

ZHANG, Dinghuai; CHEN, Ricky TQ; MALKIN, Nikolay, et al. Unifying Generative Models with GFlowNets and Beyond. **ICML Beyond Bayes workshop**, 2022.

ZHANG, Dinghuai; DAI, Hanjun, et al. Let the Flows Tell: Solving Graph Combinatorial Optimization Problems with GFlowNets. In: NEURIPS (NeurIPS). [S.l.: s.n.], 2023.

ZHANG, Dinghuai; MALKIN, Nikolay, et al. Generative flow networks for discrete probabilistic modeling. In: INTERNATIONAL Conference on Machine Learning (ICML). [S.l.: s.n.], 2022.

ZHAO, Yuan et al. Streaming Variational Monte Carlo. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2023.

ZHOU, Ming Yang et al. PhyloGFN: Phylogenetic inference with generative flow networks. In: THE Twelfth International Conference on Learning Representations. [S.l.: s.n.], 2024.

ZHU, Yiheng et al. **Sample-efficient Multi-objective Molecular Optimization with GFlowNets**. [S.l.: s.n.], 2023. arXiv: 2302.04040 [cs.LG].

Appendix

APPENDIX A – Additional background

A.1 1-Weisfeiler-Leman isomorphism test

Here, we denote a graph as a tuple $G = (V, E)$, where $V = \{1, 2, \ldots, n\}$ is the set of nodes and $E \subseteq V \times V$ is the set of edges. More specifically, we consider attributed graphs — i.e., each node $v \in V$ has associated features $x_v \in \mathbb{R}^d$. Also, we denote the set of neighbors of a node v in the graph as $\mathcal{N}_v = \{u : (u, v) \in E\}$.

1-WL test. The Weisfeiler-Lehman isomorphism test (WEISFEILER; LEHMAN, 1968) assigns colors for all nodes in an attributed input graph G by applying the following iterative procedure:

Initialization: The colors of all nodes in G are initialized using the initial node features: $\forall v \in V, c^0(v) = x_v$. If node features are not available, all nodes receive identical colors;

Refinement: At step ℓ, the colors of all nodes are refined using a hash (injective) function: for all $v \in V$, we apply $c^{\ell+1}(v) = \textsc{Hash}(c^\ell(v), \{\!\{c^\ell(u) : (u, v) \in E\}\!\})$;

Termination: The test is carried out for two graphs in parallel and stops when the multisets of corresponding colors diverge, returning non-isomorphic. If the algorithm runs until the number of different colors stops increasing, the test is deemed inconclusive.

A.2 FL- and LED-GFlowNets

As mentioned earlier, both the forward-looking (FL-) and Learning Energy Decompositions (LED-) GFlowNets are built upon the principle of enhancing credit assignment via reparamaterizing the flow function F as a logarithmic residual of a (either hand-crafted or learned) basic potential function ϕ, i.e., $\log F(s, s') = \log \phi(s, s') + \log \tilde{F}(s, s')$, and $\tilde{F}(s, s') = \tilde{F}(s)p_F(s'|s)$ in the usual policy-based parameterization. Under this novel perspective, the DB loss becomes

$$\mathcal{L}_{\mathrm{DB}}(p_F, p_B, F) = \mathbb{E}_\tau \left[\frac{1}{\#\tau} \sum_{(s,s') \in \tau} \left(\phi(s, s') + \log \frac{\tilde{F}(s)p_F(s'|s)}{\tilde{F}(s')p_B(s|s')} \right)^2 \right], \tag{A.1}$$

with the constraint that $\sum_{(s,s') \in \tau} \phi(s, s') = \log R(x)$, in which x is the unique terminal state in the trajectory τ. Recall that, for FL-GFlowNets, $\phi(s, s') = \xi(s') - \xi(s)$ for a hand-crafted energy function ξ such that $\xi(s_o) = 0$ and $\xi(x) = \log R(x)$ (PAN; MALKIN, et al., 2023a, Assumption 1). For LED-GFlowNets, $\phi(s, s')$ is parameterized as an neural network taking as input a concatenation of the vectorial representations of s and s'. The parameters of ϕ are then learned by minimizing

$$\mathcal{L}_{\text{LS}}(\tau) = \mathbb{E}_{(m_{s,s'})_{(s,s')\in\tau} \sim \text{Bernoulli}(1-\gamma)} \left[\left(\frac{1}{\#\tau} \xi(x) - \frac{1}{C} \sum_{(s,s')\in\tau} m_{s,s'} \phi_\theta(s,s') \right)^2 \right], \qquad \text{(A.2)}$$

in which $\{m_{s,s'}\}_{(s,s')}$ is a Dropout mask and $C = \sum_{(s,s')\in\tau} m_{s,s'}$ is the number of unmaked transitions. During training, we interleave gradient-based updates of the potential function ϕ and (p_F, p_B, F) until a chosen stopping criterion (e.g., maximum number of epochs) is satisfied.

APPENDIX B – Proofs

Proofs for Chapter 3

We will consider the measurable space of *trajectories* $(\mathcal{P}_{\mathcal{S}}, \Sigma_P)$, with $\mathcal{P}_{\mathcal{S}} = \{(s, s_1, \ldots, s_n, s_f) \in \mathcal{S}^{n+1} \times \{s_f\} \colon 0 \leq n \leq N-1\}$ and Σ_P as the σ-algebra generated by $\bigcup_{n=1}^{N+1} \Sigma^{\otimes n}$. For notational convenience, we use the same letters for representing the measures and kernels of $(\mathcal{S}, \Sigma)$ and their natural product counterparts in $(\mathcal{P}_{\mathcal{S}}, \Sigma_P)$, which exist by Carathéodory extension's theorem (WILLIAMS, D., 1991); for example, $\nu(B) = \nu^{\otimes n}(B)$ for $B = (B_1, \ldots, B_n) \in \Sigma^{\otimes n}$ and $p_{F_\theta}(\tau | s_o; \theta)$ is the density of $P_F^{\otimes n+1}(s_o, \cdot)$ for $\tau = (s_o, s_1, \ldots, s_n, s_f)$ relatively to $\mu^{\otimes n}$. In this case, we will write τ for a generic element of $\mathcal{P}_{\mathcal{S}}$ and x for its terminal state (which is unique by Definition 3.1.1).

Proof of Proposition 3.2.1

We will show that the gradient of the expected on-policy TB loss matches the gradient of the KL divergence between the forward and backward policies. Firstly, note that

$$\begin{aligned}
\nabla_\theta \mathrm{KL}[P_F || P_B] &= \nabla_\theta \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\log \frac{p_F(\tau | s_o; \theta)}{p_B(\tau)} \right] \\
&= \nabla_\theta \int_\tau \log \frac{p_F(\tau | s_o; \theta)}{p_B(\tau)} \mathrm{d}P_F(s_o, \mathrm{d}\tau) \\
&= \nabla_\theta \int_\tau \log \frac{p_F(\tau | s_o; \theta)}{p_B(\tau)} p_F(\tau | s_o; \theta) \mathrm{d}\kappa_f(s_o, \mathrm{d}\tau) \\
&= \int_\tau \nabla_\theta \log \frac{p_F(\tau | s_o; \theta)}{p_B(\tau)} P_F(s_o, \mathrm{d}\tau) \\
&\quad + \int_\tau \log \frac{p_F(\tau | s_o; \theta)}{p_B(\tau)} \nabla_\theta p_F(\tau | s_o; \theta) \mathrm{d}\kappa_f(s_o, \mathrm{d}\tau)
\end{aligned}$$

by Leibniz's rule for integrals and the product rule for derivatives. Then, since $\nabla_\theta f(\theta) = f(\theta) \nabla \log f(\theta)$ for any differentiable function $f \colon \theta \mapsto f(\theta)$,

$$\begin{aligned}
&\nabla_\theta \mathrm{KL}\left[P_F || P_B\right] \\
&= \mathop{\mathbb{E}}_{\tau \sim P_F(s_o, \cdot)} \left[\nabla_\theta \log p_F(\tau | s_o) + \log \frac{p_F(\tau | s_o)}{p_B(\tau)} \nabla_\theta \log p_F(\tau | s_o) \right] \\
&= \mathop{\mathbb{E}}_{\tau \sim P_F(s_o, \cdot)} \left[\log \frac{p_F(\tau | s_o)}{p_B(\tau)} \nabla_\theta \log p_F(\tau | s_o) \right];
\end{aligned} \tag{B.1}$$

we omitted the dependency of P_F (and of p_F thereof) on the parameters θ for conciseness. On the other hand,

$$\nabla_\theta \mathcal{L}_{TB}(\tau; \theta) = 2 \left(\log \frac{p_F(\tau | s_o; \theta)}{p_B(\tau)} \right) \nabla_\theta \log p_F(\tau) \tag{B.2}$$

by the chain rule for derivatives. Thus,

$$\mathbb{E}_{\tau\sim P_F(s_o,\cdot)}\nabla_\theta\mathcal{L}_{TB}(\tau;\theta) = 2\nabla_\theta\mathrm{KL}[P_F||P_B], \tag{B.3}$$

ensuring that the equivalence between $\mathcal{L}_{TB}$ and KL in terms of expected gradients holds in a context broader than that of finitely supported distributions (MALKIN; LAHLOU, et al., 2023).

Proof of Lemma 3.3.2

Henceforth, we will recurrently refer to the score estimator for gradients of expectations (WILLIAMS, R. J., 1992), namely,

$$\nabla_\theta \underset{\tau\sim P_F(s_o,\cdot)}{\mathbb{E}}[f_\theta(\tau)] = \underset{\tau\sim P_F(s_o,\cdot)}{\mathbb{E}}\left[\nabla_\theta f_\theta(\tau) + f_\theta(\tau)\nabla_\theta \log p_F(\tau|s_o;\theta)\right], \tag{B.4}$$

which can be derived using the arguments of the preceding section. In this context, the Renyi-α's divergence satisfies

$$\nabla_\theta R_\alpha(P_F||P_B) = \frac{\nabla_\theta\mathbb{E}_{\tau\sim P_F(s_o,\cdot)}[g(\tau,\theta)]}{(\alpha-1)\mathbb{E}_{\tau\sim P_F(s_o,\cdot)}g(\tau,\theta)},$$

with $g(\tau;\theta) = \left({}^{p_B(\tau|x)r(x)}/_{p_F(\tau|s_o;\theta)}\right)^{1-\alpha}$ and $\alpha \neq 1$; similarly, the Tsallis-α's divergence abides by

$$\nabla_\theta T_\alpha(P_F||P_B) = \frac{1}{(\alpha-1)}\nabla_\theta\mathbb{E}_{\tau\sim P_F(s_o,\cdot)}[g(\tau,\theta)]. \tag{B.5}$$

The statement then follows by substituting $\nabla_\theta\mathbb{E}_{\tau\sim P_F(s_o,\cdot)}[g(\tau,\theta)]$ with the corresponding score estimator given by Equation (B.4).

Proof of Lemma 3.3.1

Forward KL divergence. The gradient of $\mathrm{KL}[P_B||P_F]$ is straightforwardly obtained through the application of Leibniz's rule for integrals,

$$\nabla_\theta\mathrm{KL}[P_B||P_F] = -\mathbb{E}_{\tau\sim P_B(s_f,\cdot)}\left[\nabla_\theta\log p_F(\tau|s_o;\theta)\right],$$

since the averaging distribution P_B do not depend on the varying parameters θ. However, as we compute Monte Carlo averages over samples of P_F, we apply an importance reweighting scheme (OWEN, 2013, Chapter 9) to the previous expectation to infer that, up to a positive multiplicative constant,

$$\nabla_\theta\mathrm{KL}[P_B||P_F] \overset{C}{=} -\mathbb{E}_{\tau\sim P_F(s_o,\cdot)}\left[\frac{p_B(\tau|x)r(x)}{p_F(\tau|s_o;\theta)}\nabla_\theta\log p_F(\tau|s_o;\theta)\right],$$

with $\overset{C}{=}$ denoting equality up to a positive multiplicative constant. We emphasize that most modern stochastic gradient methods for optimization, such as Adam (KINGMA; BA,

2014) and RMSProp (HINTON; SRIVASTAVA; SWERSKY, 2012), remain unchanged when we multiply the estimated gradients by a fixed quantity; thus, we may harmlessly compute gradients up to multiplicative constants.

Reverse KL divergence. We verified in Equation (B.1) that

$$\nabla_\theta \mathrm{KL}[P_F || P_B] = \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\log \frac{p_F(\tau | s_o)}{p_B(\tau)} \nabla_\theta s_\theta(\tau) \right].$$

Since $p_B(\tau) = p_B(\tau|x)^{r(x)}/z$ and $\mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \nabla_\theta s_\theta(\tau) = 0$, the quantity $\nabla_\theta \mathrm{KL}[P_F || P_B]$ may be rewritten as

$$\begin{aligned}\nabla_\theta \mathrm{KL}[P_F || P_B] = \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} & \left[\log \frac{p_F(\tau | s_o)}{p_B(\tau)} \nabla_\theta s_\theta(\tau) \right] \\ & + \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[(\log Z) \nabla_\theta s_\theta(\tau) \right] \\ & = \mathbb{E}_{\tau \sim P_F(s_o, \cdot)} \left[\log \frac{p_F(\tau | s_o)}{p_B(\tau | x) r(x)} \nabla_\theta s_\theta(\tau) \right],\end{aligned}$$

in which x is the terminal state corresponding to the trajectory τ. Thus proving the statement in Lemma 3.3.1.

Proof of Proposition 3.3.1

We will derive an expression for the optimal baseline of a vector-valued control variate. For this, let f be the averaged function and $g \colon \tau \mapsto g(\tau)$ be the control variate. Assume, without loss of generality, that $\mathbb{E}_\pi[g] = 0$ for the averaging distribution π over the space of trajectories. In this case, the optimal baseline for the control variate $a^\star$ is found by

$$a^\star = \arg\min_{a \in \mathbb{R}} \mathrm{Tr} \left(\mathrm{Cov}_{\tau \sim \pi}[f(\tau) - a \cdot g(\tau)] \right). \tag{B.6}$$

Thus,

$$a^\star = \arg\min_{a \in \mathbb{R}} \mathrm{Tr} \Bigg(- 2a \cdot \mathrm{Cov}_\pi[f(\tau), g(\tau)] + a^2 \mathrm{Cov}_\pi(g(\tau)) \Bigg),$$

which is a convex optimization problem solved by

$$\begin{aligned} a^\star &= \frac{\mathrm{Tr}\left(\mathrm{Cov}_\pi[f(\tau), g(\tau)]\right)}{\mathrm{Tr}\left(\mathrm{Cov}_\pi[g(\tau)]\right)} \\ &= \frac{\mathrm{Tr}\ \mathbb{E}_\pi[(f - \mathbb{E}_\pi[f(\tau)]) g(\tau)^T]}{\mathrm{Tr}\ \mathbb{E}_\pi[g(\tau) g(\tau)^T]} \\ &= \frac{\mathbb{E}_\pi[\mathrm{Tr}\ (f - \mathbb{E}_\pi[f(\tau)])^T g(\tau)]}{\mathbb{E}_\pi[\mathrm{Tr}\ g(\tau)^T g(\tau)]} \\ &= \frac{\mathbb{E}_\pi[(f - \mathbb{E}_\pi[f(\tau)])^T g(\tau)]}{\mathbb{E}_\pi[g(\tau)^T g(\tau)]}, \end{aligned} \tag{B.7}$$

in which we used the circular property of the trace. This equation exactly matches the result in Proposition 3.3.1. In practice, we use $g(\tau) = \nabla_\theta \log p_F(\tau)$ for both the reverse KL- and α-divergences, rendering a baseline $a^\star$ that depends non-linearly on the sample gradients and is hence difficult to compute in a GPU-powered autodiff framework efficiently. We thus use Equation (3.7) to estimate $a^\star$.

Proofs for Chapter 4

Proof of Proposition 4.1.1

We are assuming that $p_\top^{(t)} \propto \pi_t$ and that $\mathbb{E}_{\tau \sim \xi}[\mathcal{L}_{SB}(\tau)] = 0$ for a distribution ξ of full support. Thus, $\mathcal{L}_{SB}(\tau) = 0$ for all τ. As a consequence,

$$p_F^{(t+1)}(\tau) = \left(\frac{p_B^{(t+1)}(\tau|x)}{p_B^{(t)}(\tau|x)}\right) \cdot \frac{Z_t}{Z_{t+1}} \cdot p_F^{(t)}(\tau) f(\mathcal{D}_{t+1}|x). \tag{B.8}$$

By assumption, $(p_F^{(t)}, p_B^{(t)}, Z_t)$ satisfy the trajectory balance condition with respect to π_t, therefore, $Z_t \cdot p_F^{(t)}(\tau) = p_B^{(t)}(\tau|x)\pi_t(x)$. Thus,

$$\frac{p_F^{(t+1)}(\tau)}{p_B^{(t+1)}(\tau|x)} = Z_{t+1}\pi_t(x) f(\mathcal{D}_{t+1}|x). \tag{B.9}$$

Finally, by summing over $\tau \rightsquigarrow x$:

$$\begin{aligned} p_\top^{(t+1)}(x) &\coloneqq \mathbb{E}_{\tau \sim p_B^{(t+1)}(\cdot|x)}\left[\frac{p_F^{(t+1)}(\tau)}{p_B^{(t+1)}(\tau|x)}\right] \propto \pi_t(x) f(\mathcal{D}_{t+1}|x) && \text{(B.10)} \\ &\propto \pi_{t+1}(x). && \text{(B.11)} \end{aligned}$$

Proof of Proposition 4.3.1

Firstly, we note that $\delta_{LS}^{\xi}(p, q)$ is a metric. Thus, by the triangle inequality,

$$\delta_{LS}^{\pi_{t+1}}\left(p_\top^{(t+1)}, \pi_{t+1}\right) \leq \delta_{LS}^{\pi_{t+1}}\left(p_\top^{(t+1)}, \hat{p}_\top^{(t+1)}\right) + \delta_{LS}^{\pi_{t+1}}\left(\hat{p}_\top^{(t+1)}, \pi_{t+1}\right). \tag{B.12}$$

The first term of the right-hand-side of the preceding equation corresponds to the estimation error associated to the GFlowNet's learning problem. For the second term, note that $\hat{p}_\top^{(t+1)}(x) = \frac{Z_t}{\hat{Z}_{t+1}} p_\top^{(t)}(x) f(\mathcal{D}_{t+1}|x)$ by assumption ($\hat{p}_\top^{(t+1)}$ satisfies the SB condition). Thus,

$$\begin{aligned} \log \hat{p}_\top^{(t+1)}(x) - \log \pi_{t+1}(x) &= \log \frac{Z_t}{\hat{Z}_{t+1}} + \log p_\top^{(t)}(x) f(\mathcal{D}_{t+1}|x) - \log \frac{Z_t^\star \pi_t(x)}{Z_{t+1}^\star} f(\mathcal{D}_{t+1}|x) \\ &= \log \frac{Z_{t+1}^\star}{\hat{Z}_{t+1}} + \log \frac{Z_t}{Z_t^\star} + \log \frac{p_\top^{(t)}(x)}{\pi_t(x)}. \end{aligned}$$

The result follows by a further application of the triangle inequality to $\delta_{LS}^{\pi_{t+1}}\left(p_\top^{(t+1),\star}, \pi_{t+1}\right)$, namely,

$$\begin{aligned} \delta_{LS}^{\pi_{t+1}}\left(\hat{p}_\top^{(t+1)}, \pi_{t+1}\right) &= \mathbb{E}_{x \sim \pi_{t+1}}\left[\left(\log \frac{Z_{t+1}^\star}{\hat{Z}_{t+1}} + \log \frac{Z_t}{Z_t^\star} + \log \frac{p_\top^{(t)}(x)}{\pi_t(x)}\right)^2\right]^{1/2} \\ &\leq \left|\log \frac{\hat{Z}_{t+1}}{Z_{t+1}^\star}\right| + \left|\log \frac{Z_t}{Z_t^\star}\right| + \delta_{LS}^{\pi_{t+1}}\left(p_\top^{(t)}, \pi_t\right). \end{aligned} \tag{B.13}$$

Proposition 4.3.1 is obtained by plugging Equation (B.13) into Equation (B.12).

Proof of Proposition 4.3.2

This result, which follows from reasoning similar to the one in Proposition 4.3.1 above, aims to show the dependence of the model's performance on the newly observed dataset. In this sense, notice that

$$TV\left(p_{\top}^{(t+1)}, \pi_{t+1}\right) \leq TV\left(p_{\top}^{(t+1)}, \hat{p}_{\top}^{(t+1)}\right) + TV\left(\hat{p}_{\top}^{(t+1)}, \pi_{t+1}\right). \tag{B.14}$$

Thus, since $\hat{p}_{\top}^{(t+1)}(x) = \frac{Z_t}{\hat{Z}_{t+1}} p_{\top}^{(t)}(x) f(\mathcal{D}_{t+1}|x)$,

$$\begin{aligned} TV\left(\hat{p}_{\top}^{(t+1)}, \pi_t\right) &= \frac{1}{2}\sum_{x\in\mathcal{X}} \left|p_{\top}^{(t+1)}(x) - \pi_{t+1}(x)\right| \\ &= \frac{1}{2}\sum_{x\in\mathcal{X}} f(\mathcal{D}_{t+1}|x)\left|\frac{Z_t}{\hat{Z}_{t+1}} p_{\top}^{(t)}(x) - \frac{Z_t^{\star}}{Z_{t+1}^{\star}}\pi_t(x)\right| \\ &\leq \frac{1}{2} f(\mathcal{D}_{t+1}|\hat{x}) \sum_{x\in\mathcal{X}} \left|\frac{Z_t}{\hat{Z}_{t+1}} p_{\top}^{(t)}(x) - \frac{Z_t^{\star}}{Z_{t+1}^{\star}}\pi_t(x)\right|. \end{aligned} \tag{B.15}$$

The result follows by plugging Equation (B.15) into Equation (B.14).

Proof of Proposition 4.3.3

This result follows from the successive application of the triangle, Pinsker‘s (CSISZÁR; KÖRNER, 2011) and Jensen's inequality applied to the KL divergence. More specifically, first note that the optimal distribution under the KL streaming criterion satisfies $\hat{p}_{\top}^{(t+1)} \propto p_{\top}^{(t)} f(\mathcal{D}_{t+1}|x)$. Then, by the triangle inequality,

$$TV\left(p_{\top}^{(t+1)}, \pi_{t+1}\right) \leq TV\left(p_{\top}^{(t+1)}, \hat{p}_{\top}^{(t+1)}\right) + TV\left(\hat{p}_{\top}^{(t+1)}, \pi_{t+1}\right). \tag{B.16}$$

For the first term, note that

$$TV\left(p_{\top}^{(t+1)}, \hat{p}_{\top}^{(t+1)}\right) \leq \frac{1}{2}\sqrt{\mathcal{D}_{KL}\left[p_{\top}^{(t+1)} || \hat{p}_{\top}^{(t+1)}\right]} \leq \frac{1}{2}\sqrt{\mathcal{D}_{KL}\left[p_{F}^{(t+1)} || p\right]}, \tag{B.17}$$

since $\hat{p}_{F}^{(t+1)} \propto p$ by definition; recall that $p(\tau) = p_{F}^{(t)}(\tau) f(\mathcal{D}_{t+1}|x)$. Here, the first inequality follows from Pinsker's inequality and the second one, from the data-processing inequality. For the second term, note that

$$\hat{p}_{\top}^{(t+1)}(x) = \frac{p_{\top}^{(t)}(x) f(\mathcal{D}_{t+1}|x)}{\sum_{y\in\mathcal{X}} p_{\top}^{(t)}(y) f(\mathcal{D}_{t+1}|y)} = \frac{p_{\top}^{(t)}(x) f(\mathcal{D}_{t+1}|x)}{\mathbb{E}_{y \in p_{\top}^{(t)}}\left[f(\mathcal{D}_{t+1}|x)\right]}, \tag{B.18}$$

with a corresponding representation of π_{t+1} as a function of π_t and $f(\mathcal{D}_{t+1}|x)$. The result follows by pluggin Equation (B.18) and Equation (B.17) into Equation (B.16).

Proofs for Chapter 5

Proof of Lemma 5.1.1

It stems directly from the trajectory balance that, for any trajectory $\tau^\star \in \mathcal{T}$:

$$Z \prod_{s \to s' \in \tau^\star} p_F(s \to s') = R(x) \prod_{s \to s' \in \tau^\star} p_B(s' \to s) \tag{B.19}$$

$$\iff Z = R(x) \prod_{s \to s' \in \tau^\star} \frac{p_B(s' \to s)}{p_F(s \to s')} \tag{B.20}$$

Therefore, applying this identity to τ and τ' and equating the right-hand-sides (RHSs) yields Equation (5.2). We are left with the task of proving the converse. Note we can rewrite Equation (5.2) as:

$$R(x) \prod_{s \to s' \in \tau} \frac{p_B(s' \to s)}{p_F(s \to s')} = R(x') \prod_{s \to s' \in \tau'} \frac{p_B(s' \to s)}{p_F(s \to s')}. \tag{B.21}$$

If Equation (5.2) holds for any pair (τ, τ'), we can vary τ' freely for a fixed τ — which implies the RHS of the above equation must be a constant with respect to τ'. Say this constant is c, then:

$$R(x) \prod_{s \to s' \in \tau} \frac{p_B(s' \to s)}{p_F(s \to s')} = c \tag{B.22}$$

$$\iff R(x) \prod_{s \to s' \in \tau} p_B(s' \to s) = c \prod_{s \to s' \in \tau} p_F(s \to s'), \tag{B.23}$$

and summing the above equation over all $\tau \in \mathcal{T}$ yields:

$$\sum_{\tau \in \mathcal{T}} R(x) \prod_{s \to s' \in \tau} p_B(s' \to s) = c \sum_{\tau \in \mathcal{T}} \prod_{s \to s' \in \tau} p_F(s \to s') \tag{B.24}$$

$$\implies \sum_{\tau \in \mathcal{T}} R(x) \prod_{s \to s' \in \tau} p_B(s' \to s) = c \tag{B.25}$$

Furthermore, note that:

$$\sum_{x \in \mathcal{X}} R(x) \sum_{\tau \in T(x)} \prod_{s \to s' \in \tau} p_B(s' \to s) = c \tag{B.26}$$

$$\implies \sum_{x \in \mathcal{X}} R(x) = c \tag{B.27}$$

$$\implies Z = c \tag{B.28}$$

Plugging $Z = c$ into Equation (B.22) yields the trajectory balance condition.

Proof of Theorem 5.2.1

The proof is based on the following reasoning. We first show that, given the satisfiability of the aggregating balance condition, the marginal distribution over the terminating states is proportional to

$$\mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right], \tag{B.29}$$

as stated in the text. Then, we verify that this distribution is the same as

$$p_{\top}(x) \propto \prod_{1 \le i \le N} R_i(x) \tag{B.30}$$

if the local balance conditions are satisfied. This proves the sufficiency of the aggregating balance condition for building a model that samples from the correct product distribution. The necessity follows from Proposition 16 of (BENGIO; LAHLOU, et al., 2023) and from the observation that the local balance conditions are equivalent to ${}^{p_F^{(i)}(\tau)}/_{p_B^{(i)}(\tau|x)} = R_i(x)$ for each $i = 1, \dots, N$.

Next, we provide a more detailed discussion about this proof. Similarly to Appendix B, notice that the contrastive nature of the aggregating balance condition implies that, if

$$\prod_{1 \le i \le N} \frac{\left(\prod_{s \to s' \in \tau} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)}{\left(\prod_{s \to s' \in \tau'} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)} = \frac{\left(\prod_{s \to s' \in \tau} \frac{p_F(s,s')}{p_B(s',s)}\right)}{\left(\prod_{s \to s' \in \tau'} \frac{p_F(s,s')}{p_B(s',s)}\right)}, \tag{B.31}$$

then

$$p_F(\tau) = c \left(\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right) p_B(\tau|x) \tag{B.32}$$

for a constant $c > 0$ that does not depend either on x or on τ. Hence, the marginal distribution over a terminating state $x \in \mathcal{X}$ is

$$\begin{aligned} p_{\top}(x) &:= \sum_{\tau \rightsquigarrow x} \prod_{s \to s' \in \tau} p_F(s \to s') && \text{(B.33)} \\ &= c \sum_{\tau \rightsquigarrow x} \left(\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right) p_B(\tau|x) && \text{(B.34)} \\ &= c \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right]. && \text{(B.35)} \end{aligned}$$

Correspondingly, ${}^{p_F^{(i)}(\tau)}/_{p_B^{(i)}(\tau|x)} \propto R_i(x)$ for every $i = 1, \dots, N$ and every τ leading to x due to the satisfiability of the local balance conditions. Thus,

$$p_{\top}(x) \propto \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} R_i(x) \right] = \prod_{1 \le i \le N} R_i(x), \tag{B.36}$$

which attests the sufficiency of the aggregating balance condition for the distributional correctness of the global model.

Proof of Theorem 5.2.2

Initially, recall that the Jeffrey divergence, known as the symmetrized KL divergence, is defined as

$$\mathcal{D}_J(p, q) = \mathrm{KL}[p||q] + \mathrm{KL}[q||p] \tag{B.37}$$

for any pair p and q of equally supported distributions. Then, let

$$\hat{\pi}(x) = \hat{Z}\, \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right] \tag{B.38}$$

be the marginal distribution over the terminating states of a GFlowNet satisfying the aggregating balance condition. On the one hand, notice that

$$\begin{aligned}
\mathrm{KL}[\pi||\hat{\pi}] &= \mathbb{E}_{x \sim \pi} \left[\log \frac{\pi(x)}{\hat{\pi}(x)} \right] && \text{(B.39)} \\
&= \mathbb{E}_{x \sim \pi} \left[\log \pi(x) - \log Z \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right] \right] && \text{(B.40)} \\
&= -\mathbb{E}_{x \sim \pi} \left[\log \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x) \pi_i(x)} \right] \right] - \log \hat{Z} + \log Z && \text{(B.41)} \\
&\le -\mathbb{E}_{x \sim \pi} \left[\log \prod_{1 \le i \le N} (1 - \alpha_i) \right] - \log \hat{Z} + \log Z && \text{(B.42)} \\
&= \log \frac{Z}{\hat{Z}} + \sum_{1 \le i \le N} \log \left(\frac{1}{1 - \alpha_i} \right), && \text{(B.43)}
\end{aligned}$$

in which $Z \coloneqq \left(\sum_{x \in \mathcal{X}} \prod_{1 \le i \le N} \pi_i(x) \right)^{-1}$ is π's normalization constant. On the other hand,

$$\begin{aligned}
\mathrm{KL}[\pi||\hat{\pi}] &= \mathbb{E}_{x \sim \hat{\pi}} \left[\log \frac{\hat{\pi}(x)}{\pi(x)} \right] && \text{(B.44)} \\
&= \mathbb{E}_{x \sim \hat{\pi}} \left[\log Z \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right] - \log \pi(x) \right] && \text{(B.45)} \\
&= \mathbb{E}_{x \sim \hat{\pi}} \left[\log \mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x) \pi_i(x)} \right] \right] + \log \hat{Z} - \log Z && \text{(B.46)} \\
&\le \mathbb{E}_{x \sim \hat{\pi}} \left[\log \prod_{1 \le i \le N} (1 + \beta_i) \right] + \log \hat{Z} - \log Z && \text{(B.47)} \\
&= \log \frac{\hat{Z}}{Z} + \sum_{1 \le i \le N} \log (1 + \beta_i). && \text{(B.48)}
\end{aligned}$$

Thus, the Jeffrey divergence between the targeted product distribution π and the effectively learned distribution $\hat{\pi}$ is

$$\begin{aligned}
\mathcal{D}_J(\pi, \hat{\pi}) &= \mathrm{KL}[\pi||\hat{\pi}] + \mathrm{KL}[\hat{\pi}||\pi] && \text{(B.49)} \\
&\le \log \frac{Z}{\hat{Z}} + \sum_{1 \le i \le N} \log \left(\frac{1}{1 - \alpha_i} \right) + \log \frac{\hat{Z}}{Z} + \sum_{1 \le i \le N} \log (1 + \beta_i) && \text{(B.50)} \\
&= \sum_{1 \le i \le N} \log \left(\frac{1 + \beta_i}{1 - \alpha_i} \right). && \text{(B.51)}
\end{aligned}$$

Proof of Theorem 5.1.1

We firstly recall the construction of the unbiased REINFORCE gradient estimator (WILLIAMS, R. J., 1992), which was originally designed as a method to implement gradient-ascent algorithms to tackle associative tasks involving stochastic rewards in reinforcement learning. Let p_θ be a probability density (or mass function) differentiably parametrized by θ and $f_\theta \colon \mathcal{X} \rightarrow \mathbb{R}$ be a real-value function over $\mathcal{X}$ possibly dependent on θ. Our goal is to estimate the gradient

$$\nabla_\theta \mathbb{E}_{x \sim p_\theta}[f_\theta(x)], \tag{B.52}$$

which is not readily computable due to the dependence of p_θ on θ. However, since

$$\begin{aligned}
\nabla_\theta \mathbb{E}_{x \sim p_\theta}[f_\theta(x)] &= \nabla_\theta \int_{x \in \mathcal{X}} f_\theta(x) p_\theta(x) \mathrm{d}x && \text{(B.53)} \\
&= \int_{x \in \mathcal{X}} \left((\nabla_\theta f_\theta(x)) p_\theta(x) \right) \mathrm{d}x + \int_{x \in \mathcal{X}} \left((\nabla_\theta p_\theta(x)) f_\theta(x) \right) \mathrm{d}x && \text{(B.54)} \\
&= \mathbb{E}_{x \sim p_\theta} \left[\nabla_\theta f_\theta(x) + f_\theta(x) \nabla_\theta \log p_\theta(x) \right], && \text{(B.55)}
\end{aligned}$$

the gradient of f_θ's expected value under p_θ may be unbiasedly estimated by averaging the quantity $\nabla_\theta f_\theta(x) + f_\theta(x) \nabla_\theta \log p_\theta(x)$ over samples of p_θ. We use this identity to compute the KL divergence between the forward and backward policies of a GFlowNet. In this sense, notice that

$$\begin{aligned}
\nabla_\theta \mathrm{KL}[p_F || p_B] &= \nabla_\theta \mathbb{E}_{\tau \sim p_F} \left[\log \frac{p_F(\tau)}{p_B(\tau)} \right] && \text{(B.56)} \\
&= \mathbb{E}_{\tau \sim p_F} \left[\nabla_\theta \log p_F(\tau) + \left(\log \frac{p_F(\tau)}{p_B(\tau)} \right) \nabla_\theta \log p_F(\tau) \right] && \text{(B.57)} \\
&= \mathbb{E}_{\tau \sim p_F} \left[\left(\log \frac{p_F(\tau)}{p_B(\tau)} \right) \nabla_\theta \log p_F(\tau) \right], && \text{(B.58)}
\end{aligned}$$

as $\mathbb{E}_{\tau \sim p_F}[\nabla_\theta \log p_F(\tau)] = \nabla_\theta \mathbb{E}_{\tau \sim p_F}[1] = 0$. In contrast, the gradient of the contrastive balance loss with respect to θ is

$$\begin{aligned}
\nabla_\theta \mathcal{L}_{CB}(\tau, \tau', \theta) &= \nabla_\theta \left(\log \frac{p_F(\tau)}{p_B(\tau)} - \log \frac{p_F(\tau')}{p_B(\tau')} \right)^2 && \text{(B.59)} \\
&= 2 \left(\log \frac{p_F(\tau)}{p_B(\tau)} - \log \frac{p_F(\tau')}{p_B(\tau')} \right) \left(\nabla_\theta \log p_F(\tau) - \nabla_\theta \log p_F(\tau') \right), && \text{(B.60)}
\end{aligned}$$

whose expectation under the outer product distribution $p_F \otimes p_F$ equals the quantity $4 \nabla_\theta \mathrm{KL}[p_F || p_B]$ in Equation (B.56). Indeed, as

$$\mathbb{E}_{\tau \sim p_F} \left[\left(\log \frac{p_F(\tau')}{p_B(\tau')} \right) \nabla_\theta \log p_F(\tau) \right] = 0, \tag{B.61}$$

with an equivalent identity obtained by interchanging τ and τ',

$$\mathop{\mathbb{E}}_{(\tau,\tau')\sim p_F\otimes p_F}\left[\nabla_\theta \mathcal{L}_{CB}(\tau,\tau',\theta)\right] = \tag{B.62}$$

$$\mathop{\mathbb{E}}_{(\tau,\tau')\sim p_F\otimes p_F}\left[2\left(\log\frac{p_F(\tau)}{p_B(\tau)} - \log\frac{p_F(\tau')}{p_B(\tau')}\right)\left(\nabla_\theta \log p_F(\tau) - \nabla_\theta \log p_F(\tau')\right)\right] = \tag{B.63}$$

$$\mathop{\mathbb{E}}_{(\tau,\tau')\sim p_F\otimes p_F}\left[2\left(\log\frac{p_F(\tau)}{p_B(\tau)}\right)\nabla_\theta \log p_F(\tau) + 2\left(\log\frac{p_F(\tau')}{p_B(\tau')}\right)\nabla_\theta \log p_F(\tau')\right] = \tag{B.64}$$

$$\mathop{\mathbb{E}}_{\tau\sim p_F}\left[4\left(\log\frac{p_F(\tau)}{p_B(\tau)}\right)\nabla_\theta \log p_F(\tau)\right] = 4\nabla_\theta \mathrm{KL}[p_F||p_B]. \tag{B.65}$$

Thus, the on-policy gradient of the contrastive balance loss equals in expectation the gradient of the KL divergence between the forward and backward policies of a GFlowNet.

Proofs for Chapter 6

We will often use $\|p-q\|_{TV}$ to represent the total variation distance between the probability measures p and q for consistency with Publication **V**.

Proof of Remark 6.1.1

The terminal states of the modified flow network will have two types of nodes, with flow $\frac{F}{g^h}$ and $\frac{F}{g^h}+\delta_i$, with $\delta_i \geq 0$ and $\sum_{i=1}^{g^{h-1}} \delta_i = \delta$. We normalize those probabilities to obtain the individual probabilities for each terminal state, which determines the density of each sample. From that, we can proceed to compute the total variation distance between $\tilde{p}_T$ and π.

$$\begin{aligned}
\|\tilde{p}_T - \pi\|_{TV} &= \frac{1}{2}\sum_{x\in\mathcal{X}} |\tilde{p}_T(x) - \pi(x)| \\
&= \frac{1}{2}\left[(g^h - g^{h-1})\left|\frac{F}{g^h}\frac{1}{F+\delta} - \frac{1}{g^h}\right| + \sum_{i=1}^{g^{h-1}}\left|\frac{F+g^h\delta_i}{g^h}\frac{1}{F+\delta} - \frac{1}{g^h}\right|\right] \\
&= \frac{1}{2}\left[\frac{g^h\delta - g^{h-1}\delta + \sum_{i=1}^{g^{h-1}}|g^h\delta_i - \delta|}{g^h(F+\delta)}\right].
\end{aligned}$$

We can lower bound $\sum_{i=1}^{g^{h-1}}|g^h\delta_i - \delta|$, by considering that $\sum_{i=1}^{g^{h-1}}(g^h\delta_i - \delta) = g^h\delta - g^{h-1}\delta$, taking the absolute value of the result and each element of the sum to obtain $g^h\delta - g^{h-1}\delta \leq \sum_{i=1}^{g^{h-1}}|g^h\delta_i - \delta|$. Thus we obtain the lower bound

$$\begin{gathered}
\frac{1}{2}\left[\frac{g^h\delta - g^{h-1}\delta + g^h\delta - g^{h-1}\delta}{g^h(F+\delta)}\right] \leq \frac{1}{2}\left[\frac{g^h\delta - g^{h-1}\delta + \sum_{i=1}^{g^{h-1}}|g^h\delta_i - \delta|}{g^h(F+\delta)}\right] \\
\left(1-\frac{1}{g}\right)\frac{\delta}{F+\delta} \leq ||\tilde{p}_T - \pi||_{TV}.
\end{gathered}$$

This lower bound is reached when all error terms in the terminal states have the same value $\delta_i = \frac{\delta}{g^h}$.

To upper bound $|g^h\delta_i - \delta|$ we apply the triangle inequality, obtaining $|g^h\delta_i - \delta| \leq g^h\delta_i + \delta$ and $\sum_{i=1}^{g^{h-1}} |g^h\delta_i - \delta| \leq g^h\delta + g^{h-1}\delta$, from which we obtain the upper bound

$$\begin{aligned}\|\tilde{p}_T - \pi\|_{TV} &\leq \frac{1}{2}\left[\frac{g^h\delta - g^{h-1}\delta + g^h\delta + g^{h-1}\delta}{g^h(F+\delta)}\right] \\ &\leq \frac{\delta}{F+\delta}.\end{aligned}$$

To obtain a tighter bound we break the sum $\sum_{i=1}^{g^{h-1}} |g^h\delta_i - \delta|$ by partitioning the sum into the first I terms $S_A = g^h \sum_{i=1}^{I} |\delta_i - \frac{\delta}{g^h}|$ with $\delta_i < \frac{\delta}{g^h}$ and subsequent $g^{h-1} - I$ terms $S_B = g^h \sum_{j=I+1}^{g^{h-1}} |\delta_j - \frac{\delta}{g^h}|$ with $\delta_j \geq \frac{\delta}{g^h}$. By construction, we know that $S_A + g^h \sum_{i=1}^{I} \delta_i + g^h \sum_{j=I+1}^{g^{h-1}} \delta_j - S_B = g^{h-1}\delta$, simplifying to $S_B - S_A = \delta(g^h - g^{h-1})$. We rewrite $S_A + S_B = S_B - S_A + 2S_A = \delta(g^h - g^{h-1}) + 2S_A$, and by triangle inequality on S_A, we obtain the upper bound $\sum_{i=1}^{g^{h-1}} |g^h\delta_i - \delta| = S_A + S_B \leq g^h\delta - g^{h-1}\delta + 2I\delta$. Setting $I = g^{h-1} - 1$ (the biggest value it can have without breaking the constraints on δ_i), it simplifies to $S_A + S_B \leq g^h\delta + g^{h-1}\delta - 2\delta$

$$\begin{aligned}\|\tilde{p}_T - \pi\|_{TV} &\leq \frac{1}{2}\left[\frac{g^h\delta - g^{h-1}\delta + \sum_{i=1}^{g^{h-1}} |g^h\delta_i - \delta|}{g^h(F+\delta)}\right] \\ &\leq \frac{1}{2}\left[\frac{g^h\delta - g^{h-1}\delta + g^h\delta + g^{h-1}\delta - 2\delta}{g^h(F+\delta)}\right] \\ &\leq \left[\frac{g^h\delta - \delta}{g^h(F+\delta)}\right] \\ &\leq \left(1 - \frac{1}{g^h}\right)\frac{\delta}{F+\delta}.\end{aligned}$$

Proof of Theorem 6.1.1

To demonstrate this result, we will need the following facts regarding the function $f(x)\colon x \in \mathbb{R}^n \mapsto \sum_{i=1}^{n} |x_i - a_i|$ for positive constants a_i.

Lemma B.0.1 (Convexity). *Let $\Delta_{n+1} = \{x \in \mathbb{R}^n \colon x_i \geq 0 \wedge \sum_{i=1}^{n} x_i = 1\}$ and $a \in \mathbb{R}^n$. Then, $f\colon \Delta_{n+1} \to \mathbb{R}$ defined by $f(x) = \sum_{i=1}^{n} |x_i - a_i|$ is convex.*

Proof. It follows from $f(\alpha x + (1-\alpha)y) = \sum_{i=1}^{n} |\alpha x_i - \alpha a_i + (1-\alpha)y_i - (1-\alpha)a_i| \leq \alpha \sum_{i=1}^{n} |x_i - a_i| + (1-\alpha)\sum_{i=1}^{n} |y_i - a_i| = \alpha f(x) + (1-\alpha)f(y)$ for any $\alpha \in [0,1]$ and $x, y \in \Delta_{n+1}$. □

Lemma B.0.2 (Maximality at edges). *Let $e_i \in \mathbb{R}^n$ satisfy $e_{ij} = 0$ for $j \neq i$ and $e_{ii} = 1$. Then, the function f from Lemma B.0.1 achieves its maximum at $\arg\max_{1 \leq i \leq n} f(e_i)$.*

Proof. We will show that, for each $x \in \Delta_{n+1}$, there is a i for which $f(e_i) \geq f(x)$. In particular, f is maximized at one of the e_i's. For this, note that

$$f(x) = f\left(\sum_{i=1}^{n} x_i e_i\right) \leq \sum_{i=1}^{n} x_i f(e_i) \leq \max_{1 \leq i \leq n} f(e_i) \tag{B.66}$$

due to the convexity of f. Thus, f is upper bounded by $\max_{1 \leq i \leq n} f(e_i)$. Conversely, there is a e_i for which this upper bound is attained. Hence, $\arg\max_x f(x) \supseteq \arg\max_{1 \leq i \leq n} f(e_i)$. □

Lemma B.0.3 (Minimality). *Let f be the function of Lemma B.0.1 and assume that $a \geq 0$ and $\sum_{i=1}^{n} a_i \leq 1$. Then, f is minimized by $1 - \sum_{i=1}^{n} a_i$.*

Proof. Choose a $j \in \{1, \dots, n\}$ arbitrarily. Since $x_j = 1 - \sum_{i=1, i \neq j}^{n} x_i$,

$$\sum_{i=1}^{n} |x_i - a_i| = \sum_{i=1, i \neq j}^{n} |a_i - x_i| + \left| a_j - 1 + \sum_{i=1, i \neq j}^{n} x_i \right| \geq \left| \sum_{i=1}^{n} a_i - 1 \right|. \tag{B.67}$$

Correspondingly, the lower bound in Equation (B.67) is achieved when $x_i = a_i$ for $i \neq j$ and $x_j = 1 - \sum_{i=1, i \neq j}^{n} a_i \geq 0$. This ensures that f is minimized by $1 - \sum_{i=1}^{n} a_i$. □

In words, Lemma B.0.1 and Lemma B.0.2 ensure that the TV distance between finitely supported distributions is convex and attains its maximum at a Dirac delta.

Proof of Theorem 6.1.1. Initially, let δ_x be the amount of extra flow reaching $x \in \mathcal{X}$ and define $\beta_x = {}^{\delta_x}/_{\delta}$. Then,

$$\|p_\top - \tilde{\pi}\|_{TV} = \frac{1}{2} \sum_{x \in \mathcal{X}} |p_\top(x) - \pi(x)| = \frac{1}{2} \sum_{x \in \mathcal{D}_{s^\star}} |p_\top(x) - \pi(x)| + \frac{1}{2} \sum_{x \in \mathcal{D}_{s^\star}^c} |p_\top(x) - \pi(x)|. \tag{B.68}$$

Since $p_\top(x) = {}^{\tilde{\pi}(x) + \delta_x}/_{F + \delta}$ for $x \in \mathcal{D}_{s^\star}$ and $p_\top(x) = {}^{\tilde{\pi}(x)}/_{F + \delta}$ for $x \in \mathcal{D}_{s^\star}^c$,

$$\sum_{x \in \mathcal{D}_{s^\star}^c} |p_\top(x) - \pi(x)| = \frac{\delta}{F + \delta} \sum_{x \in \mathcal{D}_{s^\star}^c} \pi(x). \tag{B.69}$$

On the other hand,

$$\sum_{x \in \mathcal{D}_{s^\star}} |p_\top(x) - \pi(x)| = \sum_{x \in \mathcal{D}_{s^\star}} \left| \frac{\tilde{\pi}(x) + \delta_x}{F + \delta} - \frac{\tilde{\pi}(x)}{F} \right| = \frac{\delta}{F + \delta} \sum_{x \in \mathcal{D}_{s^\star}} \left| \beta_x - \frac{\tilde{\pi}(x)}{F} \right|. \tag{B.70}$$

By Lemma B.0.2, the function $f \colon \beta \mapsto \sum_{x \in \mathcal{D}_{s^\star}} |\beta_x - \pi(x)|$ is maximized at

$$\begin{aligned} \max_{y \in \mathcal{D}_{s^\star}} f(e_y) &= \max_{y \in \mathcal{D}_{s^\star}} \sum_{x \in \mathcal{D}_{s^\star}} |e_{xy} - \pi(x)| \\ &= \max_{y \in \mathcal{D}_{s^\star}} \left(\sum_{x \in \mathcal{D}_{s^\star}, x \neq y} \pi(x) \right) + (1 - \pi(y)) \\ &= 1 + \sum_{x \in \mathcal{D}_{s^\star}} \pi(x) - 2 \min_{y \in \mathcal{D}_{s^\star}} \pi(y). \end{aligned} \tag{B.71}$$

Similarly, Lemma B.0.3 ensures that

$$\min_{\beta \in \Delta_{\#\mathcal{D}_{s^\star}+1}} f(\beta) = 1 - \sum_{x \in \mathcal{D}_{s^\star}} \pi(x). \tag{B.72}$$

Thus, since $\sum_{x \in \mathcal{D}_{s^\star}} \pi(x) = 1 - \sum_{x \in \mathcal{D}^c_{s^\star}} \pi(x)$,

$$\frac{\delta}{F+\delta}\left(1 - \sum_{x \in \mathcal{D}_{s^\star}} \pi(x)\right) \leq \|p_\top - \pi\|_{TV} \leq \frac{\delta}{F+\delta}\left(1 - \min_{y \in \mathcal{D}_{s^\star}} \pi(y)\right). \tag{B.73}$$

□

Proof of Theorem 6.2.1

As stepping stones towards proving Theorem 6.2.1, we first lay down Lemma B.0.4 and Lemma B.0.5.

Lemma B.0.4. *Let $G = (V, E)$ and $G' = (V', E')$ be two non-isomorphic trees of size at most n. Let ϕ be the node embedding map of a 1-WL GNN with at least $2n - 1$ layers. Then, $\phi_v \neq \phi_{v'}$ for all $v \in V$ and $v' \in V'$.*

Proof. Recall 1-WL GNNs can distinguish any pair of non-isomorphic trees. Let $\mathcal{T}_n$ and $\mathcal{T}'_n$ denote the sets of computation trees (CTs) for each node in G and G' after n layers, respectively. Likewise, let $\mathcal{T}_{2n-1}$ and $\mathcal{T}'_{2n-1}$ denote the sets of CTs after $2n+1$ layers. Since both graphs are non-isomorphic, 1-WL has already converged with n steps — the maximum diameter of a tree is $n-1$. Without loss of generality, $\mathcal{T}_n - \mathcal{T}'_n \neq \varnothing$, i.e., there is at least one CT in $\mathcal{T}_n$ that is not isomorphic to any tree in $\mathcal{T}'_n$. The same holds for $2n-1$ layers, i.e., $\mathcal{T}_{2n-1} - \mathcal{T}'_{2n-1} \neq \varnothing$. Note that a CT $T_n \in \mathcal{T}_n - \mathcal{T}'_n$ is also a subtree of any $T_{2n-1} \in \mathcal{T}_{2n-1}$. Since $T_n \notin \mathcal{T}'_n$, T_n is not a subtree of any CT in $\mathcal{T}'_{2n-1}$ — otherwise it would be in $\mathcal{T}'_n$ too. In other words, $\mathcal{T}_{2n-1} \cap \mathcal{T}'_{2n-1} = \varnothing$, implying directly our claim. □

Lemma B.0.5. *Let $G = (V, E)$ and $G' = (V', E')$ be any two trees of size at most n, i.e., $|V|$ and $|V'| \leq n$. Also, let $I = (U, \emptyset)$ and $I' = (U', \emptyset)$ be graphs comprising isolated nodes, and ϕ be the node embedding map of a 1-WL GNN with at least $2n-1$ layers. If $\{\phi_v, \phi_u\} = \{\phi_{v'}, \phi_{u'}\}$ for any $(v, u) \in V \times U$ and $(v', u') \in V' \times U'$, then the trees $(V \cup \{u\}, E \cup \{(v, u)\})$ and $(V' \cup \{u'\}, E' \cup \{(v', u')\})$ are isomorphic.*

Proof. If $\{\phi_v, \phi_u\} = \{\phi_{v'}, \phi_{u'}\}$, then we either have that *i)* $\phi_v = \phi_{v'}$ and $\phi_u = \phi_{u'}$ or *ii)* $\phi_v = \phi_{u'}$ and $\phi_{v'} = \phi_u$. In the first case, we can apply Lemma B.0.4 to conclude that $G \cong G'$ (with associated bijection g_1). Since $\phi_u = \phi_{u'}$, we know that $x_u = x_{u'}$ and the corresponding singleton graphs are trivially isomorphic as well (with bijection g_2). Finally, we can build a bijection g between the vertices of the merged graphs by making $g(v) = g_1(v)$ if $v \in V$ and $g(u) = g_2(u) = u'$. For the second case, Lemma B.0.4 implies G and G' are singletons with $x_u = x_{v'}$ and $x_v = x_{u'}$. The result is a totally disconnected graph, except for an edge linking nodes with identical features in both graphs. □

Armed with the previous lemmata, Theorem 6.2.1 is straightforward assuming GNN depth $2n-1$. From Lemma B.0.5, we know that the action embeddings for any two nodes have an empty intersection. Likewise, two actions have the same embedding only if they leave from the same state and arrive at the same state. Therefore, all edges in the SG receive different embeddings. Recall that GNN embeddings are fed to MLP layers, which are universal approximators given enough width. Therefore, a 1-WL GNN followed by MLP can approximate any policy forward p_F. The same applies to the backward policy p_B. We can use the same combination to get state embeddings, which allow approximating any node flow function F. Therefore, we can choose the triplet (p_F, p_B, F) respecting the DB conditions, for instance.

Proof of Theorem 6.2.2

Assume there is a 1-WL GFlowNet sampling from π. Since $\mathcal{G}$ is tree-structured, the mass arriving at $T(s_1) \cup T(s_2)$ must arrive through s — i.e., all paths from s_0 to some $x \in T(s_1) \cup T(s_2)$ traverse s. Furthermore, there is no directed path from s' to any terminal in $T(s'')$ or vice-versa, otherwise the skeleton (i.e., undirected structure) of $\mathcal{G}$ would contain a cycle. Then, $F(s,s') = \sum_{x \in T(s')} R(x)$ and $F(s,s') = \sum_{x \in T(s')} R(x)$, implying $F(s,s') \neq F(s,s'')$.

Proof of Theorem 6.2.3

Since child embeddings are included as additional inputs to LA-GFlowNets, it follows directly that LA-GFlowNets are at least as expressive as 1-WL GFlowNets. We are left with showing the converse does not hold. In Figure 24, we provide a construction for which 1-WL GFlowNets fail but LA-GFlowNets do not.

Proof of Theorem 6.3.1

Firstly, let $S = \{x_1, \ldots, x_B\} \subseteq \mathcal{X}$ and

$$e(S) = \frac{1}{2} \sum_{x \in S} \left| \frac{p_\top(x)}{p_\top(S)} - \frac{R(x)}{R(S)} \right|, \tag{B.74}$$

in which $p_\top(S) = \sum_{x \in S} p_\top(x)$ and $R(S) = \sum_{x \in S} R(x)$, as the TV distance between the restrictions of $p_\top$ and R to S. For conciseness, we write $p_\top^{(S)}(x) = {}^{p_\top(x)}/_{p_\top(S)}$ and $\pi^{(S)}(x) = {}^{R(x)}/_{R(S)}$. We also denote by $\pi(x) = {}^{R(x)}/_{R(\mathcal{X})}$ the normalized reward in $\mathcal{X}$. Similarly, we define $e(p) = \mathbb{E}_{S \sim p}[e(S)]$. Then, we first show that $e(p) = 0$ when $\mathrm{TV}(p_\top, \pi) = 0$. For this, note that $\mathrm{TV}(p_\top, \pi) = 0$ implies $p_\top(x) = \pi(x)$ for every x and hence $p_\top(S) = \pi(S)\ \forall S \subseteq \mathcal{X}$. Thus,

$$e(p) := \mathbb{E}_{S \sim p} \left[\frac{1}{2} \sum_{x \in S} |p_\top^{(S)}(x) - \pi^{(S)}(x)| \right] = 0. \tag{B.75}$$

On the other hand, assume that $e(p) = 0$. Recall that p is a distribution of full support over $\{S \subseteq \mathcal{X} \colon |S| = B\}$ and that $B \geq 2$. In particular, $e(p)$ ensures that

$$e(S, \theta) \coloneqq \frac{1}{2} \sum_{x \in S} \left| \frac{p_{\top}(x)}{p_{\top}(S)} - \frac{\pi(x)}{\pi(S)} \right| = 0. \tag{B.76}$$

Hence, $p_{\top}(S)\pi(x) = \pi(S)p_{\top}(x)$ for each S and $x \in S$. Write then $S = S' \cup \{x\}$ and conclude that $p_{\top}(S')\pi(x) = \pi(S')p_{\top}(x)$ for every S' and $x \notin S'$. Thus, by summing both members of this equality across $x' \notin S'$, we notice that

$$p_{\top}(S')(1 - \pi(S')) = \pi(S')(1 - p_{\top}(S')), \tag{B.77}$$

i.e., $p_{\top}(S') = \pi(S')$. Thus, by iterating this procedure, we conclude that $p_{\top}(x) = \pi(x)$ for all S' and $x \notin S'$. Since S' and x were chosen arbitrarily, $p_{\top}(x) = \pi(x)$ for every $x \in \mathcal{X}$. Consequently, $\mathrm{TV}(p_{\top}, \pi) = 0$. This ensures the equivalence between $e(p)$ and $\mathrm{TV}(p_{\top}, \pi)$ in terms of characterizing the GFlowNet's distributional correctness.

Proof of Corollary 6.3.1

Recall the definition of $e(S)$ in Equation (B.74). We start demonstrating that $P_S(\cdot; \beta)$ is indeed a probability distribution. Clearly, $p_{\top}(S; \beta) \geq 0$ for every $S \subseteq \mathcal{X}$. On the other hand,

$$\begin{aligned}
\sum_{S \subseteq \mathcal{X}} P_S(S; \beta) &= \sum_{S \subseteq \mathcal{X}, \#S = \beta} \binom{n-1}{\beta-1}^{-1} \underbrace{\sum_{x \in S} p_{\top}(x)}_{p_{\top}(S)} \\
&= \sum_{S \subseteq \mathcal{X}, \#S = \beta} \binom{n-1}{\beta-1}^{-1} p_{\top}(S) \\
&= \sum_{x \in \mathcal{X}} \binom{n-1}{\beta-1}^{-1} \binom{n-1}{\beta-1} p_{\top}(x) = 1,
\end{aligned} \tag{B.78}$$

since each $p_{\top}(x)$ appears exactly $\binom{n-1}{\beta-1}$ times on the sum above. Hence, $P_S(\cdot; \beta)$ is a probability distribution. As for the rest, let $\hat{e} = \mathbb{E}_{S \sim p}[e(S)]$, $\#\mathcal{X} = n$, $\mathcal{P}_{\beta} = \{S \subseteq \mathcal{X} \colon \#S = \beta\}$, and $\Delta = \frac{n}{2\beta} \max_{S \in \mathcal{P}_{\beta}} |p_{\top}(S) - \pi(S)|$. We will first show that

$$\mathrm{TV}(p_{\top}, \pi) - \hat{e} \leq \Delta. \tag{B.79}$$

Then, we will verify that $\mathrm{TV}(p_{\top}, \pi) - \hat{e} \geq -\Delta$. These inequalities will jointly imply Corollary 6.3.1. In this scenario, note there are $\binom{n-1}{\beta-1}$ subsets of $\mathcal{X}$ with β elements containing a $x \in \mathcal{X}$. Thus,

$$\mathrm{TV}(p_{\top}, \pi) = \frac{1}{2} \sum_{S \in \mathcal{P}_{\beta}} \sum_{x \in S} \binom{n-1}{\beta-1}^{-1} |p_{\top}(x) - \pi(x)|. \tag{B.80}$$

For conciseness, define $d_{TV} = \mathrm{TV}(p_\top, \pi)$. Hence,

$$\begin{aligned}
d_{TV} - \hat{e} &= \frac{1}{2} \sum_{S \in \mathcal{P}_\beta} \sum_{x \in S} \binom{n-1}{\beta-1}^{-1} |p_\top(x) - \pi(x)| - P_S(S) \left| \frac{p_\top(x)}{p_\top(S)} - \frac{\pi(x)}{\pi(S)} \right| \\
&\leq \frac{1}{2} \sum_{S \in \mathcal{P}_\beta} \sum_{x \in S} \binom{n-1}{\beta-1}^{-1} \left(\left| p_\top(x) - \frac{p_\top(S)}{\pi(S)} \pi(x) \right| + \pi(x) \left| 1 - \frac{p_\top(S)}{\pi(S)} \right| \right) \\
&\qquad\qquad - \frac{P_S(S)}{p_\top(S)} \left| p_\top(x) - \frac{\pi(S)}{p_\top(S)} \pi(x) \right| \\
&= \frac{1}{2} \sum_{S \in \mathcal{P}_\beta} \sum_{x \in S} \binom{n-1}{\beta-1}^{-1} \pi(x) \left| 1 - \frac{p_\top(S)}{\pi(S)} \right| \\
&= \frac{1}{2} \binom{n-1}{\beta-1}^{-1} \sum_{S \in \mathcal{P}_\beta} |p_\top(S) - \pi(S)| \\
&\leq \frac{1}{2} \binom{n-1}{\beta-1}^{-1} \binom{n}{\beta} \max_{S \in \mathcal{P}_\beta} |p_\top(S) - \pi(S)| = \frac{n}{2\beta} \Delta
\end{aligned} \tag{B.81}$$

since ${}^{P_S(S)}/_{p_\top(S)} = \binom{n-1}{\beta-1}^{-1}$ and there are $\binom{n}{\beta}$ β-sized subsets of $\mathcal{X}$. For the reverse inequality, notice that

$$\begin{aligned}
d_{TV} - \hat{e} &= \frac{1}{2} \sum_{S \in \mathcal{P}_\beta} \sum_{x \in S} \binom{n-1}{\beta-1}^{-1} |p_\top(x) - \pi(x)| - P_S(S) \left| \frac{p_\top(x)}{p_\top(S)} - \frac{\pi(x)}{\pi(S)} \right| \\
&\geq \frac{1}{2} \sum_{S \in \mathcal{P}_\beta} \sum_{x \in S} \binom{n-1}{\beta-1}^{-1} |p_\top(x) - \pi(x)| \\
&\qquad - P_S(S) \left(\left| \frac{p_\top(x)}{p_\top(S)} - \frac{\pi(x)}{p_\top(S)} \right| + \left| \frac{\pi(x)}{p_\top(S)} - \frac{\pi(x)}{\pi(S)} \right| \right) \\
&= -\frac{1}{2} \sum_{S \in \mathcal{P}_\beta} \binom{n-1}{\beta-1}^{-1} p_\top(S) \sum_{x \in S} \pi(x) \left| \frac{1}{p_\top(S)} - \frac{1}{\pi(S)} \right| \\
&= -\frac{1}{2} \binom{n-1}{\beta-1}^{-1} \sum_{S \in \mathcal{P}_\beta} |p_\top(S) - \pi(S)| \geq -\frac{n}{2\beta} \max_{S \in \mathcal{P}_\beta} |p_\top(S) - \pi(S)| .
\end{aligned} \tag{B.82}$$

Proof of Corollary 6.3.2

Again, recall the definition of $e(S)$ in Equation (B.74). We now provide a self-contained proof of Corollary 6.3.2, which follows from Corollary 6.3.1 and Hoeffding's inequality (ALQUIER, 2021). Firstly, let $\hat{e} = \mathbb{E}_{S \sim p}[e(S)]$ and $e_i = e(S_i)$. Since $\hat{e} - e_i \in [-1, 1]$, Hoeffding's inequality yields

$$\mathbb{E} \left[\exp \left\{ \lambda \left(\hat{e} - \frac{1}{m} \sum_{1 \leq i \leq m} e_i \right) \right\} \right] \leq \exp \left\{ \frac{\lambda^2}{2m} \right\} . \tag{B.83}$$

Then, Chernoff's bound implies

$$\mathbb{P}_{S_1, \dots, S_m} \left[\hat{e} \geq \frac{1}{m} \sum_{1 \leq i \leq m} e_i + s \right] \leq \mathbb{E} \left[\exp \left\{ \lambda \left(\hat{e} - \frac{1}{m} \sum_{1 \leq i \leq m} e_i \right) \right\} \right] e^{-\lambda s} \leq \exp \left\{ \frac{\lambda^2}{2m} - \lambda s \right\}$$

due to Equation (B.83). This upper bound is minimized when $\lambda = sm$. In this case, $\lambda^2/2m - \lambda s = -s^2m/2$. By letting $s = -2\log\delta/m$, we verify that

$$\mathbb{P}_{S_1,\dots,S_m}\left[\hat{e} \geq \frac{1}{m}\sum_{1 \leq i \leq m} e_i + \sqrt{\frac{2\log\frac{1}{\delta}}{m}}\right] \leq \delta. \tag{B.84}$$

Then, Corollary 6.3.1 and the complementary of the preceding inequality imply

$$\mathbb{P}_{S_1,\dots,S_m}\left[\mathrm{TV}(p_\top, \pi) \leq \frac{1}{m}\sum_{1 \leq i \leq m} e_i + \max_{S \subseteq \mathcal{X}, |S| = B} |p_\top(S) - \pi(S)| + \sqrt{\frac{2\log\frac{1}{\delta}}{m}}\right] \geq 1 - \delta. \tag{B.85}$$

Proof of Proposition 6.3.1

As detailed Appendix A.2, the global minimizer of both FL- and LED-GFlowNets' learning objectives satisfy $\sum_{(s,s') \in \tau} \phi(s, s') = -\log R(x)$ for every trajectory τ. Since $\mathcal{L}_{\text{LED}}(s, s') = 0$,

$$\tilde{F}(s)\exp\{\phi_\theta(s, s')\}p_F(s, s') = p_B(s', s)\tilde{F}(s')$$

for every trajectory finishing at x. Therefore, for every trajectory $\tau \rightsquigarrow x$,

$$\begin{aligned} p_F(\tau) &= p_B(\tau|x)\frac{\tilde{F}(x)}{\tilde{F}(s_o)}\prod_{(s,s') \in \tau}\exp\{-\phi(s, s')\} \\ &= p_B(\tau|x)\frac{\tilde{F}(x)}{\tilde{F}(s_o)}\exp\left\{-\sum_{(s,s') \in \tau}\phi(s, s')\right\} \\ &= p_B(\tau|x)\frac{\tilde{F}(x)}{\tilde{F}(s_o)}R(x). \end{aligned}$$

Hence,

$$p_\top(x) = \sum_{\tau \rightsquigarrow x} p_F(\tau) = \sum_{\tau \rightsquigarrow x}\frac{\tilde{F}(x)R(x)}{\tilde{F}(s_o)}p_B(\tau|x) \propto \tilde{F}(x)R(x)\sum_{\tau \rightsquigarrow x} p_B(\tau|x) = \tilde{F}(x)R(x), \tag{B.86}$$

ensuring that the marginal distribution learned by terminally unrestricted FL- and LED-GFlowNets does not necessarily match GFlowNet's target distribution.

Proofs for Chapter 7

Proof of Lemma 7.1.1

We simply note that the space of T-sized subsets of $\{1, \dots, W\}$ has size $\binom{W}{T}$ and the space of T-sized subsets of $\{2, \dots, W\}$ has size $\binom{W-1}{T}$. Since

$$\frac{\binom{W-1}{T}}{\binom{W}{T}} = \frac{W - T}{W} \to 1 \tag{B.87}$$

when $W \to \infty$, we can always find for any $\xi \in (0, 1)$ a W and a T, both of which potentially depending on ξ, for which $|\mathcal{X}'| \geq \xi|\mathcal{X}|$. For the cases considered in Figure 28, in particular, we compute the following proportions: $(32-6)/32 = 81.25\%$ and $(64-6)/64 \approx 90.63\%$.

Proof of Proposition 7.1.1

Our proof has three steps. Firstly, we use Hölder's inequality to bound the expectation of $|\pi(x) - p_\top(x)|$. Secondly, we rely on Jensen's inequality to bound $|\pi(x) - p_\top(x)|$ with an expectation of $|p_F(\tau)/p_B(\tau|x) - \pi(x)|$ over τ. Thirdly, we convert the probabilities to a log-scale with a simple technical argument based on the Taylor expansion of log. For this, let $\phi(x) = |\pi(x) - p_\top(x)|$. Then,

$$\begin{aligned}
\mathbb{E}_{x\sim q_{E,T}}[\phi(x)] &= \sum_{x\in\mathcal{X}} \phi(x) q_{E,T}(x) \\
&= \sum_{x\in\mathcal{X}} \phi(x) \cdot \frac{q_{E,T}(x)}{p_{E,T}(x)} p_{E,T}(x) \\
&\leq \left(\sum_{x\in\mathcal{X}} \phi(x)^q p_{E,T}(x)\right)^{\frac{1}{q}} \left(\sum_{x\in\mathcal{X}} \left(\frac{q_{E,T}(x)}{p_{E,T}(x)}\right)^p p_{E,T}(x)\right)^{\frac{1}{p}} \\
&= \left(\mathbb{E}_{x\sim p_{E,T}}[\phi(x)^q]\right)^{1/q} \left(\mathbb{E}_{x\sim p_{E,T}}\left[\left(\frac{q_{E,T}(x)}{p_{E,T}(x)}\right)^p\right]\right)^{\frac{1}{p}}
\end{aligned} \tag{B.88}$$

for any $p, q > 1$ such that $1/p + 1/q = 1$. For $p = q = 2$, this bound becomes

$$\mathbb{E}_{x\sim q_{E,T}}[\phi(x)] \leq \left(\mathbb{E}_{x\sim p_{E,T}}[\phi(x)^2] \left(\chi^2(q_{E,T}||p_{E,T}) + 1\right)\right)^{\frac{1}{2}}. \tag{B.89}$$

For GFlowNets, we may write $p_\top(x) = \mathbb{E}_{\tau\sim p_B}[p_F(\tau)/p_B(\tau|x)]$. Hence, by Jensen's inequality,

$$\begin{aligned}
\mathbb{E}_{x\sim p_{E,T}}\left[\phi(x)^2\right] &= \mathbb{E}_{x\sim p_{E,T}}\left[\left(\mathbb{E}_{\tau\sim p_B}\left[\frac{p_F(\tau)}{p_B(\tau|x)} - \pi(x)\right]\right)^2\right] \\
&\leq \mathbb{E}_{x\sim p_{E,T}}\left[\mathbb{E}_{\tau\sim p_B}\left[\left(\frac{p_F(\tau)}{p_B(\tau|x)} - \pi(x)\right)^2\right]\right].
\end{aligned} \tag{B.90}$$

In conclusion, we show that

$$\left(\frac{p_F(\tau)}{p_B(\tau|x)} - \pi(x)\right)^2 \lesssim \left(\log\frac{p_F(\tau)}{p_B(\tau|x)} - \log\pi(x)\right)^2. \tag{B.91}$$

In fact, let $M = \max_{\tau,x} \frac{p_F(\tau)}{p_B(\tau|x)}$, which always exists due to the finiteness of the state space. For instance, $M \leq 1$ for autoregressive generative tasks (i.e., when $p_B(\tau|x) = 1$). Thus,

$$\left(\frac{p_F(\tau)}{p_B(\tau|x)} - \pi(x)\right)^2 \leq M^2\left(\frac{p_F(\tau)}{Mp_B(\tau|x)} - \frac{\pi(x)}{M}\right)^2.$$

The lemma below, which is a direct consequence of the mean value theorem, ensures that the quantity above is bounded above by the log-squared difference between $p_F(\tau)/p_B(\tau|x)$ and $\pi(x)$.

Lemma B.0.6 (Lipschitzness of $x \mapsto e^x$). *For every $x, y \in (0,1]$, $|\log x - \log y| \geq |x - y|$.*

Proof. Consider $f\colon (-\infty, 0] \to \mathbb{R}$, $f\colon t \mapsto e^t$, and notice that $|f'(t)| = |e^t| \leq 1$. Consequently, by the mean value theorem, f is 1-Lipstchitz and $|e^t - e^s| \leq |t - s|$ for every $t, s \in (-\infty, 0]$. By letting $\log x = t$ and $\log y = s$, we conclude that $|x - y| \leq |\log x - \log y|$ for $x, y \in (0, 1]$. □

In summary, we have shown that

$$\underset{x \sim q_{E,T}}{\mathbb{E}}[\phi(x)] \lesssim \left(\mathbb{E}_{x\sim p_{E,T}}\mathbb{E}_{\tau \sim p_B}\left(\log \frac{p_F(\tau)}{p_B(\tau|x)} - \log \pi(x)\right)^2 \left(\chi^2(q_{E,T}||p_{E,T}) + 1\right)\right)^{1/2}. \tag{B.92}$$

The statement thereby follows by considering an uniform reference distribution, $q_{E,T}(x) = \frac{1}{|\mathcal{X}|}$,

$$\begin{aligned}\mathrm{TV}\left(p_\top, \pi\right) &= \frac{|\mathcal{X}|}{2}\mathbb{E}_{x\sim q_{E,T}}[\phi(x)] \\ &\lesssim \left(\mathbb{E}_{x\sim p_{E,T}}\mathbb{E}_{\tau \sim p_B}\left(\log \frac{p_F(\tau)}{p_B(\tau|x)} - \log \pi(x)\right)^2 \left(\chi^2(q_{E,T}||p_{E,T}) + 1\right)\right)^{1/2}.\end{aligned}$$

Proof of Proposition 2.4.1

For completeness, we provide a proof of Proposition 2.4.1. Clearly, it is enough to show that

$$L_{\mathrm{FCS}}(P) \leq \hat{L}_{\mathrm{FCS}}(P) + \sqrt{\frac{\eta}{2}} \ \text{ and } \ L_{\mathrm{FCS}}(P) \leq \hat{L}_{\mathrm{FCS}}(P) + \eta + \sqrt{\eta(\eta + 2\hat{L}_{\mathrm{FCS}}(P))}, \tag{B.93}$$

in which we omit the dependence of $\hat{L}_{\mathrm{FCS}}$ on the dataset $\mathcal{T}_n$ for conciseness. We recall that $\eta = \frac{\mathrm{KL}(P||Q) + \log 2\sqrt{n_\alpha}/\delta}{n_\alpha}$, with $n_\alpha = \lfloor (1-\alpha)n \rfloor$, is the complexity term that depends on the prior Q, posterior P, confidence δ, and the number of data points n_α. Notably, both inequalities directly follow from (MAURER, 2004, Theorem 5) bound: with probability $1 - \delta$ over $\mathcal{T}_n$,

$$\mathrm{kl}(\hat{L}_{\mathrm{FCS}}(P)||L_{\mathrm{FCS}}(P)) \leq \eta, \tag{B.94}$$

in which kl represents the binary KL divergence, i.e., $\mathrm{kl}(p||q) = p \log \frac{p}{q} + (1-p)\log\frac{1-p}{1-q}$. Below, we show that $\mathrm{kl}(p||q)$ is greater than or equal to $(p-q)^2/2q$ when $p < q$.

Lemma B.0.7. *(BOUCHERON; LUGOSI; MASSART, 2013, Exercise 2.8). Let $h(t) = (1-t)\log(1-t) + t$ and $p\colon \{1, 0\} \to [0,1]$ (resp. q) represent the PMF of a Bernoulli with parameter $p \in [0,1]$. Then,*

$$\mathbb{E}_{x \sim \mathcal{B}e(q)} h\left(1 - \frac{p(x)}{q(x)}\right) = \mathrm{kl}(p||q) \tag{B.95}$$

and $h(t) \geq \frac{t^2}{2}$ for $t \in [0,1]$. In particular, $\mathrm{kl}(p||q) \geq (p-q)^2/2q$ when $p \leq q$.

Proof. Equation B.95 follows from a direct algebraic manipulation of the left-hand side. On the other hand, define

$$g(t) = h(t) - \frac{t^2}{2} \tag{B.96}$$

for $t \in [0, 1]$. Then, g is continuous, $g(0) = 0$, and $g(t) \to {}^1\!/_2$ when $t \to 1$. Also, $g'(t) = -\log(1-t) - t \geq 0$ for $t \in [0, 1]$ since $-\log(1-t) = |\log(1-t)| \geq t$. In conclusion,

$$\mathbb{E}_{x\sim\mathcal{B}e(q)} h\left(1 - \frac{p(x)}{q(x)}\right) \geq \frac{1}{2}\mathbb{E}_{x\sim\mathcal{B}e(q)}\left(1 - \frac{p(x)}{q(x)}\right)^2 \geq \frac{(q-p)^2}{2q} \tag{B.97}$$

when $p \leq q$. □

By the symmetry of Equation B.94 with respect to L_{FCS} and $\hat{L}_{\text{FCS}}$, we conclude that

$$L_{\text{FCS}}(P) - \hat{L}_{\text{FCS}}(P) \leq \sqrt{2L_{\text{FCS}}(P)\eta}. \tag{B.98}$$

Under these circumstances, the inequality $L_{\text{FCS}}(P) \leq \hat{L}_{\text{FCS}}(P) + \eta + \sqrt{\eta(\eta + 2\hat{L}_{\text{FCS}}(P))}$ is obtained by solving the above quadratic inequality on $\sqrt{L_{\text{FCS}}(P)}$. Through a similar reasoning, $\text{kl}(p||q) \geq 2(p-q)^2$ by Pinsker's inequality and, consequently, $L_{\text{FCS}}(P) \leq \hat{L}_{\text{FCS}}(P) + \sqrt{{}^\eta\!/_2}$. These results jointly entail Equation (2.4).

Proof of Lemma 7.3.1

Recall that $\tilde{p}_T(x) = \sum_{\tau\to x} \tilde{p}_F(\tau)$ for $\tilde{p}_F^{\alpha,p_F} = \alpha p_U + (1-\alpha)p_F$, in which we make the dependence of $\tilde{p}_F$ on α and on the (unconstrained) policy p_F explicit. Let $\mathcal{F}(\alpha, p_F)$ be the family of such policies and $\mathcal{F}(\alpha)$ be the set of α-greedy policies. It is straightforward to see that $\mathcal{F}(\alpha)$ is a convex set. Clearly, it is enough to ensure that $\min_{\alpha>0,p_F} \min_x \tilde{p}_T^{\alpha,p_F}(x) \leq \min_x \pi(x)$, namely, that the rarest object can be sampled correctly by properly adjusting p_F and a (non-zero) α. Indeed, Bengio [ref, Theorem 8] showed that, for each given backward policy p_B and positive reward R, there is a unique forward policy p_F for which the marginal $p_\top(x) \propto R(x)$ for each $x \in \mathcal{X}$. Hence, since $\tilde{p}_T^{\alpha,p_F} = \alpha p_{U,T} + (1-\alpha)p_\top$, with $p_{U,T}$ being the marginal of p_U over $\mathcal{X}$, the realizability of $\mathcal{F}(\alpha)$ is ensured when α satisfies $\min_{x\in\mathcal{X}} \alpha p_{U,T}(x) < \min_{x\in\mathcal{X}} \pi(x)$, i.e., $\alpha < {}^{\min_x \pi(x)}\!/_{\min_x p_{U,T}(x)}$, in which case we may set a p_F such that $p_\top(x) = \frac{1}{1-\alpha}(\pi(x) - \alpha p_{U,T}(x))$. As an example, consider the set generation task, the details of which are provided in Section 7.1. There, p_U induces an uniform distribution over $\mathcal{X}$ and we may set $\alpha = {}^N\!/_2 \min_x \pi(x) < {}^{\min_x \pi(x)}\!/_{\min_x p_{T,U}(x)}$. Importantly, our analysis is not considering the (limited) expressivity of the chosen parametric model for the policy network, which touches on a mostly open problem in the deep learning literature. Rather, we are concerned with the *feasibility* of finding a transition policy $\tilde{p}_F$ *consistent* and *compatible* with the given target distribution R, in the sense of (BENGIO; LAHLOU, et al., 2023, Definition 4, Definition 20).

Proof of Theorem 7.3.1

We first show that the risk function is bounded. Then, Equation 7.3 follows directly from (MAURER, 2004, Theorem 5) and Jensen's inequality. Under these conditions, notice that

$$\mathrm{KL}(p_B || p_F) = \mathbb{E}_{\tau \sim p_B}[\log p_B(\tau)] - \mathbb{E}_{\tau \sim p_B}[\log p_F(\tau)] = -H[p_B] - \mathbb{E}_{\tau \sim p_B}[\log p_F(\tau)]. \quad \text{(B.99)}$$

Also, by definition,

$$\begin{aligned} p_F(\tau) &= \prod_{(s,s') \in \tau} p_F(s'|s) \\ &= \prod_{(s,s') \in \tau} (\alpha p_U(s'|s) + (1 - \alpha) p_F(s'|s)) \\ &\geq \alpha^{|\tau|} \prod_{(s,s') \in \tau} \frac{1}{|\mathrm{Ch}(s)|} \geq \left(\frac{\alpha}{\max_{s \in \tau} |\mathrm{Ch}(s)|} \right)^{|\tau|}, \end{aligned} \quad \text{(B.100)}$$

and, consequently,

$$-\mathbb{E}_{\tau \sim p_B}[\log p_F(\tau)] \leq - \min_{\tau} |\tau| \log \left(\frac{\alpha}{\max_{s \in \tau} |\mathrm{Ch}(s)|} \right) = \underbrace{\max_{\tau} |\tau| \log \left(\frac{\max_{s \in \tau} |\mathrm{Ch}(s)|}{\alpha} \right)}_{= M_T},$$

i.e., $\mathrm{KL}(p_B || p_F) \leq -H[p_B] + M_T$. In conclusion, the convexity of the KL divergence along with the the fact that $p_\top$ and π are respectively convex functions of p_F and p_B imply that $\mathrm{KL}(\pi || p_\top) \leq \mathrm{KL}(p_B || p_F)$. The rest follows from (MAURER, 2004, Theorem 5) applied to $\mathrm{KL}(p_B || p_F)$.

Proof of Theorem 7.3.2

Our proof has three main ingredients. Firstly, we build upon a Azuma-Hoeffding-type inequality to bound the expected transition-level error with the observed empirical error. Secondly, we derive a trajectory-level bound of the transition-level results by relying on McAllester's linear PAC-Bayes inequality. Thirdly, we combine these results with a standard union bound argument. To start with, (BEYGELZIMER et al., 2011, Theorem 1) shows that the martingale $M_t = \sum_{1 \leq i \leq t} M(S_i, S_{i-1})$ defined above, with $A_i \leq M(S_i, S_{i-1}) \leq B_i$ and $B_i - A_i \leq C$, satisfies

$$\mathbb{E}\left[\exp\left\{\lambda M_t - (e-2)\lambda^2 V_t\right\}\right] \leq 1, \quad \text{(B.101)}$$

in which $V_t = \sum_{1 \leq i \leq t} M(S_i, S_{i-1})^2$ and $\lambda \in [0, {}^1\!/\!_C]$. In our context, $|M(S_i, S_{i-1})| \leq 2U$ by the triangle inequality, and we can take $C = 2U$. By assumption, $V_t \leq K$ for all $t \leq t_m$.

Then, for the martingale $M_t(\theta)$ and corresponding $V_t(\theta)$, with θ representing the parameters of the forward policy, by Donsker-Varadhan's variational formula, we notice that

$$\mathbb{E}_{\theta \sim P}\left[\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)\right] \leq \mathrm{KL}(P || Q) + \log \mathbb{E}_{\theta \sim Q}\left[\exp\left\{\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)\right\}\right].$$

Similarly to (SELDIN et al., 2012, Theorem 1), let $\delta_t = {}^{\delta}/_{2t_m}$, so that $\sum_{1 \leq t \leq t_m} \delta_t = {}^{\delta}/_{2}$. Then, by Markov's inequality, with probability at least $1 - \delta_t$,

$$\mathbb{E}_{\theta \sim Q}\left[\exp\left\{\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)\right\}\right] \leq \frac{1}{\delta_t}\mathbb{E}\left[\mathbb{E}_{\theta \sim Q}\left[\exp\left\{\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)\right\}\right]\right];$$

where the outer expectation is with respect to the joint distribution of $\{S_1, \ldots, S_t\}$. By Tonelli's theorem and Equation B.101, the right-hand side of the equation above satisfies

$$\begin{aligned}\frac{1}{\delta_t}\mathbb{E}\left[\mathbb{E}_{\theta \sim Q}\left[\exp\left\{\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)\right\}\right]\right] &= \frac{1}{\delta_t}\mathbb{E}_{\theta \sim Q}\mathbb{E}\left[\exp\left\{\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)\right\}\right] \\ &\leq \frac{1}{\delta_t} \leq \frac{2t_m}{\delta}.\end{aligned}$$

Consequently, by Donsker-Varadhan's formula applied to $\lambda M_t(\theta) - (e-2)\lambda^2 V_t(\theta)$, bounding its exponential moment as above, and a union bound over t, yields with probability at least $1 - \frac{\delta}{2}$,

$$\mathbb{E}_{\theta \sim P}\left[M_t(\theta)\right] \leq (e-2)\lambda \mathbb{E}_{\theta \sim P}\left[V_t(\theta)\right] + \frac{\mathrm{KL}(P||Q) + \log t_m + \log \frac{2}{\delta}}{\lambda}. \tag{B.102}$$

Hence, by the definition of $M_t(\theta)$ and the bounded-variance assumption,

$$\begin{aligned}\mathbb{E}_{\theta \sim P}\left[\frac{1}{t}\sum_{1 \leq i \leq t}\mathbb{E}[\mathcal{L}_{\mathrm{DB}}(S_i, S_{i-1})|S_{<i}]\right] &\leq \frac{1}{t}\sum_{1 \leq i \leq t}\mathcal{L}_{\mathrm{DB}}(S_i, S_{i-1}) \\ &\quad + (e-2)\lambda \cdot \frac{K}{t} + \frac{\mathrm{KL}(P||Q) + \log t_m + \log {}^{2}/_{\delta}}{t\lambda}.\end{aligned} \tag{B.103}$$

Nextly, let $S_1^{(j)}$ be independent samples from a forward policy $p_F(\cdot|s_o)$ for $1 \leq j \leq n$ and $\{S_1^{(j)}, \ldots, S_{t_j}^{(j)}\}$ be the correspondingly observed trajectories. Also, we recall that

$$\mathcal{L}(\theta) = \mathbb{E}_{S_1, S_2, \ldots, S_t}\left[\frac{1}{t}\sum_{1 \leq i \leq t}\mathbb{E}\left[\mathcal{L}_{\mathrm{DB}}(S_i, S_{i-1})|S_{<i}\right]\right] \tag{B.104}$$

and define

$$\hat{L}(\theta) = \frac{1}{n}\sum_{1 \leq j \leq n}\frac{1}{t_j}\sum_{1 \leq i \leq t_j}\mathbb{E}\left[\mathcal{L}_{\mathrm{DB}}(S_i^{(j)}, S_{i-1}^{(j)})|S_{<i}^{(j)}\right]; \tag{B.105}$$

the inner expectations are computed with respect to the Markovian data-generating process (recall that the conditional expectation $\mathbb{E}[\mathcal{L}_{\mathrm{DB}}(S_i, S_{i-1})|S_{<i}]$ is a random variable). By assumption, $\mathcal{L}(\theta) \leq U$. Hence, McAllester's linear PAC-Bayes inequality (MCALLESTER, D., 2013, Theorem 2) entails, with probability at least $1 - \frac{\delta}{2}$ over draws of $\{S_1, \ldots, S_t\}$,

$$\mathbb{E}_{\theta \sim P}\left[\mathcal{L}(\theta)\right] \leq \frac{1}{\beta}\mathbb{E}_{\theta \sim P}\left[\hat{L}(\theta)\right] + \frac{U}{2\beta(1-\beta)} \cdot \frac{\mathrm{KL}(P||Q) + \log {}^{2}/_{\delta}}{n}. \tag{B.106}$$

Under these conditions, equations B.103 and B.106 jointly imply that, by a standard union-bound argument,

$$\begin{aligned}\mathbb{E}_{\theta \sim P}\left[\mathcal{L}(\theta)\right] \leq \frac{1}{\beta}\mathbb{E}_{\theta \sim P}\Bigg[\frac{1}{n}\sum_{1 \leq j \leq n}\Bigg(\frac{1}{t_j}\sum_{1 \leq i \leq t}\mathcal{L}_{\mathrm{DB}}(S_i^{(j)}, S_{i-1}^{(j)}) + \frac{(e-2)\lambda K}{t_j} + \\ \frac{\mathrm{KL}(P||Q) + \log t_m + \log {}^{2}/_{\delta}}{t_j\lambda}\Bigg)\Bigg] + \frac{U}{2\beta(1-\beta)} \cdot \frac{\mathrm{KL}(P||Q) + \log {}^{2}/_{\delta}}{n}\end{aligned}$$

with probability $1-\delta$ over draws of $(S_1, \dots, S_t)$. Since $nt_j \geq T$ for all t_j, as we observe T transitions (n is between $\lfloor T/t_{min} \rfloor$ and $\lceil T/t_m \rceil$, with t_{min} being the minimum length of a complete trajectory), and $t_j \leq t_m$, as t_m is the trajectory's maximum lenght, the result above is equivalent to

$$\mathbb{E}_{\theta \sim P}[\mathcal{L}(\theta)] \leq \frac{1}{\beta}\mathbb{E}_{\theta \sim P}\left[\underbrace{\frac{1}{n}\sum_{1 \leq j \leq n}\left(\frac{1}{t_j}\sum_{1 \leq i \leq t}\mathcal{L}_{\mathrm{DB}}(S_i^{(j)}, S_{i-1}^{(j)})\right)}_{=\hat{\mathcal{L}}(\theta)}\right] + \frac{(e-2)\lambda K}{T} +$$
$$\frac{\mathrm{KL}(P||Q) + \log t_m + \log {}^2\!/\!_\delta}{T\lambda} + \frac{U}{2\beta(1-\beta)} \cdot \frac{\mathrm{KL}(P||Q) + \log {}^2\!/\!_\delta}{n}. \tag{B.107}$$

By aggregating the terms corresponding to $\mathrm{KL}(P||Q)$ and $\log {}^2\!/\!_\delta$, we derive the desired upper bound on the expected risk of the DB loss.

Proof of Theorem 7.4.1

Intuitively, when each balance condition is satisfied, each state s on $\mathcal{I}_j$ is sampled in proportion to $F_j(s)$ and, conditioned on s, each terminal state will be sampled in proportion to ${}^{R(x)}\!/\!_{F_j(s)}$, implying that, marginally, each x is sampled proportionally to $R(x)$. In the following, we make this argument rigorous. We first consider the case in which $x \in \mathcal{X} \setminus \mathcal{S}_o$. As we are assuming that $F_j(s)p_F(\tau|s) = p_B(\tau|x)R(x)$ for each trajectory τ starting at $s \in \mathcal{I}_j$ and finishing at x, we must conclude that

$$p_\top^j(x|s) = \sum_{\tau \colon s \rightsquigarrow x} p_F(\tau|s) = \frac{R(x)}{F_j(s)} \sum_{\tau \colon s \rightsquigarrow x} p_B(\tau|x). \tag{B.108}$$

On the other hand, since $F_o(s_o)p_F^o(\tau|s) = p_B^o(\tau|s)F_j(s)$ for $s \in \mathcal{I}_j$,

$$p_\top^o(s|s_o) = \sum_{\tau \colon s_o \rightsquigarrow s} p_F(\tau|s_o) = \frac{F_j(s)}{F_o(s_o)} \sum_{\tau \colon s_o \rightsquigarrow s} p_B(\tau|s) = \frac{F_j(s)}{F_o(s_o)}, \tag{B.109}$$

as the probability of reaching s_o by starting from s and following p_B is equal to one since s_o is the only sink state of the transposed state graph. In this context,

$$\begin{aligned} p_\top(x|s_o) &= \sum_{1 \leq j \leq m} \sum_{s \in \mathcal{I}_j} p_\top^j(x|s) p_\top^o(s|s_o) \\ &= \sum_{1 \leq j \leq m} \sum_{s \in \mathcal{I}_j} \frac{F_j(s)}{F_o(s_o)} \cdot \frac{R(x)}{F_j(s)} \sum_{\tau \colon s \rightsquigarrow x} p_B^j(\tau|x) \\ &= \sum_{1 \leq j \leq m} \frac{R(x)}{F_o(s_o)} \sum_{1 \leq j \leq m} \sum_{s \in \mathcal{I}_j} \sum_{\tau \colon s \rightsquigarrow x} p_B^j(\tau|x) \\ &= \frac{R(x)}{F_o(s_o)} \sum_{s \in \bigcup_{1 \leq j \leq m} \mathcal{I}_j} p_B^j(\tau|s) = \frac{R(x)}{F_o(s_o)}; \end{aligned} \tag{B.110}$$

i.e., $p_\top(x|s_o)$ samples x proportionally to $R(x)$. For the forth line above, we relied on the fact that the probability of eaching $\bigcup \mathcal{I}_j$ is equal to one when starting at $x \in \mathcal{X} \setminus \mathcal{S}_o$ and following p_B. Correspondingly, when $x \in \mathcal{X}$, it follows from the satisfiability of the trajectory balance condition that $p_\top(x|s_o) \propto R(x)$. This ensures SAL is a sound distributed learning algorithm for GFlowNets.

Proof of Lemma 3.3.2

The global minimizer of Equation 7.5 satisfies, for every j, $F_j(s)p_F^j(\tau) = R(x)p_B^j(\tau|x)$ for every trajectory $\tau\colon s \rightsquigarrow x$ starting at $s \in \mathcal{I}_j$ and finishing at $x \in \mathcal{X}_j$. Consequently,

$$p_\top^j(x|s) = \sum_{\tau\colon s\rightsquigarrow x} p_F^j(\tau|s) = \sum_{\tau\colon s\rightsquigarrow x} \frac{p_B^j(\tau|x)R(x)}{F_j(s)} = \frac{R(x)}{F_j(s)} \sum_{\tau\colon s\rightsquigarrow x} p_B^j(\tau|s). \tag{B.111}$$

Similarly,

$$\sum_{\tau\colon s\rightsquigarrow x} F_j(s)p_F^j(\tau|s) = \sum_{\tau\colon s\rightsquigarrow x} p_B^j(\tau|x)R(x) \tag{B.112}$$

implies that

$$F_j(s) = \sum_{\tau\colon s\rightsquigarrow x} p_B^j(\tau|x)R(x) \tag{B.113}$$

since $\sum_{\tau\colon s\rightsquigarrow x} p_F^j(\tau|s)$ for every s. These equations jointly entail the proposition.

Proof of Proposition 4.3.1

As in the demonstrations above, we consider two cases in separate. First, when $x \in \mathcal{X} \cap \mathcal{S}_o$, then $p_\top(x) = {}^{R(x)}/_{Z_F+Z_R}$ due to the satisfiability of the balance condition by the model. Hence,

$$\begin{aligned}\sum_{x\in\mathcal{X}\cap\mathcal{S}_o} |p_\top(x) - \pi(x)| &= \sum_{x\in\mathcal{X}\cap\mathcal{S}_o} \left| \frac{R(x)}{Z_F+Z_R} - \frac{R(x)}{Z} \right| \\ &= \left| \frac{1}{Z_F+Z_R} - \frac{1}{Z} \right| \sum_{x\in\mathcal{X}\cap\mathcal{S}_o} R(x) = \left| \frac{1}{Z_F+Z_R} - \frac{1}{Z} \right| Z_R.\end{aligned} \tag{B.114}$$

Second, when $x \in \mathcal{X} \setminus \mathcal{S}_o$, we note that

$$\begin{aligned} p_\top(x) &= \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} \sum_{\tau\colon s_o\rightsquigarrow s\rightsquigarrow x} p_F(\tau|s_o) \\ &= \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} \left(\sum_{\tau\colon s_o\rightsquigarrow s} p_F^o(\tau|s_o) \right) \left(\sum_{\tau\colon s\rightsquigarrow x} p_F^j(\tau'|s) \right) \\ &= \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} p_\top^o(s)p_\top^j(x|s) = \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} \frac{F_j(s)}{Z_F+Z_R} \cdot p_\top^j(x|s). \end{aligned} \tag{B.115}$$

Similarly, for any $\mathcal{X}$-valued function f,

$$f(x) = \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} \frac{F_j(s)}{Z_F} \cdot f(x); \tag{B.116}$$

hence,

$$\begin{aligned} \pi(x) - p_\top(x) &= \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} \left(\frac{F_j(s)}{Z_F} \cdot \pi(x) - \frac{F_j(s)}{Z_F+Z_R} \cdot p_\top^j(x|s) \right) \\ &= \sum_{1\le j\le m} \sum_{s\in\mathcal{S}_j} \frac{F_j(s)}{Z_F} \left(\pi(x) - \frac{Z_F}{Z_F+Z_R} p_\top^j(x|s) \right) \\ &= \mathbb{E}_{s\sim p^o_{T,\setminus\mathcal{X}}} \left[\left(\pi(x) - \frac{Z_F}{Z_F+Z_R} p_\top^{f(s)}(x|s) \right) \right]. \end{aligned} \tag{B.117}$$

By recalling that $\mathrm{TV}(\pi, p_\top) = \frac{1}{2}\left(\sum_{x\in\mathcal{X}} |\pi(x) - p_\top(x)|\right)$, this result, along with Equation B.114 and Jensen's inequality applied to the function $x \mapsto |x|$, implies the proposition. To further strengthen our intuition, we also consider directly bounding the accuracy of $\mathcal{G}_o$ as a function of the trajectory-level inaccuracies of each $\mathcal{G}_j$. For this, we re-write $\pi(x)$ as

$$\pi(x) = \sum_{1\leq j\leq m} \sum_{s\in\mathcal{S}_j} \pi(x) \sum_{\tau\colon s\rightsquigarrow x} p_B^j(\tau|x). \tag{B.118}$$

Correspondingly, by recalling the property $p_\top(x) = \sum_{1\leq j\leq m} \sum_{s\in\mathcal{S}_j} p_\top^o(s) p_\top^j(x|s)$, we conclude

$$\begin{aligned} |\pi(x) - p_\top(x)| &= \left| \sum_{1\leq j\leq m} \sum_{s\in\mathcal{S}_j} \sum_{\tau\colon s\rightsquigarrow x} \pi(x) p_B(\tau|x) - p_F(\tau|s) p_\top^o(s) \right| \\ &\leq \sum_{1\leq j\leq m} \underbrace{\sum_{s\in\mathcal{S}_j} \sum_{\tau\colon s\rightsquigarrow x} \left| \pi(x) p_B^j(\tau|x) - p_F^j(\tau|s) p_\top^o(s) \right|}_{\text{Error associated to the } j\text{th client}}. \end{aligned} \tag{B.119}$$

Hence, the total variation distance between π and $p_\top$ is bounded above by

$$\mathrm{TV}(\pi, p_\top) = \frac{Z_R}{2} \left| \frac{1}{Z} - \frac{1}{Z_R + Z_F} \right| + \frac{1}{2} \sum_{x\in\mathcal{X}\setminus\mathcal{S}_o} \sum_{1\leq j\leq m} \underbrace{\sum_{s\in\mathcal{S}_j} \sum_{\tau\colon s\rightsquigarrow x} \left| \pi(x) p_B^j(\tau|x) - p_F^j(\tau|s) p_\top^o(s) \right|}_{\text{Error associated to the } j\text{th client}}.$$

For tree-shaped state graphs, the second term of the equation above can be significantly simplified by noticing that (i) each x is uniquely associated to a j, a relationship which we denote by $g(x) = j$, and (ii) that $p_B^j(\tau|x) = 1$ and $p_F^j(\tau|s) = p_\top^j(x|s)$. Under these conditions,

$$\mathrm{TV}(\pi, p_\top) \leq \frac{Z_R}{2} \left| \frac{1}{Z} - \frac{1}{Z_R + Z_F} \right| + \frac{1}{2} \sum_{x\in\mathcal{X}\setminus\mathcal{S}_o} \underbrace{\sum_{s\in\mathcal{S}_{g(x)}} \left| \pi(x) - p_\top^{g(x)}(x|s) p_\top^o(s) \right|}_{\text{Error associated to the } j=g(x)\text{th model}}. \tag{B.120}$$

Proof of Proposition 7.4.1

We proceed by strong induction on the number k of fixed-horizon partitions. For $k = 1$, the result above is equivalent to Equation 7.4.1. Assume, then, that the statement holds for j fixed-horizon partitions of the state graph for all $j < k$. Let $\mathcal{G}_i$, $0 \leq i \leq k$, be a sequence of GFlowNets satisfying the amortized trajectory balance condition. By induction, each $x \in \bigcup_{1\leq j\leq k-1} \mathcal{X}_j$ is sampled proportionally to $\sum_{1\leq j\leq k-1} \mathbf{1}[x \in \mathcal{X}_j] R_j(s)$. In particular, if $x \in \mathcal{X} \cup \bigcup_{1\leq j\leq k-1} \mathcal{X}_j$, then x is sampled proportionally to $R(x)$. For what remains, let $x \in \mathcal{X} \setminus \bigcup_{1\leq j\leq k-1} \mathcal{X}_j$. Hence, for each state $s \in \bigcup_{1\leq j\leq m_k} \mathcal{I}_{k,j} \subseteq \mathcal{X}_{k-1}$ and each trajectory $\tau\colon s \rightsquigarrow x$,

$$F_k(s) p_F(\tau|s) = p_B(\tau|x) R(x), \tag{B.121}$$

i.e., $p_F(\tau|s) = {}^{p_B(\tau|x)R(x)}/_{F_k(s)}$. Thus, by marginalizing out the non-terminal components of τ,

$$p_\top(x|s) = \frac{R(x)}{F_k(s)} \sum_{\tau\colon s\rightsquigarrow x} p_B(\tau|x) \tag{B.122}$$

and, since each s is sampled proportionally to $R_{k-1}(s) \coloneqq F_k(s)$,

$$\begin{aligned} p_\top(x) &\propto \sum_{s \in \bigcup_{1 \le j \le m_k} \mathcal{I}_{k,j}} p_\top(x|s) F_k(s) \\ &= \sum_{s \in \bigcup_{1 \le j \le m_k} \mathcal{I}_{k,j}} F_k(s) \cdot \frac{R(x)}{F_k(s)} \sum_{\tau \colon s \rightsquigarrow x} p_B(\tau|x) \\ &= R(x) \underbrace{\sum_{s \in \bigcup_{1 \le j \le m_k} \mathcal{I}_{k,j}} \sum_{\tau \colon s \rightsquigarrow x} p_B(\tau|x)}_{=1} = R(x). \end{aligned} \tag{B.123}$$

This ensures that each $x \in \mathcal{X} \setminus \bigcup_{1 \le j \le k-1} \mathcal{X}_j$ is sampled proportionally to $R(x)$. By induction, each $x \in \mathcal{X}$ is sampled proportionally to $R(x)$. Hence, the recursive instance of SAL is a sound approach for sampling objects proportionally to a reward function.