Embarrassingly Parallel GFlowNets

Tiago da Silva [1] **Luiz Max Carvalho** [1] **Amauri Souza** [2 3] **Samuel Kaski** [2 4] **Diego Mesquita** [1]

Abstract

GFlowNets are a promising alternative to MCMC sampling for discrete compositional random variables. Training GFlowNets requires repeated evaluations of the unnormalized target distribution, or reward function. However, for large-scale posterior sampling, this may be prohibitive since it incurs traversing the data several times. Moreover, if the data are distributed across clients, employing standard GFlowNets leads to intensive client-server communication. To alleviate both these issues, we propose *embarrassingly parallel* GFlowNet (EP-GFlowNet). EP-GFlowNet is a provably correct divide-and-conquer method to sample from product distributions of the form $R(\cdot) \propto R_1(\cdot)...R_N(\cdot)$ — e.g., in parallel or federated Bayes, where each R_n is a local posterior defined on a data partition. First, in parallel, we train a local GFlowNet targeting each R_n and send the resulting models to the server. Then, the server learns a global GFlowNet by enforcing our newly proposed *aggregating balance* condition, requiring a single communication step. Importantly, EP-GFlowNets can also be applied to multi-objective optimization and model reuse. Our experiments illustrate the EP-GFlowNets's effectiveness on many tasks, including parallel Bayesian phylogenetics, multi-objective multiset, sequence generation, and federated Bayesian structure learning.

1. Introduction

Generative Flow Networks (GFlowNets) (Bengio et al., 2021) are powerful sampling tools and, despite being in their infancy, have become popular alternatives to Markov Chain Monte Carlo methods in finite discrete state spaces. While the most straight-forward use of MCMC-like methods is to sample from Bayesian posteriors (Deleu et al., 2022; 2023; Atanackovic et al., 2023b), GFlowNets find varied applications in combinatorial optimization (Zhang et al., 2023a), multi-objective active learning (Jain et al., 2023), and design of biological sequences (Jain et al., 2022). Inheriting jargon from reinforcement learning, GFlowNets learn to sample from $R : \mathcal{X} \to \mathbb{R}^+$ by incrementally refining a policy function p_F that incrementally builds objects $x \in \mathcal{X}$ by augmenting an initial state s_0. In Bayesian context, R is proportional to some posterior $p(\cdot|\mathcal{D})$, and $\mathcal{X}$ is its support.

Nonetheless, similarly to running MCMC chains, training GFlowNets requires repeatedly evaluating R. For posterior sampling, this implies repeated sweeps through the data. Furthermore, if the data is scattered across many clients — e.g., in Bayesian federated learning (El Mekkaoui et al., 2021; Vono et al., 2022) —, the multiple communication rounds between clients and the server may be a bottleneck.

In the realm of MCMC, *embarrassingly parallel* methods (Neiswanger et al., 2014) address both aforementioned issues simultaneously through a divide-and-conquer scheme. In a nutshell, given an N-partition $\mathcal{D}_1, \ldots, \mathcal{D}_N$ of the data $\mathcal{D}$, the strategy consists of sampling in parallel from N *subposteriors* defined on different data shards:

$$R_n(x) \propto p(\mathcal{D}_n|x)p(x)^{1/N} \quad \forall n = 1 \ldots N,$$

and subsequently combining the results in a server to get approximate samples from the full posterior $p(x|\mathcal{D})$, or equivalently from $R \propto R_1 R_2 \ldots R_N$. In federated settings, the data partition reflects how data arises in client devices, and this class of algorithms incurs a single communication round between the clients and the server. However, these methods are tailored towards continuous random variables, and their combination step relies on approximating (or sampling from) the product of sample-based continuous surrogates of the subposteriors, e.g., kernel density estimators (Neiswanger et al., 2014), Gaussian processes (Nemeth & Sherlock, 2018; de Souza et al., 2022), or normalizing flows (Mesquita et al., 2019). Consequently, they are not well-suited to sample from discrete state spaces.

We propose EP-GFlownet, the first embarrassingly parallel sampling method for discrete distributions. EP-GFlowNets start off by learning N local GFlowNets in parallel to sample proportional to their corresponding reward functions $R_1, \ldots, R_N$ and send the resulting models to the server.

[1]Getulio Vargas Foundation [2]Aalto University [3]Federal Institute of Ceará [4]University of Manchester. Correspondence to: Diego Mesquita <diego.mesquita@fgv.br>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Then, the server learns a global GFlowNets by enforcing our newly proposed *aggregating balance* (AB) condition, which ensures the global model correctly samples from the product of (sampling distributions of) the local GFlowNets. Notably, sampling from products of GFlowNets is challenging since no simple composition (e.g., avg./product) of local policies induces the product distribution (Du et al., 2023).

Towards deriving the AB, we introduce a novel balance criterion to train conventional/local GFlowNets, the *contrastive balance* (CB) condition. Compared to broadly used training criteria (Bengio et al., 2023; Malkin et al., 2022), CB often leads to faster convergence, which may be attributed to its minimal parametrization, requiring only forward and backward policies. We also show that, in expectation, minimizing the CB is equivalent to minimizing the variance of an estimator of the log-partition from Zhang et al. (2023c).

Remarkably, the applicability of EP-GFlowNets goes beyond Bayesian inference. For instance, it applies to multi-objective tasks in which we need to sample from a combination of different objective functions (Daulton et al., 2021; Jain et al., 2023). EP-GFlowNets can also be applied for model reuse (Garipov et al., 2023), allowing sampling from log-pools of experts without retraining individual models.

Our experiments validate EP-GFlowNets in different contexts, including multi-objective multiset, parallel Bayesian phylogenetic inference, and federated Bayesian structure learning. In summary, our **contributions** are:

1. We propose EP-GFlowNet, the first algorithm for embarrassingly parallel sampling in discrete state spaces using GFlowNets. We provide theoretical guarantees of correctness, and also analyze EP-GFlowNet's robustness to errors in the estimation of local GFlowNets;
2. We present the contrastive balance (CB) condition, show it is a sufficient and necessary condition for sampling proportionally to a reward, and analyze its connection to variational inference (VI);
3. We substantiate our methodological contributions with experiments on five different tasks. Our empirical results i) showcase the accuracy of EP-GFlowNets; ii) show that, in some cases, using the CB as training criterion leads to faster convergence compared to popular loss functions; iii) illustrate EP-GFlowNets' potential in two notable applications: Bayesian phylogenetic inference, and Bayesian network structure learning.

2. Preliminaries

Notation. We represent a *directed acyclic graph* (DAG) over nodes V and with adjacency matrix $A \in \{1,0\}^{|V|\times|V|}$ as $G = (V, A)$. A *forward policy* over V in G is a function $p\colon V \times V \to \mathbb{R}_+$ such that (i) $p(v,\cdot)$ is a probability measure over V for every $v \in V$ and (ii) $p(v,w) > 0$ if and only if $A_{vw} = 1$; we alternatively write $p(v \to w)$ and $p(w|v)$ to represent $p(v,w)$. A transition kernel p induces a conditional probability measure over the space of trajectories in G: if $\tau = (v_1 \to \cdots \to v_n)$ is a trajectory of length n in G, then $p(\tau|v_1) = \prod_{i=1}^{n-1} p(v_{i+1}|v_i)$. A *backward policy* in G is a forward policy on the transpose graph $G^\intercal = (V, A^\intercal)$.

Generative flow networks. GFlowNets are a family of amortized variational algorithms trained to sample from an unnormalized distribution over discrete and compositional objects. More specifically, let $R\colon \mathcal{X} \to \mathbb{R}_+$ be an unnormalized distribution over a finite space $\mathcal{X}$. We call R a *reward* due to terminological inheritance from the reinforcement learning literature. Define a finite set $\mathcal{S}$ and a variable s_o. Then, let G be a weakly connected DAG with nodes $V = \{s_o\} \cup \{s_f\} \cup \mathcal{S} \cup \mathcal{X}$ such that (i) there are no incoming edges to s_o, (ii) there are no outgoing edges exiting s_f and (iii) there is an edge from each $x \in \mathcal{X}$ to s_f. We call the elements of V *states* and refer to $\mathcal{X}$ as the set of *terminal states*; s_o is called the *initial state* and s_f is an absorbing state designating the end of a trajectory. We denote by $\mathcal{T}$ the space of trajectories in G starting at s_o and ending at s_f. Illustratively, $\mathcal{X}$ could be the space of biological sequences of length 32; $\mathcal{S}$, the space of sequences of lengths up to 32; and s_o, an empty sequence. The training objective of a GFlowNet is to learn a forward policy p_F over G such that the marginal distribution p_T over $\mathcal{X}$ satisfies

$$p_T(x) \coloneqq \sum_{\tau\colon \tau \text{ leads to } x} p_F(\tau|s_o) \propto R(x). \tag{1}$$

We usually parameterize p_F as a neural network and select one among the diversely proposed training criteria to estimate its parameters, which we denote by ϕ_F. These criteria typically enforce a balance condition on the Markovian process defined by p_F that provably imposes the desired property on the marginal distribution in Equation (1). For example, the *trajectory balance* (TB) criterion introduces a parametrization of the target distribution's partition function Z_{ϕ_Z} and of a backward policy $p_B(\cdot,\cdot;\phi_B)$ with parameters ϕ_Z and ϕ_B, respectively, and is defined as

$$\mathcal{L}_{TB}(\tau, \phi_F, \phi_B, \phi_Z) = \left(\log Z_{\phi_Z} - \log R(x) + \sum_{s \to s' \in \tau} \log \frac{p_F(s,s';\phi_F)}{p_B(s',s;\phi_B)}\right)^2. \tag{2}$$

Minimizing Equation (2) enforces the TB condition: $p_F(\tau;\phi_F) = Z_{\phi_Z}^{-1} R(x) \prod p_B(s',s;\phi_B)$, which implies Equation (1) if valid for all $\tau \in \mathcal{T}$. This is the most widely used training scheme for GFlowNets. In practice, some works set p_B as a uniform distribution to avoid learning ϕ_B, as suggested by Malkin et al. (2022).

Another popular approach for training GFlowNets uses the notion of *detailed balance* (DB). Here, we want

to find forward and backward policies and *state flows* F (with parameters ϕ_S) that satisfy the DB condition: $F(s;\phi_S)p_F(s,s';\phi_F) = F(s';\phi_S)p_B(s',s;\phi_B)$ if s is an non-terminal state and $F(s;\phi_S)p_F(s_f|s;\phi_F) = R(s)$ otherwise. Again, satisfying the DB condition for all edges in G entails Equation (1). Naturally, this condition leads to a transition-decomposable loss

$$\mathcal{L}_{DB}(s,s',\phi_F,\phi_B,\phi_S) = \begin{cases} \left(\log \frac{p_F(s,s';\phi_F)}{p_B(s',s;\phi_B)} + \log \frac{F(s;\phi_S)}{F(s';\phi_S)}\right)^2 & \text{if } s' \neq s_f, \\ \left(\log \frac{F(s;\phi_S)p_F(s_f|s;\phi_F)}{R(s)}\right)^2 & \text{otherwise.} \end{cases} \quad (3)$$

Recently, Pan et al. (2023a) proposed to residually reparameterize F with reference on a hand-crafted function ξ, namely, $\log F(s) = \log \tilde{F}(s) + \log \xi(s)$. This approach led to the *forward-looking (FL) GFlowNet*, whose learning objective we denote by $\mathcal{L}_{FL}$. Importantly, contrarily to Pan et al. (2023a), we do not drop the boundary conditions in Equation 3. To guarantee correctness whether using $\mathcal{L}_{DB}$ or $\mathcal{L}_{TB}$, we need to integrate the loss over some exploratory policy π fully supported in $\mathcal{T}$. In practice, π is typically a ϵ-mixture between p_F and a uniform forward policy, $(1-\epsilon)\cdot p_F + \epsilon \cdot u_F$, or a tempered version of p_F. We use the former definition for π in this work. We review alternative training schemes for GFlowNets in the supplementary material.

Problem statement. Given a set of clients $n = 1,\dots,N$, each with reward function R_n, we want to learn a GFlowNet to sample proportionally to a global reward function R defined as a product of the local rewards $R_1,\dots,R_N$. We want to do so with as little client-server communication as possible and without openly disclosing the local rewards to the server. While we focus on sampling from $R(x) := \prod_{n=1}^N R_n(x)$ in the main paper, we also provide in Appendix B.2 extensions of our theoretical results to exponentially weighted rewards of the form $R(x) := \prod_{n=1}^N R_n(x)^{w_n}$ with $w_1,\dots,w_N > 0$.

3. Method

This section derives a provably correct framework for embarrassingly parallel GFlowNets based on the newly developed concept of *aggregating balance* (Section 3.1). As a cornerstone, we introduce the *contrastive balance (CB) condition*, a new balance condition requiring minimal parametrization. Notably, Section 3.2 shows the CB condition yields a sound learning objective for training conventional GFlowNets.

3.1. Embarrassingly Parallel GFlowNets

To circumvent the restrictions imposed by the problem statement, we propose a divide-and-conquer scheme. First, each client trains their own GFlowNet to sample proportionally to their local reward, sending the estimated forward and backward policies $p_F^{(n)}$ and $p_B^{(n)}$ to a centralizing server. Then, the server estimates the policies (p_F, p_B) of a novel GFlowNet that approximately samples from R solely based on the local policies $\{(p_F^{(n)}, p_B^{(n)})\}_{n=1}^N$, i.e., without ever evaluating any $R^{(n)}$. More specifically, the server learns a GFlowNet whose marginal distribution p_T over terminal states is proportional to the product of those from the clients $p_T^{(1)},\dots,p_T^{(N)}$. Toward this end, Theorem 3.1 delineates a necessary and sufficient condition guaranteeing the correctness of the aggregation phase, which we call *aggregating balance (AB) condition*. It is worth mentioning that Theorem 3.1 builds directly upon the contrastive balance condition in Lemma 3.5, discussed in detail in Section 3.2.

Theorem 3.1 (Aggregating balance condition)**.** *Let* $\left(p_F^{(1)}, p_B^{(1)}\right),\dots,\left(p_F^{(N)}, p_B^{(N)}\right) : V^2 \to \mathbb{R}^+$ *be pairs of forward and backward policies from* N *GFlowNets sampling respectively proportionally to* $R_1,\dots,R_N : \mathcal{X} \to \mathbb{R}^+$. *Then, another GFlowNet with forward and backward policies* $p_F, p_B \in V^2 \to \mathbb{R}^+$ *samples proportionally to* $R(x) := \prod_{n=1}^N R(x)$ *if and only if the following condition holds for all terminal trajectories* $\tau,\tau' \in \mathcal{T}$*:*

$$\prod_{n=1}^N \frac{\left(\prod_{s\to s'\in\tau} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)}{\left(\prod_{s\to s'\in\tau'} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)} = \frac{\left(\prod_{s\to s'\in\tau} \frac{p_F(s,s')}{p_B(s',s)}\right)}{\left(\prod_{s\to s'\in\tau'} \frac{p_F(s,s')}{p_B(s',s)}\right)}. \quad (4)$$

Based on Theorem 3.1, we can naturally derive a loss function enforcing Equation (4) that can be used to combine the locally trained GFlowNets. To guarantee the minimum of our loss achieves aggregating balance, it suffices to integrate Equation (4) against a distribution attributing non-zero mass to every $(\tau,\tau') \in \mathcal{T}^2$. Importantly, note the aggregating balance loss is agnostic to which loss was used to learn the local GFlowNets, as it only requires their transition functions and not, e.g., an estimate of the partition function or of the flows going though each state.

Corollary 3.2 (Aggregating balance loss)**.** *Let* $p_F^{(i)}$ *and* $p_B^{(i)}$ *be forward/backward transition functions such that* $p_T^{(i)}(x) \propto R_i(x)$ *for arbitrary reward functions* R_i *over 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. Moreover, assume that* $p_F(\cdot,\cdot;\phi_F)$ *and* $p_B(\cdot,\cdot;\phi_B)$ *denote the forward/backward policies of a GFlowNet parameterized by* ϕ_F *and* ϕ_B*. The following are equivalent:*

1. $p_T(x;\phi_F) \propto \prod_i R_i(x)$ *for all* $x \in \mathcal{X}$*;*
2. $\mathbb{E}_{(\tau,\tau')\sim\nu}\left[\mathcal{L}_{AB}(\tau,\tau',\phi_F,\phi_B)\right] = 0$ *where for trajectories* $\tau \rightsquigarrow x$ *and* $\tau \rightsquigarrow x'$

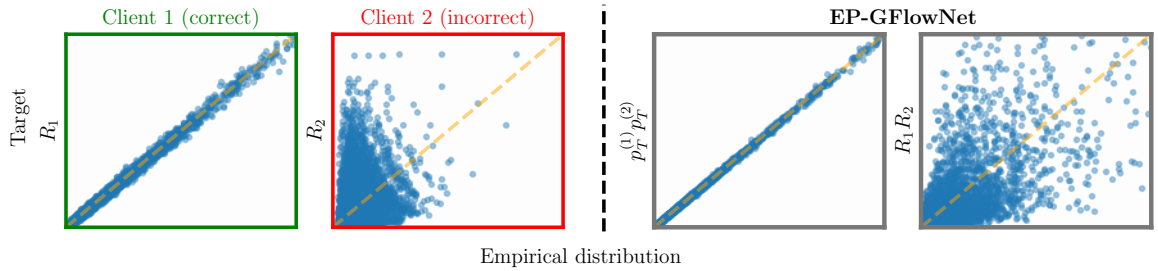

Figure 1. **EP-GFlowNet samples proportionally to a pool of locally trained GFlowNets.** If a client correctly trains their local model (green) and another client trains theirs incorrectly (red), the distribution inferred by EP-GFlowNet (mid-right) differs from the target product distribution (right).

$$\mathcal{L}_{AB}(\tau,\tau',\phi_F,\phi_B)=\Bigg(\log\frac{p_F(\tau;\phi_F)p_B(\tau'|x';\phi_B)}{p_B(\tau|x;\phi_B)p_F(\tau';\phi_F)} - \sum_{1\le i\le N}\log\frac{p_F(\tau;\phi_F)p_B(\tau'|x';\phi_B)}{p_B(\tau|x;\phi_B)p_F(\tau';\phi_F)}\Bigg)^2. \tag{5}$$

Remark 3.3 (Imperfect local inference). In practice, the local balance conditions often cannot be completely fulfilled by the local GFlowNets and the distributions $p_T^{(1)},\dots,p_T^{(N)}$ over terminal states are not proportional to the rewards $R_1,\dots,R_N$. In this case, aggregating balance implies the aggregated model samples proportionally to

$$\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\le i\le N}\frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right]. \tag{6}$$

Interestingly, the value of Equation (6) equals the expectation of a non-deterministic random variable only if the local balance conditions are not satisfied. Otherwise, the ratio $p_F^{(i)}(\tau)/p_B^{(i)}(\tau|x)$ equals $R(x)$, a constant wrt τ conditioned on τ having x as its final state. Furthermore, Equation (6) also allows us to assess the probability mass function the global EP-GFlowNet is truly drawing samples from.

As mentioned in Remark 3.3, in practice, the local GFlowNets may not be balanced with respect to their rewards, incurring errors that propagate to our aggregated model. In this context, Theorem 3.4 quantifies the extent to which these local errors impact the overall result.

Theorem 3.4 (Influence of local failures)**.** *Let $\pi_n := R_n/Z_n$ and $p_F^{(n)}$ and $p_B^{(n)}$ be the forward and backward policies of the n-th client. We use $\tau \rightsquigarrow x$ to indicate that $\tau\in\mathcal{T}$ is finished by $x\to s_f$. Suppose that the local balance conditions are lower- and upper-bounded $\forall\, n\in[[1,N]]$ as*

$$1-\alpha_n \le \min_{x\in\mathcal{X},\tau\rightsquigarrow x}\frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)\pi_n(x)} \le \max_{x\in\mathcal{X},\tau\rightsquigarrow x}\frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)\pi_n(x)} \le 1+\beta_n \tag{7}$$

where $\alpha_n\in(0,1)$ and $\beta_n>0$. The Jeffrey divergence $\mathcal{D}_J$ 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

$$\mathcal{D}_J(\pi,\hat{\pi})\le\sum_{n=1}^N\log\left(\frac{1+\beta_n}{1-\alpha_n}\right). \tag{8}$$

There are two things worth highlighting in Theorem 3.4. First, if the local models are accurately learned (i.e., $\beta_n=\alpha_n=0\,\forall n$), the RHS of Equation (8) equals zero, implying $\pi=\hat{\pi}$. Second, if either $\beta_n\to\infty$ or $\alpha_n\to 1$ for some n, the bound in Equation (8) goes to infinity — i.e., it degenerates if one of the local GFlowNets are poorly trained. This is well-aligned with the *catastrophic failure* phenomenon (de Souza et al., 2022), which was originally observed in the literature of parallel MCMC (Neiswanger et al., 2014; Nemeth & Sherlock, 2018; Mesquita et al., 2019) and refers to the incorrectness of the global model due to inadequately estimated local parameters and can result in missing modes or misrepresentation of low-density regions. Figure 1 shows a case where one of the local GFlowNets is poorly trained (Client 2's). Note that minimizing the AB objective leads to a good approximation of the product of marginal distributions over terminal states (encoded by the local GFlowNets). Nonetheless, the result is far from we have envisioned at first, i.e., the learned model significantly diverges from the product distribution $R\propto R_1R_2$. Additionally, Figure 13 in Appendix C.4 highlights that EP-GFlowNets can learn a relatively good approximation to the target distribution even in the face of inaccurate local approximations.

3.2. Contrastive balance

As a stepping stone towards proving Theorem 3.1, we develop the *contrastive balance condition*, which is sufficient for ensuring that a GFlowNet's marginal over terminal states is proportional to its target reward (Lemma 3.5).

Lemma 3.5 (Constrastive balance condition)**.** *If $p_F,p_B\in V^2\to\mathbb{R}^+$ are the forward and backward policies of a*

GFlowNet sampling proportionally to some arbitrary reward function $R : \mathcal{X} \to \mathbb{R}^+$, then, for any pair of complete trajectories $\tau, \tau' \in \mathcal{T}$ with $\tau \rightsquigarrow x$ and $\tau \rightsquigarrow x'$,

$$R(x') \prod_{s \to s' \in \tau} \frac{p_F(s, s')}{p_B(s', s)} = R(x) \prod_{s \to s' \in \tau'} \frac{p_F(s, s')}{p_B(s', s)}. \quad (9)$$

Conversely, if a GFlowNet with forward and backward policies p_F, p_B abide by Equation (9), it induces a marginal distribution over $x \in \mathcal{X}$ proportional to R.

Enforcing Lemma 3.5 results in a loss that does not depend on an estimate $\log Z_{\phi_Z}$ of the intractable log-partition function present in the TB condition. The next corollary guarantees that an instantiation of the GFlowNet parameterized by a global minimizer of $\mathcal{L}_{CB}$ (Equation (10)) correctly samples from $p(x) \propto R(x)$. We call $\mathcal{L}_{CB}$ the contrastive balance loss as it measures the contrast between randomly sampled trajectories. In practice, we observed that in some cases the CB loss leads to better results than the TB and DB losses, as we will see in Section 4.6.

Corollary 3.6 (Contrastive balance loss)**.** *Let $p_F(\cdot, \cdot; \phi_F)$ and $p_B(\cdot, \cdot; \phi_B)$ denote forward/backward policies, and $\nu : \mathcal{T}^2 \to \mathbb{R}^+$ be a full-support probability distribution over pairs of terminal trajectories. Then, $p_T(x; \phi_F) \propto R(x)$ $\forall x \in \mathcal{X}$ iff $\mathbb{E}_{(\tau,\tau') \sim \nu} [\mathcal{L}_{CB}(\tau, \tau', \phi_F, \phi_B)] = 0$ where*

$$\mathcal{L}_{CB}(\tau, \tau', \phi_F, \phi_B) = \left(\log \frac{p_F(\tau; \phi_F)}{p_B(\tau; \phi_B)} - \log \frac{p_F(\tau'; \phi_F)}{p_B(\tau'; \phi_B)} + \log \frac{R(x')}{R(x)} \right)^2. \quad (10)$$

Computational advantages of $\mathcal{L}_{CB}$. Importantly, note that $\mathcal{L}_{CB}$ incurs learning fewer parameters than TB and DB losses. Indeed, besides requiring the forward and backward policies p_F and p_B, TB requires parameterizing the partition function of R. Alternatively, DB implies using a neural network to approximate the flow through each node. In contrast, CB requires only learning p_F and p_B.

$\mathcal{L}_{CB}$ and VI. Notably, the next proposition ties the CB loss' gradient to that of a variational objective, extending the characterization of GFlowNets as VI started by Malkin et al. (2023) for the TB loss. More specifically, Theorem 3.7 states that the on-policy gradients of the CB objective coincide in expectation to the gradient of the KL divergence between the forward and backward policies.

Theorem 3.7 (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 (10) satisfies*

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

Notably, a corresponding result connecting the CB's and KL's gradients holds when we parameterize the backward policy, as we show in the Theorem [ref] in the appendix.

Connection to other balance conditions. The CB loss may be equivalently defined as the squared difference between signed violations to the TB of two independent trajectories, namely, $\mathcal{L}_{CB}(\tau, \tau') = (\mathcal{V}_{TB}(\tau) - \mathcal{V}_{TB}(\tau'))^2$, with $\mathcal{V}_{TB}(\tau) = \log \left({}^{p_F(\tau) Z} / _{p_B(\tau|x) R(x)} \right)$ satisfying $\mathcal{V}_{TB}(\tau)^2 = \mathcal{L}_{TB}(\tau)$. In this scenario, one might derive Lemma 3.5 and Theorem 3.7 as corollaries of the respective results for the TB condition by Malkin et al. (2022, Proposition 1) and by Malkin et al. (2023, Proposition 1). We present this derivation — jointly with self-contained proofs — in Appendix A and Appendix B.1. Appendix B.1 also shows that the expected value of the CB loss is proportional to the variance of a TB-based estimate of the partition function, which was used as a training loss by Zhang et al. (2023c).

4. Experiments

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* in Section 4.1, *generation of multisets* (Bengio et al., 2023; Pan et al., 2023a) in Section 4.2, *design of sequences* (Jain et al., 2022) in Section 4.3, *distributed Bayesian phylogenetic inference* (Zhang & Matsen IV, 2018) in Section 4.4, and *federated Bayesian network structure learning* (BNSL; Ng & Zhang, 2022) in Section 4.5. Since EP-GFlowNet is the first of its kind, we propose two baselines to compare it against: a centralized GFlowNet, which requires clients to disclose their rewards in a centralizing server, and a divide-and-conquer algorithm in which each client approximates its local GFlowNet with a product of categorical distributions, which 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 distributed approximate Bayesian inference (Broderick et al., 2013).

4.1. Grid world

Task description. Our grid world environment consists of a Markov decision process over an 9×9 square grid in which actions consist of choosing a direction ($\rightarrow, \uparrow$) or stopping. The reward for each state R is the sigmoid transform of its minimum distance to a reward beacon (bright yellow in Figure 2). For the distributed setting, we consider the problem of combining the rewards from different clients, each of which has two beacons placed in different positions.

Results. Figure 2 shows that EP-GFlowNet accurately approximates the targeted distribution, even in cases where

Table 1. **Quality of the parallel approximation** to the combined rewards. The table shows i) the L1 distance between the distribution induced by each method and the ground truth and ii) the average log reward of the top-800 scoring samples. Our EP-GFlowNet is consistently better than the PCVI baseline regarding L1 distance, showing approximately the same performance as a centralized GFlowNet. Furthermore, EP-GFlowNet's Top-800 score perfectly matches the centralized model, while PCVI's differ drastically. 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)

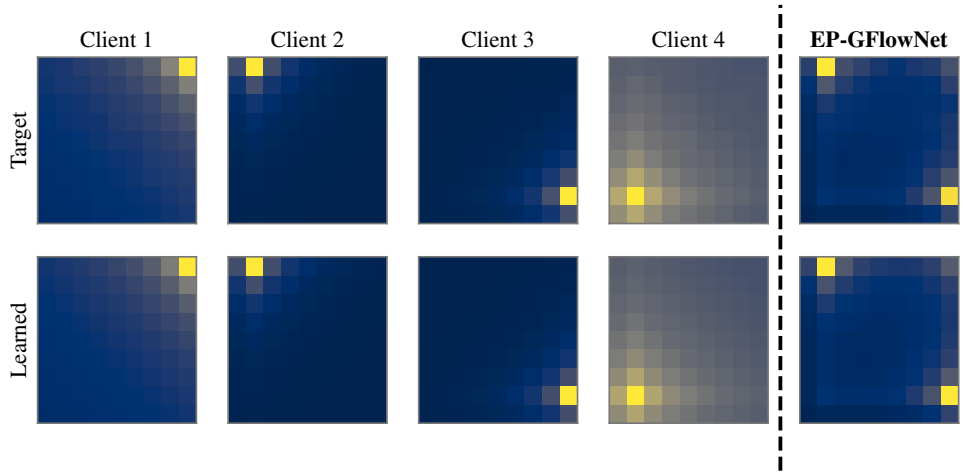

Figure 2. **Grid world.** Each heatmap represents the target distribution (first row), based on the normalized reward, and the ones learned by the local GFlowNets (second row). Results for EP-GFlowNet are in the rightmost panels. As established by Theorem 3.1, the good fit of the local models results in an accurate fit to the combined reward.

combining the client rewards leads to multiple sparsely distributed modes. Furthermore, Table 1 shows that EP-GFlowNet performs approximately on par with the centralized model — trained on the product distribution — in terms of L_1 distance (within one standard deviation), but is three orders of magnitude better than the PCVI baseline. This is also reflected in the average reward over the top 800 samples — identical to the centralized version for EP-GFlowNet, but an order of magnitude smaller for PCVI. Again, these results corroborate our theoretical claims about the correctness of our scheme for combining GFlowNets.

4.2. Multiset generation

Task description. Here, the set of terminal states comprises multisets of size S. A multiset $\mathcal{S}$ is built by iteratively adding elements from a finite dictionary U to an initially empty set. Each client n assigns a value r_u^n to each $u \in U$ and defines the log-reward of $\mathcal{S}$ as the sum of its elements' values; i.e., $\log R_n(\mathcal{S}) = \sum_{u \in \mathcal{S}} r_u^{(n)}$. In practice, the quantities $r_u^{(n)}$ are uniformly picked from the interval $[0, 1]$ for each client. We use $S = 8$ and $|U| = 10$ in our experiments.

Results. Figure 3 provides further evidence that our algorithm is able to approximate well the combined reward, even if only the local GFlowNets are given. This is further supported by the results in Table 1. Notably, EP-GFlowNet is roughly eight times more accurate than the PCVI baseline.

4.3. Design of sequences

Task description. This tasks revolves around building sequences of maximum size S. We start with an empty sequence $\mathcal{S}$ and proceed by iteratively appending an element from a fixed dictionary U. The process halts when (i) we select a special terminating token or (ii) the sequence length reaches S. In the distributed setting, we assume each client n has a score $p_s^{(n)}$ to each of the S positions within the sequence and a score $t_u^{(n)}$ to each of the $|U|$ available tokens, yielding the logarithmic reward of a sequence $\mathcal{S} = (u_1, \ldots, u_M)$ as $\log R_n(\mathcal{S}) = \sum_{i=1}^{M} p_i^{(n)} t_{u_i}^{(n)}$.

Results. Again, Figure 4 corroborates Theorem 3.1 and

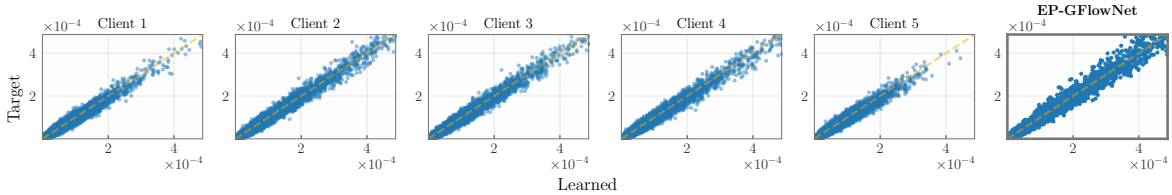

Figure 3. **Multisets: learned $\times$ ground truth distributions.** Plots compare target vs. distributions learned by GFlowNets. The five plots to the left show local models were accurately trained. Thus, a well-trained EP-GFlowNet (right) approximates well the combined reward.

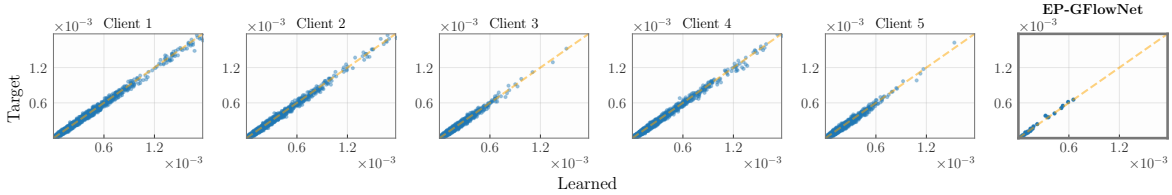

Figure 4. **Sequences: learned $\times$ ground truth distributions.** . Plots compare target to distributions learned. The five leftmost plots show local GFlowNets were well trained. Hence, as implied by Theorem 3.1, EP-GFlowNet approximates well the combined reward.

shows that EP-GFlowNet accurately samples from the product of rewards. Table 1 further reinforces this conclusion, showing a small gap in L_1 distance between EP-GFlowNet and the centralized GFlowNet trained with access to all rewards. Notably, our method is $\approx 8\times$ more accurate than PCVI. Furthermore, the Top-800 average reward of EP-GFlowNet perfectly matches the centralized model.

4.4. Bayesian phylogenetic inference

Task description. In this task, we are interested on inferring a *phylogeny* $\mathrm{T} = (t, b)$, which is a characterization of the evolutionary relationships between biological species and is composed by a tree topology t and its $(2N-1)$-dimensional vector of non-negative branch lengths b. The topology t is as a leaf-labeled complete binary tree with N leaves, each corresponding to a species. Notably, T induces a probability distribution P over the space of nucleotide sequences $Y_1, \ldots, Y_M \in \Omega^N$, where Ω is a vocabulary of *nucleobases* and Y_m denotes the nucleobases observed at the m-th site for each species. Assume t is rooted in some node r and that $\pi \in \Delta^{|\Omega|}$ is the prior probability distribution over the nucleobases' frequencies at r. Then, the marginal likelihood of a nucleobase Y_m occurring *site* m for node n is recursively defined by Felsenstein (1981)'s algorithm as

$$\mathbf{P}_n(Y_m|\mathrm{T}) = \begin{cases} \text{One-Hot}(Y_{m,n}) & \text{if } n \text{ is a leaf,} \\ [\mathbf{M}(n, n_l) \odot \mathbf{M}(n, n_r)]^\intercal & \text{otherwise,} \end{cases}$$

in which n_l and n_r are respectively the left and right children of n; $b_{n,a}$ is the length of the branch between nodes n and a; $Q \in \mathbb{R}^{|\Omega|\times|\Omega|}$ is an instantaneous rate-conversion matrix for the underlying substitution rates between nucleotides, which is given beforehand; and $\mathbf{M}(n,k) = e^{b_{n,k}Q}\mathbf{P}_k(Y_m|\mathrm{T})^\intercal$ represents the mutation probabilities from entity n to entity k after a time $b_{n,k}$. In this context, the marginal likelihood of the observed data within the site m is $\mathbf{P}_r(Y_m|\mathrm{T})^\intercal\pi$ and, assuming conditional independence of the sites given T, the overall likelihood of the data is $\mathbf{P}(\mathbf{Y}|\mathrm{T}) = \prod_{1\leq i\leq M}(\mathbf{P}_r(Y_i|\mathrm{T})^\intercal\pi)$ — which is naturally log-additive on the sites. For our experiments, π is a uniform distribution. For simplicity, we consider constant branch length, fixed throughout the experiments. For the distributed setting, we place a uniform prior over t and split 2500 nucleotide sites across five clients. In parallel MCMC (Neiswanger et al., 2014) fashion, each client trains a GFlowNet to sample from its local posterior, proportional to the product of its local likelihood and a scaled version of the prior. We further detail the generative process for building phylogenetic trees in Figure 8 in Appendix C.1.

Results. Figure 5 shows that EP-GFlowNet accurately learns the posterior distribution over the tree's topologies: the L_1 error between the learned distribution and the targeted product distribution is 0.088, whereas the average L_1 error among the clients is $0.083(\pm 0.041)$. Noticeably, this indicates the model's aptitude to learn a posterior distribution in a decentralized manner. Moreover, our results suggest the potential usefulness of GFlowNets as a scalable alternative to the notoriously inefficient MCMC-based algorithms (Zhang & Matsen IV, 2018) in the field of evolutionary biology. Indeed, Figure 12 at Appendix C.1 highlights that our distributed framework significantly reduces the training runtime for GFlowNets for Bayesian inference relatively to a centralized approach. Notably, naive strategies, like the PCVI baseline, consistently lead to sampling elements that do not belong to the support of our posterior (i.e., are invalid) — which is why we do not compare against it. In future endeavors, we plan to investigate joint parallel inference on the tree's topology and its branches' lengths using hybrid-space GFlowNets (Deleu et al., 2023). Importantly, our method is also the first provably correct algorithm enabling distributed Bayesian inference over discrete objects, which may become invaluable in real-world problems with several thousands of sites (Stamatakis & Aberer, 2013).

EP-GFlowNets' scalability wrt number of clients. To illustrate the effect of the number of clients, we perform the training of EP-GFlowNets with an increasing number of clients and a fixed data set. Strikingly, Figure 6 shows that EP-GFlowNets are approximately robust to the number of chunks the data is partitioned into — both in terms of convergence speed and accuracy.

4.5. Federated Bayesian network structure learning

Task description. While structure learning is usually carried out on centralized data, there are situations in which the possibly sensitive data is scattered across a number of clients (Reisach et al., 2021; Lorch et al., 2021). In this case, local datasets may be small and individually provide insufficient information to draw meaningful conclusions. Thus, we extend the work of Deleu et al. (2022) and show that EP-GFlowNets can efficiently learn beliefs over Bayesian networks in a federated setting (Ng & Zhang, 2022), drawing strength from multiple data sources. To this end, let $\mathbf{X} \in \mathbb{R}^d$ be a random variable and $\mathbf{B} \in \mathbb{R}^{d\times d}$ be a real-valued matrix with sparsity pattern determined by a DAG $\mathcal{G}$ with nodes $[[1, d]]$. We assume that $\mathbf{X}$ follows a linear structural equation model $\mathbf{X} = \mathbf{B}\mathbf{X} + \mathbf{N}$, with $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ representing independent Gaussian noise (Bielby & Hauser, 1977). Also, we consider 10 fixed clients, each owing a private dataset upon which a DAG-GFlowNet (Deleu et al., 2022) is trained. Our objective is to train a GFlowNet sampling proportionally to the belief distribution defined by the product of the local rewards without directly accessing the clients' datasets, akin to (Ng & Zhang, 2022). See Appendix D for further discussion.

Results. Figure 14 in Appendix D shows that the distribution learned by EP-GFlowNets over some topological features of the DAG accurately matches the target product distribution. Also, Figure 15 highlights that the federated model finds structures with a significantly higher score wrt the complete data than the clients'. Importantly, we remark that the clients could refine the global distribution by incorporating an expert's knowledge in a personalized fashion (Bharti et al., 2022). However, we leave this to future works.

4.6. Evaluating the CB loss

Section 3.2 presents the CB loss as a natural development given the theory of EP-GFlowNets. To evaluate its utility as a criterion to train GFlowNets in the conventional centralized (non-parallel) setting, we report the evolution during training of the L_1 error of the GFlowNet wrt the normalized reward for models trained using DB, TB, and our CB. We do so for all tasks in our experiments, with all GFlowNets using the same architectures for the forward and backward policies (more details in supplement). Notably, CB led to the best convergence rate in the multiset generation, phylogeny and BNSL tasks (Figure 7), while still performing on par with DB in the remaining domains. An explanation is that CB incurs a considerably simpler parametrization than DB and TB — as we do not require estimating the flow going through each state or the target partition function. Indeed, when using Deleu et al. (2022)'s reparametrization for DB, which obviates the estimation of state flows from state graphs where all states are terminal, implemented for grid world and sequences in Figure 7, DB and CB perform similarly. Moreover, to ensure the observed difference between CB and TB is not due to an insufficiently high learning rate for the log-partition function, Appendix C.4 shows results comparing the CB to TB with different lr's for $\log Z_{\phi_Z}$ in the multiset experiments (Figure 9). Noticeably, CB outperforms TB for all rates tested. A rigorous understanding, however, of the different virtues of the diversely proposed balance conditions remains a lacking issue in the literature and is an important course of action for future research.

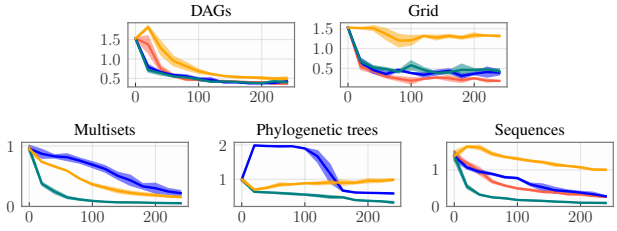

Figure 7. $\mathcal{L}_{CB}$ performs competitively with $\mathcal{L}_{TB}$, $\mathcal{L}_{DB}$ and $\mathcal{L}_{DBC}$ in the training of conventional GFlowNets.

5. Conclusions

We proposed EP-GFlowNet as a simple and elegant solution for distributed inference over discrete distributions and federated learning of GFlowNets, which we validate on an extensive suite of experiments. Our method enjoys theoretical guarantees and builds on the concept of contrastive balance (CB). Our theoretical analysis i) guarantees correctness when local models are perfectly trained and ii) allows us to quantify the impact of errors of local models on the global one. We also observed that using CB loss led to faster convergence for the local clients when intermediate states are not terminal — while being otherwise competitive.

Remarkably, we believe EP-GFlowNets pave the way for a range of applications of distributed and federated discrete Bayesian inference. We also believe EP-GFlowNets will be useful to scale up Bayesian inference by amortizing the cost of expensive likelihood computations over different clients. In the realm of multi-objective optimization, EP-GFlowNets enable sampling from a combination of rewards (Jain et al., 2023) by leveraging pre-trained GFlowNets — even without directly accessing the rewards. as recently explored by Garipov et al. (2023). We also believe the modular nature of EP-GflowNets may be instrumental in enabling the reuse and combination of large-scale models (Du et al., 2023).

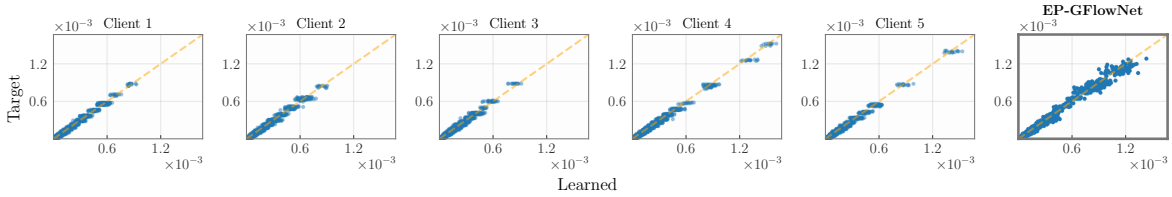

Figure 5. **Bayesian phylogenetic inference: learned $\times$ ground truth distributions.** Following the pattern in Figures 2-4, the goodness-of-fit from local GFlowNets (Clients 1-5) is directly reflected in the distribution learned by EP-GFlowNet.

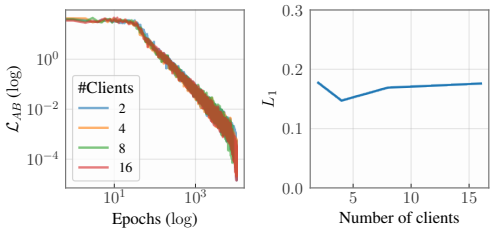

Figure 6. **EP-GFlowNet are relatively stable** wrt the number of clients. The learning curve of the aggregation phase (left) and the accuracy achieved by the resulting model in terms of L_1 norm (right) are roughly the same for varying number of clients.

Impact statement

We proposed a framework to approximate log-pools of GFlowNets using another GFlowNet. Our method has a broad range of applications in large-scale Bayesian inference, federated learning, and multi-objective optimization. Similar to other sampling methods, our work inherits the ethical concerns involved in formulating its target distribution — and, e.g., may propagate negative societal biases or be used to generate malicious content. On the other hand, our framework enables model reuse, which can have a positive impact on the amount of carbon emissions stemming from training large generative models.

Acknowledgments

Tiago da Silva, Luiz Max Carvalho, and Diego Mesquita acknowledge the support of the Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ, grant 200.151/2023), the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP, grant 2023/00815-6), and the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq, grant 404336/2023-0). Amauri Souza and Samuel Kaski were supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI), EU Horizon 2020 (European Network of AI Excellence Centres ELISE, grant agreement 951847), UKRI Turing AI World-Leading Researcher Fellowship (EP/W002973/1).

We acknowledge Aalto Science-IT Project from Computer Science IT and FGV TIC for the provided comp. resources.

References

Atanackovic, L., Tong, A., Hartford, J., Lee, L. J., Wang, B., and Bengio, Y. DynGFN: Towards bayesian inference of gene regulatory networks with gflownets. In *Advances in Neural Processing Systems (NeurIPS)*, 2023a.

Atanackovic, L., Tong, A., WANG, B., Lee, L. J., Bengio, Y., and Hartford, J. DynGFN: Towards bayesian inference of gene regulatory networks with GFlownets. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=e7MK5Vq44Q.

Bengio, E., Jain, M., Korablyov, M., Precup, D., and Bengio, Y. Flow network based generative models for non-iterative diverse candidate generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. Gflownet foundations. *Journal of Machine Learning Research (JMLR)*, 2023.

Bharti, A., Filstroff, L., and Kaski, S. Approximate bayesian computation with domain expert in the loop, 2022.

Bielby, W. T. and Hauser, R. M. Structural equation models. *Annual review of sociology*, 3(1):137–161, 1977.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. Streaming variational bayes. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

Chang, Q., Qu, H., Zhang, Y., Sabuncu, M., Chen, C., Zhang, T., and Metaxas, D. N. Synthetic learning: Learn from distributed asynchronized discriminator gan without sharing medical image data. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

da Silva, T., Silva, E., Ribeiro, A., Góis, A., Heider, D., Kaski, S., and Mesquita, D. Human-in-the-loop causal discovery under latent confounding using ancestral gflownets. *arXiv preprint:2309.12032*, 2023.

Daulton, S., Balandat, M., and Bakshy, E. Parallel bayesian optimization of multiple noisy objectives with expected hypervolume improvement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

de Souza, D., Mesquita, D., Kaski, S., and Acerbi, L. Parallel MCMC without embarrassing failures. In *Artificial Intelligence and Statistics (AISTATS)*, 2022.

Deleu, T. and Bengio, Y. Generative flow networks: a markov chain perspective, 2023.

Deleu, T., Góis, A., Emezue, C. C., Rankawat, M., Lacoste-Julien, S., Bauer, S., and Bengio, Y. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2022.

Deleu, T., Nishikawa-Toomey, M., Subramanian, J., Malkin, N., Charlin, L., and Bengio, Y. Joint Bayesian inference of graphical structure and parameters with a single generative flow network. In *Advances in Neural Processing Systems (NeurIPS)*, 2023.

Du, Y., Durkan, C., Strudel, R., Tenenbaum, J. B., Dieleman, S., Fergus, R., Sohl-Dickstein, J., Doucet, A., and Grathwohl, W. Reduce, reuse, recycle: compositional generation with energy-based diffusion models and mcmc. In *Proceedings of the 40th International Conference on Machine Learning*. JMLR.org, 2023.

El Mekkaoui, K., Mesquita, D., Blomstedt, P., and Kaski, S. Federated stochastic gradient langevin dynamics. In *Uncertainty in artificial intelligence (UAI)*, 2021.

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

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Garipov, T., Peuter, S. D., Yang, G., Garg, V., Kaski, S., and Jaakkola, T. Compositional sculpting of iterative generative processes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Geffner, T., Antoran, J., Foster, A., Gong, W., Ma, C., Kiciman, E., Sharma, A., Lamb, A., Kukla, M., Pawlowski, N., et al. Deep end-to-end causal inference. *arXiv preprint arXiv:2202.02195*, 2022.

Graves, A. and Graves, A. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Hernandez-Garcia, A., Saxena, N., Jain, M., Liu, C.-H., and Bengio, Y. Multi-fidelity active learning with gflownets, 2023.

Hinton, G. E. Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8): 1771–1800, 08 2002.

Hong, J., Zhu, Z., Yu, S., Wang, Z., Dodge, H. H., and Zhou, J. Federated adversarial debiasing for fair and transferable representations. In *ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD)*, 2021.

Hu, E. J., Jain, M., Elmoznino, E., Kaddar, Y., Lajoie, G., Bengio, Y., and Malkin, N. Amortizing intractable inference in large language models. *arXiv preprint 2310.04363*, 2023a.

Hu, E. J., Malkin, N., Jain, M., Everett, K. E., Graikos, A., and Bengio, Y. Gflownet-em for learning compositional latent variable models. In *International Conference on Machine Learning (ICLR)*, 2023b.

Husmeier, D. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic bayesian networks. *Bioinformatics*, 19 (17):2271–2282, November 2003. ISSN 1367-4803. doi: 10.1093/bioinformatics/btg313. URL http://dx.doi.org/10.1093/bioinformatics/btg313.

Jain, M., Bengio, E., Hernandez-Garcia, A., Rector-Brooks, J., Dossou, B. F. P., Ekbote, C. A., Fu, J., Zhang, T., Kilgour, M., Zhang, D., Simine, L., Das, P., and Bengio, Y. Biological sequence design with GFlowNets. In *International Conference on Machine Learning (ICML)*, 2022.

Jain, M., Raparthy, S. C., Hernandez-Garcia, A., Rector-Brooks, J., Bengio, Y., Miret, S., and Bengio, E. Multi-objective GFlowNets. In *International Conference on Machine Learning (ICML)*, 2023.

Jukes, T. H. and Cantor, C. R. *Evolution of Protein Molecules*, pp. 21–132. Elsevier, 1969. ISBN 9781483232119. doi: 10.1016/b978-1-4832-3211-9.50009-7. URL http://dx.doi.org/10.1016/B978-1-4832-3211-9.50009-7.

Lahlou, S., Deleu, T., Lemos, P., Zhang, D., Volokhova, A., Hernández-Garcıa, A., Ezzine, L. N., Bengio, Y., and Malkin, N. A theory of continuous generative flow networks. In *International Conference on Machine Learning (ICML)*, 2023.

Lorch, L., Rothfuss, J., Schölkopf, B., and Krause, A. Dibs: Differentiable bayesian structure learning. *Advances in Neural Information Processing Systems*, 34:24111–24123, 2021.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Maas, A. L., Hannun, A. Y., Ng, A. Y., et al. Rectifier nonlinearities improve neural network acoustic models. In *International Conference on Machine Learning (ICML)*, 2013.

Malkin, N., Jain, M., Bengio, E., Sun, C., and Bengio, Y. Trajectory balance: Improved credit assignment in GFlownets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Malkin, N., Lahlou, S., Deleu, T., Ji, X., Hu, E., Everett, K., Zhang, D., and Bengio, Y. GFlowNets and variational inference. *International Conference on Learning Representations (ICLR)*, 2023.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017a.

McMahan, H. B., Moore, E., Ramage, D., Hampson, S., and y Arcas, B. A. Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics (AISTATS)*, 2017b.

Mesquita, D., Blomstedt, P., and Kaski, S. Embarrassingly parallel MCMC using deep invertible transformations. In *Uncertainty in Artificial Intelligence (UAI)*, 2019.

Neiswanger, W., Wang, C., and Xing, E. P. Asymptotically exact, embarrassingly parallel MCMC. In *Uncertainty in Artificial Intelligence (UAI)*, 2014.

Nemeth, C. and Sherlock, C. Merging MCMC subposteriors through Gaussian-process approximations. *Bayesian Analysis*, 13(2):507–530, 2018.

Ng, I. and Zhang, K. Towards federated bayesian network structure learning with continuous optimization. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Pan, L., Malkin, N., Zhang, D., and Bengio, Y. Better training of GFlowNets with local credit and incomplete trajectories. In *International Conference on Machine Learning (ICML)*, 2023a.

Pan, L., Zhang, D., Courville, A., Huang, L., and Bengio, Y. Generative augmented flow networks. In *International Conference on Learning Representations (ICLR)*, 2023b.

Pan, L., Zhang, D., Jain, M., Huang, L., and Bengio, Y. Stochastic generative flow networks. In *Uncertainty in Artificial Intelligence (UAI)*, 2023c.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Pearl, J. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Pearl, J. *Causality*. Cambridge University Press, Cambridge, UK, 2 edition, 2009. ISBN 978-0-521-89560-6. doi: 10.1017/CBO9780511803161.

Pearl, J. and Mackenzie, D. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1st edition, 2018. ISBN 046509760X.

Polato, M. Federated variational autoencoder for collaborative filtering. In *International Joint Conference on Neural Networks (IJCNN)*, 2021.

Qu, H., Zhang, Y., Chang, Q., Yan, Z., Chen, C., and Metaxas, D. Learn distributed gan with temporary discriminators. In *European Conference on Computer Vision (ECCV)*, 2020.

Reisach, A. G., Seiler, C., and Weichwald, S. Beware of the simulated dag! causal discovery benchmarks may be easy to game. *Advances in Neural Information Processing Systems*, 34, 2021.

Richter, L., Boustati, A., Nüsken, N., Ruiz, F., and Akyildiz, O. D. Vargrad: A low-variance gradient estimator for variational inference. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Scott, S. L., Blocker, A. W., Bonassi, F. V., Chipman, H. A., George, E. I., and McCulloch, R. E. Bayes and big data: The consensus monte carlo algorithm. *International Journal of Management Science and Engineering Management*, 11, 2016.

Shen, M. W., Bengio, E., Hajiramezanali, E., Loukas, A., Cho, K., and Biancalani, T. Towards understanding and improving gflownet training. *arXiv preprint arXiv:2305.07170*, 2023.

Stamatakis, A. and Aberer, A. J. Novel parallelization schemes for large-scale likelihood-based phylogenetic inference. In *2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, pp. 1195–1204. IEEE, 2013.

Vono, M., Plassier, V., Durmus, A., Dieuleveut, A., and Moulines, E. Qlsd: Quantised langevin stochastic dynamics for bayesian federated learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2022.

Wang, X., Guo, F., Heller, K. A., and Dunson, D. B. Parallelizing MCMC with random partition trees. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Xu, J., Tong, X., and Huang, S.-L. Personalized federated learning with feature alignment and classifier collaboration. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=SXZr8aDKia.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *International Conference on Learning Representations (ICLR)*, 2019.

Zhang, C. and Matsen IV, F. A. Variational bayesian phylogenetic inference. In *International Conference on Learning Representations (ICLR)*, 2018.

Zhang, D., Chen, R. T., Malkin, N., and Bengio, Y. Unifying generative models with gflownets and beyond. *ICML Beyond Bayes workshop*, 2022a.

Zhang, D., Malkin, N., Liu, Z., Volokhova, A., Courville, A., and Bengio, Y. Generative flow networks for discrete probabilistic modeling. In *International Conference on Machine Learning (ICML)*, 2022b.

Zhang, D., Dai, H., Malkin, N., Courville, A., Bengio, Y., and Pan, L. Let the flows tell: Solving graph combinatorial optimization problems with gflownets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Zhang, D., Pan, L., Chen, R. T., Courville, A., and Bengio, Y. Distributional gflownets with quantile flows. *arXiv preprint arXiv:2302.05793*, 2023b.

Zhang, D. W., Rainone, C., Peschl, M., and Bondesan, R. Robust scheduling with gflownets. In *International Conference on Learning Representations (ICLR)*, 2023c.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. DAGs with NO TEARS: Continuous Optimization for Structure Learning. In *Advances in Neural Information Processing Systems*, 2018.

Zhou, M., Yan, Z., Layne, E., Malkin, N., Zhang, D., Jain, M., Blanchette, M., and Bengio, Y. Phylogfn: Phylogenetic inference with generative flow networks, 2023.

A. Proofs

A.1. Proof of Lemma 3.5

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

$$Z \prod_{s \to s' \in \tau^\star} p_F(s \to s') = R(x) \prod_{s \to s' \in \tau^\star} p_B(s' \to s) \tag{11}$$

$$\iff Z = R(x) \prod_{s \to s' \in \tau^\star} \frac{p_B(s' \to s)}{p_F(s \to s')} \tag{12}$$

Therefore, applying this identity to τ and τ' and equating the right-hand-sides (RHSs) yields Equation (9). We are left with the task of proving the converse. Note we can rewrite Equation (9) as:

$$R(x) \prod_{s \to s' \in \tau} \frac{p_B(s' \to s)}{p_F(s \to s')} = R(x') \prod_{s \to s' \in \tau'} \frac{p_B(s' \to s)}{p_F(s \to s')}. \tag{13}$$

If Equation (9) 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:

$$R(x) \prod_{s \to s' \in \tau} \frac{p_B(s' \to s)}{p_F(s \to s')} = c \tag{14}$$

$$\iff R(x) \prod_{s \to s' \in \tau} p_B(s' \to s) = c \prod_{s \to s' \in \tau} p_F(s \to s'), \tag{15}$$

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

$$\sum_{\tau \in \mathcal{T}} R(x) \prod_{s \to s' \in \tau} p_B(s' \to s) = c \sum_{\tau \in \mathcal{T}} \prod_{s \to s' \in \tau} p_F(s \to s') \tag{16}$$

$$\implies \sum_{\tau \in \mathcal{T}} R(x) \prod_{s \to s' \in \tau} p_B(s' \to s) = c \tag{17}$$

Furthermore, note that:

$$\sum_{x \in \mathcal{X}} R(x) \sum_{\tau \in T(x)} \prod_{s \to s' \in \tau} p_B(s' \to s) = c \tag{18}$$

$$\implies \sum_{x \in \mathcal{X}} R(x) = c \tag{19}$$

$$\implies Z = c \tag{20}$$

Plugging $Z = c$ into Equation (14) yields the trajectory balance condition.

A.2. Proof of Theorem 3.1

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

$$\mathbb{E}_{\tau \sim p_B(\cdot|x)} \left[\prod_{1 \le i \le N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)} \right], \tag{21}$$

as stated in Remark 3.3. Then, we verify that this distribution is the same as

$$p_T(x) \propto \prod_{1 \le i \le N} R_i(x) \tag{22}$$

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 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 $i = 1, \ldots, N$.

Next, we provide a more detailed discussion about this proof. Similarly to Appendix A.1, notice that the contrastive nature of the aggregating balance condition implies that, if

$$\prod_{1\leq i\leq N} \frac{\left(\prod_{s\to s'\in\tau} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)}{\left(\prod_{s\to s'\in\tau'} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)} = \frac{\left(\prod_{s\to s'\in\tau} \frac{p_F(s,s')}{p_B(s',s)}\right)}{\left(\prod_{s\to s'\in\tau'} \frac{p_F(s,s')}{p_B(s',s)}\right)}, \tag{23}$$

then

$$p_F(\tau) = c\left(\prod_{1\leq i\leq N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right) p_B(\tau|x) \tag{24}$$

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

$$p_T(x) \coloneqq \sum_{\tau\rightsquigarrow x} \prod_{s\to s'\in\tau} p_F(s\to s') \tag{25}$$

$$= c\sum_{\tau\rightsquigarrow x}\left(\prod_{1\leq i\leq N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right) p_B(\tau|x) \tag{26}$$

$$= c\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\leq i\leq N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right]. \tag{27}$$

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

$$p_T(x) \propto \mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\leq i\leq N} R_i(x)\right] = \prod_{1\leq i\leq N} R_i(x), \tag{28}$$

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

A.3. Proof of Theorem 3.4

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

$$\mathcal{D}_J(p, q) = \mathcal{D}_{KL}[p||q] + \mathcal{D}_{KL}[q||p] \tag{29}$$

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

$$\hat{\pi}(x) = \hat{Z}\,\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\leq i\leq N} \frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right] \tag{30}$$

be the marginal distribution over the terminating states of a GFlowNet satisfying the aggregating balance condition (see Remark 3.3 and Appendix A.2). On the one hand, notice that

$$\mathcal{D}_{KL}[\pi||\hat{\pi}] = \mathbb{E}_{x\sim\pi}\left[\log\frac{\pi(x)}{\hat{\pi}(x)}\right] \tag{31}$$

$$= \mathbb{E}_{x\sim\pi}\left[\log\pi(x) - \log Z\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\le i\le N}\frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right]\right] \tag{32}$$

$$= -\mathbb{E}_{x\sim\pi}\left[\log\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\le i\le N}\frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)\pi_i(x)}\right]\right] - \log\hat{Z} + \log Z \tag{33}$$

$$\le -\mathbb{E}_{x\sim\pi}\left[\log\prod_{1\le i\le N}(1-\alpha_i)\right] - \log\hat{Z} + \log Z \tag{34}$$

$$= \log\frac{Z}{\hat{Z}} + \sum_{1\le i\le N}\log\left(\frac{1}{1-\alpha_i}\right), \tag{35}$$

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,

$$\mathcal{D}_{KL}[\pi||\hat{\pi}] = \mathbb{E}_{x\sim\hat{\pi}}\left[\log\frac{\hat{\pi}(x)}{\pi(x)}\right] \tag{36}$$

$$= \mathbb{E}_{x\sim\hat{\pi}}\left[\log Z\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\le i\le N}\frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right] - \log\pi(x)\right] \tag{37}$$

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

$$\le \mathbb{E}_{x\sim\hat{\pi}}\left[\log\prod_{1\le i\le N}(1+\beta_i)\right] + \log\hat{Z} - \log Z \tag{39}$$

$$= \log\frac{\hat{Z}}{Z} + \sum_{1\le i\le N}\log\left(1+\beta_i\right). \tag{40}$$

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

$$\mathcal{D}_J(\pi,\hat{\pi}) = \mathcal{D}_{KL}[\pi||\hat{\pi}] + \mathcal{D}_{KL}[\hat{\pi}||\pi] \tag{41}$$

$$\le \log\frac{Z}{\hat{Z}} + \sum_{1\le i\le N}\log\left(\frac{1}{1-\alpha_i}\right) + \log\frac{\hat{Z}}{Z} + \sum_{1\le i\le N}\log\left(1+\beta_i\right) \tag{42}$$

$$= \sum_{1\le i\le N}\log\left(\frac{1+\beta_i}{1-\alpha_i}\right). \tag{43}$$

A.4. Proof of Theorem 3.7

We firstly recall the construction of the unbiased REINFORCE gradient estimator (Williams 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 $f_\theta\colon \mathcal{X}\to\mathbb{R}$ be a real-value function over $\mathcal{X}$ possibly dependent on θ. Our goal is to estimate the gradient

$$\nabla_\theta\mathbb{E}_{x\sim p_\theta}[f_\theta(x)], \tag{44}$$

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

$$\nabla_\theta \mathbb{E}_{x\sim p_\theta}[f_\theta(x)] = \nabla_\theta \int_{x\in\mathcal{X}} f_\theta(x) p_\theta(x) \mathrm{d}x \tag{45}$$

$$= \int_{x\in\mathcal{X}} \left((\nabla_\theta f_\theta(x)) p_\theta(x)\right) \mathrm{d}x + \int_{x\in\mathcal{X}} \left((\nabla_\theta p_\theta(x)) f_\theta(x)\right) \mathrm{d}x \tag{46}$$

$$= \mathbb{E}_{x\sim p_\theta}\left[\nabla_\theta f_\theta(x) + f_\theta(x) \nabla_\theta \log p_\theta(x)\right], \tag{47}$$

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

$$\nabla_\theta \mathcal{D}_{KL}[p_F || p_B] = \nabla_\theta \mathbb{E}_{\tau\sim p_F}\left[\log \frac{p_F(\tau)}{p_B(\tau)}\right] \tag{48}$$

$$= \mathbb{E}_{\tau\sim p_F}\left[\nabla_\theta \log p_F(\tau) + \left(\log \frac{p_F(\tau)}{p_B(\tau)}\right) \nabla_\theta \log p_F(\tau)\right] \tag{49}$$

$$= \mathbb{E}_{\tau\sim p_F}\left[\left(\log \frac{p_F(\tau)}{p_B(\tau)}\right) \nabla_\theta \log p_F(\tau)\right], \tag{50}$$

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

$$\nabla_\theta \mathcal{L}_{CB}(\tau, \tau', \theta) = \nabla_\theta \left(\log \frac{p_F(\tau)}{p_B(\tau)} - \log \frac{p_F(\tau')}{p_B(\tau')}\right)^2 \tag{51}$$

$$= 2\left(\log \frac{p_F(\tau)}{p_B(\tau)} - \log \frac{p_F(\tau')}{p_B(\tau')}\right)\left(\nabla_\theta \log p_F(\tau) - \nabla_\theta \log p_F(\tau')\right), \tag{52}$$

whose expectation under the outer product distribution $p_F \otimes p_F$ equals the quantity $4\nabla_\theta \mathcal{D}_{KL}[p_F||p_B]$ in Equation (48). Indeed, as

$$\mathbb{E}_{\tau\sim p_F}\left[\left(\log \frac{p_F(\tau')}{p_B(\tau')}\right) \nabla_\theta \log p_F(\tau)\right] = 0, \tag{53}$$

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

$$\mathbb{E}_{(\tau,\tau')\sim p_F\otimes p_F}[\nabla_\theta \mathcal{L}_{CB}(\tau, \tau', \theta)] = \tag{54}$$

$$\mathbb{E}_{(\tau,\tau')\sim p_F\otimes p_F}\left[2\left(\log \frac{p_F(\tau)}{p_B(\tau)} - \log \frac{p_F(\tau')}{p_B(\tau')}\right)\left(\nabla_\theta \log p_F(\tau) - \nabla_\theta \log p_F(\tau')\right)\right] = \tag{55}$$

$$\mathbb{E}_{(\tau,\tau')\sim p_F\otimes p_F}\left[2\left(\log \frac{p_F(\tau)}{p_B(\tau)}\right) \nabla_\theta \log p_F(\tau) + 2\left(\log \frac{p_F(\tau')}{p_B(\tau')}\right) \nabla_\theta \log p_F(\tau')\right] = \tag{56}$$

$$\mathbb{E}_{\tau\sim p_F}\left[4\left(\log \frac{p_F(\tau)}{p_B(\tau)}\right) \nabla_\theta \log p_F(\tau)\right] = 4\nabla_\theta \mathcal{D}_{KL}[p_F||p_B]. \tag{57}$$

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.

B. Additional theoretical results

This section rigorously lays out further theoretical results which were only briefly stated in the main paper. Firstly, section B.1 (i) shows that $\mathcal{L}_{CB}$ is, in expectation, equivalent to the *variance loss* considered by Zhang et al. (2023c); and (ii) provides alternative and shorter proofs for Corollary 3.6 and Theorem 3.7 based on previously published results for $\mathcal{L}_{TB}$. Secondly, section B.2 extends our aggregation scheme to accommodate generic logarithmic pooling of GFlowNets, significantly expanding the potential applicability of EP-GFlowNets to personalized federated learning (by implementing a client-level fine-tuned log-pool (Xu et al., 2023)) and to model composition (such as negation (Du et al., 2023; Garipov et al., 2023)).

B.1. Relationship of $\mathcal{L}_{CB}$ to other losses

Alternative proofs for Corollary 3.6 and Theorem 3.7. To start with, we define $\mathcal{V}_{TB}(\tau) = \log p_F(\tau) + \log Z - \log p_B(\tau|x_\tau) - \log R(x_\tau)$ for the signed violation to the TB condition in log-space; x_τ represents the terminal state of the complete trajectory τ. Note then that

$$\mathcal{L}_{CB}(\tau, \tau') = \left(\mathcal{V}_{TB}(\tau) - \mathcal{V}_{TB}(\tau')\right)^2 \text{ and } \mathcal{L}_{TB}(\tau) = \mathcal{V}_{TB}(\tau)^2. \tag{58}$$

In this context, Proposition 1 of (Malkin et al., 2022) states the sufficiency of the condition $\mathcal{V}_{TB}(\tau)^2 = 0$ — and equivalently of $\mathcal{V}_{TB}(\tau) = 0$ — for each τ to ensure that the GFlowNets' policies sample proportionally to the reward distribution $R(x)$. As a consequence, $\mathcal{L}_{CB}(\tau, \tau') = 0$ ensures that $\mathcal{V}_{TB}$ is constant and, in particular, that there exists a $c > 0$ such that

$$\mathcal{V}_{TB} + (\log c - \log Z) = \log p_F(\tau) + \log c - \log p_B(\tau|x_\tau) - \log R(x_\tau) = 0. \tag{59}$$

This condition corresponds to a reparametrization of the TB and entails, by Proposition 1 of (Malkin et al., 2022), Corollary 3.6. Moreover, since

$$\mathbb{E}_{(\tau,\tau')\sim p_F \otimes p_F}\left[\nabla_{\theta_F} \mathcal{L}_{CB}(\tau, \tau')\right] = \mathbb{E}_{(\tau,\tau')\sim p_F \otimes p_F}\left[2\left(\mathcal{V}_{TB}(\tau) - \mathcal{V}_{TB}(\tau')\right) \nabla_{\theta_F}\left(\mathcal{V}_{TB}(\tau) - \mathcal{V}_{TB}(\tau')\right)\right] \tag{60}$$

by the chain rule,

$$\mathbb{E}_{\tau\sim p_F}\left[\nabla_{\theta_F} \mathcal{L}_{TB}(\tau)\right] = \mathbb{E}_{\tau\sim p_F}\left[2\mathcal{V}_{TB}(\tau)\nabla_{\theta_F}\mathcal{V}_{TB}(\tau)\right] = 2\mathcal{D}_{KL}\left[p_F||p_B\right] \tag{61}$$

by Proposition 1 of (Malkin et al., 2023), and

$$\mathbb{E}_{\tau\sim p_F}\left[\nabla_{\theta_F} \mathcal{V}_{TB}(\tau)\right] = \sum_\tau p_F(\tau)\nabla_{\theta_F} \log p_F(\tau) = \nabla_{\theta_F} \sum_\tau p_F(\tau) = 0, \tag{62}$$

one may conclude that

$$\mathbb{E}_{(\tau,\tau')\sim p_F \otimes p_F}\left[\mathcal{L}_{CB}(\tau, \tau')\right] = 4\mathcal{D}_{KL}\left[p_F||p_B\right], \tag{63}$$

thereby showing Theorem 3.7.

Relationship of $\mathcal{L}_{CB}$ to $\mathcal{L}_{VL}$. Zhang et al. (2023c) proposed to train a GFlowNet by minimizing an expectation of the *variance loss*,

$$\mathcal{L}_{VL}(\tau) = \left(\mathcal{V}_{TB}(\tau) - \mathbb{E}_{\tau'}[\mathcal{V}_{TB}(\tau')]\right)^2; \tag{64}$$

in practice, the inner expectation was locally estimated using the average violation of batch of trajectories. Notably, the CB loss, derived from our proposed contrastive balance condition, equals twice $\mathcal{L}_{VL}$ in expectation — when the training policy samples the trajectories independently, since

$$\mathbb{E}_{\tau,\tau'}\left[\left(\mathcal{V}_{TB}(\tau) - \mathcal{V}_{TB}(\tau')\right)^2\right] = 2\mathbb{E}_\tau\left[\left(\mathcal{V}_{TB}(\tau) - \mathbb{E}_{\tau'}\left[\mathcal{V}_{TB}(\tau')\right]\right)^2\right] = 2\mathbb{E}_\tau\left[\mathcal{L}_{VL}(\tau)\right]. \tag{65}$$

Remarkably, this equation may also be elegantly deduced from the general fact that the expectation of the squared difference between two i.i.d. random variables equals twice their variance.

$\mathcal{L}_{CB}$ and VI when p_B is parameterized. The theorem below shows that the CB loss coincides with the KL divergence between the forward and backward policies in terms of expected gradients when p_F and p_B are disjointly parameterized.

Theorem B.1. *Let ϕ_B represent the parameters of the backward policy. Also, let $p_B \otimes p_B$ be the outer-product distribution assigning probability $\frac{1}{Z^2}R(x)R(x')p_B(\tau|x)p_B(\tau|x')$ to each pair of trajectories (τ, τ') terminating respectively at x and x'. Then, the CB loss satisfies*

$$\mathbb{E}_{\tau,\tau'\sim p_B \otimes p_B}\left[\nabla_{\phi_B} \mathcal{L}_{CB}(\tau, \tau'; \phi_B)\right] = \frac{1}{4}\mathcal{D}_{KL}\left[p_B||p_F\right]. \tag{66}$$

Importantly, the parameterization of p_B doubles the number of forward and backward passes performed in training, which often obfuscates the advantages enacted by the improved credit assignment provided by an appropriate p_B (Shen et al., 2023). Hence, while acknowledging the importance of designing effective learning schemes for p_B, we fix the backward policy as an uniform.

B.2. Exponentially weighted distributions

This section extends our theoretical results and shows how to train an EP-GFlowNet to sample from a logarithmic pool of locally trained GFlowNets. Henceforth, let $R_1, \ldots, R_N \colon \mathcal{X} \to \mathbb{R}_+$ be non-negative functions over $\mathcal{X}$ and assume that each client $n = 1, \ldots, N$ trains a GFlowNet to sample proportionally to R_n. The next propositions show how to train a GFlowNet to sample proportionally to an exponentially weighted distribution $\prod_{n=1}^N R_n(x)^{\omega_n}$ for non-negative weights $\omega_1, \ldots, \omega_N$. We omit the proofs since they are essentially identical to the ones presented in Appendix A.

Firstly, Theorem 3.1′ below proposes a modified balance condition for the global GFlowNet and shows that the satisfiability of this condition leads to a generative model that samples proportionally to the exponentially weighted distribution.

Theorem 3.1′ (Aggregating balance condition)**.** *Let $\left(p_F^{(1)}, p_F^{(1)}\right), \ldots, \left(p_F^{(N)}, p_F^{(N)}\right) : V^2 \to \mathbb{R}^+$ be pairs of forward and backward policies from N GFlowNets sampling respectively proportional to $R_1, \ldots, R_N : \mathcal{X} \to \mathbb{R}^+$. Then, another GFlowNet with forward and backward policies $p_F, p_B \in V^2 \to \mathbb{R}^+$ samples proportionally to $R(x) := \prod_{n=1}^N R(x)^{\omega_n}$ if and only if the following condition holds for any terminal trajectories $\tau, \tau' \in \mathcal{T}$:*

$$\prod_{1 \le i \le N} \frac{\left(\prod_{s \to s' \in \tau} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)^{\omega_i}}{\left(\prod_{s \to s' \in \tau'} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)^{\omega_i}} = \frac{\left(\prod_{s \to s' \in \tau} \frac{p_F(s,s')}{p_B(s',s)}\right)}{\left(\prod_{s \to s' \in \tau'} \frac{p_F(s,s')}{p_B(s',s)}\right)}. \tag{67}$$

Secondly, Theorem 3.4′ provides an upper bound on the discrepancy between the targeted and the learned global distributions under controlled local errors — when the local distributions are heterogeneously pooled. Notably, it suggests that the effect of the local failures over the global approximation may be mitigated by reducing the weights associated with improperly trained local models.

Theorem 3.4′ (Influence of local failures)**.** *Let $\pi_n := R_n / Z_n$ and $p_F^{(n)}$ and $p_B^{(n)}$ be the forward and backward policies of the nth client. We use $\tau \rightsquigarrow x$ to indicate that $\tau \in \mathcal{T}$ is finished by $x \to s_f$. Suppose that the local balance conditions are lower- and upper-bounded $\forall\, n = 1, \ldots, N$ as per*

$$1 - \alpha_n \le \min_{x \in \mathcal{X}, \tau \rightsquigarrow x} \frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)\pi_n(x)} \le \max_{x \in \mathcal{X}, \tau \rightsquigarrow x} \frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)\pi_n(x)} \le 1 + \beta_n \tag{68}$$

where $\alpha_n \in (0, 1)$ and $\beta_n > 0$. The Jeffrey divergence $\mathcal{D}_J$ between the global model $\hat{\pi}(x)$ that fulfills the aggregating balance condition in Equation (67) and $\pi(x) \propto \prod_{n=1}^N \pi_n(x)^{\omega_n}$ then satisfies

$$\mathcal{D}_J(\pi, \hat{\pi}) \le \sum_{n=1}^N \omega_n \log\left(\frac{1 + \beta_n}{1 - \alpha_n}\right). \tag{69}$$

Interestingly, one could train a *conditional* GFlowNet (Bengio et al., 2021; Zhang et al., 2023c) to build an amortized generative model able to sample proportionally to $\prod_{n=1}^N R_n(x)^{\omega_n}$ for any non-negative weights $(\omega_1, \ldots, \omega_N)$ within a prescribed set. This is a promising venue for future research.

C. Additional experiments and implementation details

This section is organized as follows. First, Appendix C.1 describes the experimental setup underlying the empirical evaluation of EP-GFlowNets in Section 4. Second, Appendix C.2 exhibits the details of the variational approximations to the combined distributions used as baselines in Table 1. Third, Appendix C.3 specifies our settings for comparing the training convergence speed of different optimization objectives. Algorithm 1 illustrates the training procedure of EP-GFlowNets. The computer code for reproducing our experiments will be publicly released at github.com/ML-FGV/ep-gflownets.

C.1. Experimental setup

In the following, we applied the same optimization settings for each environment. For the stochastic optimization, we minimized the contrastive balance objective using the AdamW optimizer (Loshchilov & Hutter, 2019) for both local and global GFlowNets. We trained the models for 5000 epochs (20000 for the grid world) with a learning rate equal to $3 \cdot 10^{-3}$

Algorithm 1 Training of EP-GFlowNets

Require: $\left(p_F^{(1)}, p_B^{(1)}\right), \ldots, \left(p_F^{(K)}, p_B^{(K)}\right)$ clients' policies, $R_1, \ldots, R_K$ clients' rewards, (p_F, p_B) parameterized global policies, E number of epochs for training, u_F uniform policy
Ensure: $p^{\intercal}(x) \propto R(x) \coloneqq \prod_{1 \leq k \leq K} R_k(x)$
  parfor $k \in \{1, \ldots, K\}$ **do** ▷ Train the clients' models in parallel
    train the policies $\left(p_F^{(k)}, p_B^{(k)}\right)$ to sample proportionally to R_k
  end parfor
  for $e \in \{1, \ldots, E\}$ **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
    $L \leftarrow \frac{1}{|\mathcal{B}|} \sum_{\tau, \tau' \in \mathcal{B}} \mathcal{L}_{AB}\left(\tau, \tau'; \left\{\left(p_F^{(1)}, p_B^{(1)}\right), \ldots, \left(p_F^{(K)}, p_B^{(K)}\right)\right\}\right)$
    Update the parameters of p_F and p_B through gradient descent on L
  end for

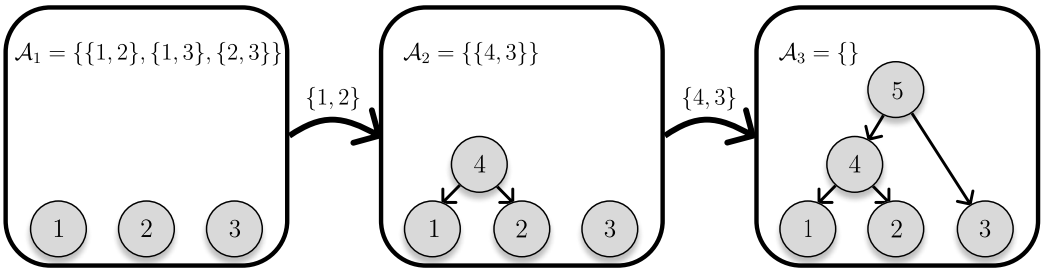

Figure 8. **An illustration of the generative process for phylogenetic trees' topologies.** We iteratively select two trees to join their roots. The final state corresponds to a single, connected graph.

with a batch size dependent upon the environment. Correspondingly, we define the L_1 error between the distributions π and $\hat{\pi}$ as two times the total variation distance between them, $\|\pi - \hat{\pi}\|_1 \coloneqq \sum_{x \in \mathcal{X}} |\pi(x) - \hat{\pi}(x)|$. For the grid world, design of sequences and federated BNSL setups, all intermediate GFlowNet states are also terminal, since they are connected to a sink state. For the remaining setups, most states are not terminal and exist solely as intermediate steps between the initial state and the target distribution's support.

Grid world. We considered a two-dimensional grid with length size 9 as the environment for the results of both Table 1 and Figure 2. To parameterize the forward policy, we used an MLP with two 64-dimensional layers and a LeakyReLU activation function between them (Maas et al., 2013). For inference, we simulated 10^6 environments to (i) compute the L_1 error between the targeted and the learned distributions; and (ii) selected the 800 most rewarding samples. We utilized a batch size equal to 1024 during both the training and inference phases.

Design of sequences. We trained the GFlowNets to generate sequences of size up to 6 with elements selected from a set of size 6. We parametrized the forward policies with a single 64-dimensional layer bidirectional LSTM network followed by an MLP with two 64-dimensional layers (Graves & Graves, 2012). For training, we used a batch size of 512. For inference, we increased the batch size to 1024 and we sampled 10^6 sequences to estimate the quantities reported in Table 1 and Figure 4.

Multiset generation. We designed the GFlowNet to generate multisets of size 8 by iteratively selecting elements from a set U of size 10. Moreover, we endowed each element within U with a learnable and randomly initialized 10-dimensional embedding. To estimate the transition probabilities at a given state s, we applied an MLP with two 64-dimensional layers to the sum of the embeddings of the elements in s. During training, we used a batch size of 512 to parallely generate multiple multisets and reduce the noiseness of the backpropagated gradients. During inference, we increased the batch size to 1024 and generated 10^6 samples to generate the results reported in Table 1 and Figure 3.

Bayesian phylogenetic inference. We devised a GFlowNet to learn a posterior distribution over the space of rooted phylogenetic trees with 7 leaves and fixed branch lengths. Each state is represented as a forest. Initially, each leaf belongs to a

different singleton tree. An action consists of picking two trees and joining their roots to a newly added node. The generative process is finished when all nodes are connected in a single tree (see Figure 8; a similar modeling was recently considered by (Zhou et al., 2023)).To estimate the policies at the (possibly partially built) tree t, we used a graph isomorphism network (GIN; Xu et al., 2019) with two 64-dimensional layers to generate node-level representations for t and then used an MLP to project the sum of these representations to a probability distribution over the viable transitions at t. We used a tempered version of the likelihood to increase the sparsity of the targeted posterior. Importantly, we selected a batch size of 512 for training and of 1024 for inference. Results for Table 1 and Figure 5 are estimates based on 10^5 trees drawn from the learned distributions. To evaluate the likelihood function, we considered Jukes & Cantor's nucleotide substitution model for the observed sites (Jukes & Cantor, 1969), which assigns the same instantaneous substitution rate for each pair of nucleotides.

Bayesian network structure learning. Each client trains a DAG-GFlowNet Deleu et al. (2022) to sample proportionally to the reward function evaluated at a small dataset of 20 points. However, in contrast Deleu et al. (2022), which implemented a linear transformer for parameterizing GFlowNet's forward policy, we use a 2-layer 128-dimensional MLP; the backward policy is fixed as an uniform distribution over a state's parents. For the global model, we increase the MLP's latent layers' size to 256 units each. To estimate the probability $\mathbb{P}_G[U \to V]$, we sample $\{G_1, \ldots, G_N\}$ and compute

$$\frac{1}{N} \sum_{1 \leq n \leq N} \mathbb{1}[U \to V | G_n], \tag{70}$$

with $\mathbb{1}[U \to V | G_n]$ indicating whether the edge $U \to V$ exists in G_n. We compare this quantity with the nominal value obtained by enumerating the target distribution π support and directly evaluating the expectation $\mathbb{E}_{G \sim \pi}[\mathbb{1}[U \to V | G]]$, similarly to (Deleu et al., 2022).

Our implementations were based on `PyTorch` (Paszke et al., 2019) and on `PyTorch Geometric` (Fey & Lenssen, 2019).

C.2. Parallel Categorical Variational Inference

As a simplistic approach to combining the locally learned distributions over compositional objects, we variationally approximate them as the product of categorical distributions over the objects' components. For this, we select the parameters that minimize the reverse Kullback-Leibler divergence between the GFlowNet's distribution p_T and the variational family $\mathcal{Q}$,

$$\hat{q} = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}[p_T || q] = \arg\min_{q \in \mathcal{Q}} -\mathbb{E}_{x \sim p_T}[\log q(x)], \tag{71}$$

which, in asymptotic terms, is equivalent to choosing the parameters that maximize the likelihood of the GFlowNet's samples under the variational model. Then, we use a logarithmic pool of these local variational approximations as a proxy for the global model. In the next paragraphs, we present the specific instantiations of this method for the domains we considered throughout our experiments. We used the same experimental setup of Appendix C.1 to train the local GFlowNets.

Grid world. An object in this domain is composed of its two coordinates in the grid. For a grid of width W and height H, we consider the variational family

$$\mathcal{Q} = \{(\phi, \psi) \in \Delta^{W+1} \times \Delta^{H+1} \colon q_{\phi,\psi}(x, y) = \mathrm{Cat}(x|\phi)\mathrm{Cat}(y|\psi)\}, \tag{72}$$

in which Δ^d is the d-dimensional simplex and $\mathrm{Cat}(\phi)$ ($\mathrm{Cat}(\psi)$) is a categorical distribution over $\{0, \ldots, W\}$ ($\{0, \ldots, H\}$) parameterized by ϕ (ψ). Then, given the N variational approximations $\left(q_{\phi^{(1)},\psi^{(1)}}\right), \ldots, \left(q_{\phi^{(N)},\psi^{(N)}}\right)$ individually adjusted to the distributions learned by the local GFlowNets, we estimate the unnormalized parameters $\tilde{\phi}$ and $\tilde{\psi}$ of the variational approximation to the global distribution over the positions within the grid as

$$\tilde{\phi} = \bigodot_{1 \leq i \leq N} \phi^{(i)} \text{ and } \tilde{\psi} = \bigodot_{1 \leq i \leq N} \psi^{(i)}. \tag{73}$$

Then, we let $\phi = {}^{\phi_u}/_{\phi_u^\intercal \mathbf{1}_{W+1}}$ and $\psi = {}^{\psi_u}/_{\psi_u^\intercal \mathbf{1}_{H+1}}$, with $\mathbf{1}_d$ as the d-dimensional vector of 1s, be the parameters of the global model.

Design of sequences. We represent sequences of size up to T over a dictionary V as a tuple $(S, (x_1, \ldots, x_S))$ denoting its size S and the particular arrangement of its elements $(x_1, \ldots, x_S)$. This is inherently modeled as a hierarchical model of categorical distributions,

$$S \sim \text{Cat}(\theta), \tag{74}$$

$$x_i \sim \text{Cat}(\phi_{i,S}|S) \text{ for } i \in \{1, \ldots, S\}, \tag{75}$$

which is parameterized by $\theta \in \Delta^T$ and $\phi_{\cdot,S} \in \mathbb{R}^{S\times|V|}$ for $S \in \{1, \ldots, T\}$. We define our family of variational approximations as the collection of all such hierarchical models and estimate the parameters θ and ϕ accordingly to Equation (71). In this case, let $\left(\theta^{(1)}, \phi^{(1)}\right), \ldots, \left(\theta^{(N)}, \phi^{(N)}\right)$ be the parameters associated with the variational approximations to each of the N locally trained GFlowNets. The unnormalized parameters $\tilde{\theta}$ and $\tilde{\phi}$ of the combined model that approximates the global distribution over the space of sequences are then

$$\tilde{\theta} = \bigodot_{1\leq i\leq N} \theta^{(i)} \text{ and } \tilde{\phi}_{\cdot,S} = \bigodot_{1\leq i\leq N} \phi^{(i)}_{\cdot,S} \text{ for } S \in \{1, \ldots, T\}, \tag{76}$$

whereas the normalized ones are $\theta = \tilde{\theta}/\tilde{\theta}^\intercal \mathbf{1}_T$ and $\phi_{\cdot,S} = \text{diag}(\tilde{\phi}_{\cdot,S}\mathbf{1}_{|V|})^{-1}\tilde{\phi}_{\cdot,S}$.

Multiset generation. We model a multiset $\mathcal{S}$ of size S as a collection of independently sampled elements from a warehouse $\mathcal{W}$ with replacement. This characterizes the variational family

$$\mathcal{Q} = \left\{ q(\cdot|\phi) \colon q(\mathcal{S}|\phi) = \prod_{s\in\mathcal{S}} \text{Cat}(s|\phi) \right\}, \tag{77}$$

in which ϕ is the parameter of the categorical distribution over $\mathcal{W}$ estimated through Equation (71). Denote by $\phi^{(1)}, \ldots, \phi^{(N)}$ the estimated parameters that disjointly approximate the distribution of N locally trained GFlowNets. We then variationally approximate the logarithmically pooled global distribution as $q(\cdot|\phi) \in \mathcal{Q}$ with $\phi = \tilde{\phi}/\tilde{\phi}^\intercal \mathbf{1}_{|\mathcal{W}|}$, in which

$$\tilde{\phi} = \bigodot_{1\leq i\leq N} \phi^{(i)}. \tag{78}$$

Notably, the best known methods for carrying out Bayesian inference over the space of phylogenetic trees are either based on Bayesian networks (Zhang & Matsen IV, 2018) or MCMC, neither of which are amenable to data parallelization and decentralized distributional approximations without specifically tailored heuristics. More precisely, the product of Bayesian networks may not be efficiently representable as a Bayesian network, and it is usually not possible to build a global Markov chain whose stationary distribution matches the product of the stationary distributions of local Markov chains. Moreover, any categorical variational approximation factorizable over the trees' clades would not be correctly supported on the space of complete binary trees and would lead to frequently sampled invalid graphs. By a similar reasoning, we do not compare EP-GFlowNets to a composition of tractable variational approximations to the local models, such as (Geffner et al., 2022; Zheng et al., 2018), in the problem of federated Bayesian structure learning.

C.3. Comparison of different training criteria

Experimental setup. We considered the same environments and used the same neural network architectures described in Appendix C.1 to parametrize the transition policies of the GFlowNets. Importantly, the implementation of the DB constraint and of the FL-GFlowNet requires the choice of a parametrization for the state flows (Bengio et al., 2023; Pan et al., 2023a). We model them as an neural network with an architecture that essentially mirrors that of the transition policies — with the only difference being the output dimension, which we set to one. Moreover, we followed suggestions in (Pan et al., 2023a; Malkin et al., 2022) and utilized a learning rate of $3 \cdot 10^{-3}$ for all parameters of the policy networks except for the partition function's logarithm $\log Z$ composing the TB constraint, for which we used a learning rate of $1 \cdot 10^{-1}$. Noticeably, we found that this heterogeneous learning rate scheme is crucial to enable the training convergence under the TB constraint.

Further remarks regarding Figure 7. In Figure 7, we observed that $\mathcal{L}_{CB}$ and $\mathcal{L}_{TB}$ perform similarly in the grid world and in design of sequences tasks. A reasonable explanation for this is that such criteria are identically parameterized in such domains, as $\mathcal{L}_{DB}$ reduces to $R(s')p_B(s|s')p_F(s_f|s) = R(s)p_F(s'|s)p_F(s_f|s')$ in environments where every state is terminal (Deleu et al., 2022). Thus, F vanishes and hence the difficult estimation of this function is avoided.

Table 2. **Quality of the parallel approximation**. The global model's performance does not critically depend on the clients' training objective; it relies only on the goodness-of-fit of their models.

	Grid World		Multisets		Sequences	
	$L_1 \downarrow$	Top-800 $\uparrow$	$L_1 \downarrow$	Top-800 $\uparrow$	$L_1 \downarrow$	Top-800 $\uparrow$
EP-GFlowNet (CB)	0.038 ($\pm$0.016)	-6.355 ($\pm$0.000)	0.130 ($\pm$0.004)	27.422 ($\pm$0.000)	0.005 ($\pm$0.002)	-1.535 ($\pm$0.000)
EP-GFlowNets (TB)	0.039 ($\pm$0.006)	-6.355 ($\pm$0.000)	0.131 ($\pm$0.018)	27.422 ($\pm$0.000)	0.006 ($\pm$0.005)	-1.535 ($\pm$0.000)

C.4. Additional experiments

Reduction in runtime achieved due to a distributed formulation. Figure 12 shows that our distributed inference framework enacts a significant reduction in runtime in the task of Bayesian phylogenetic inference relatively to a centralized approach — while maintaining a competitive performance under the L_1 metric[1]. This underlines the effectiveness of our method for Bayesian inference when the target distribution is computationally difficult to evaluate due to an expensive-to-compute likelihood function (which, for phylogenies, requires running Felsenstein's dynamic programming algorithm (Felsenstein, 1981) to carry out message-passing in a graphical model). In this context, EP-GFlowNets may be also of use in Bayesian language modeling with large language models (LLMs), in which case the likelihood function is provided by a LLM with a notoriously high computational footprint; see (Hu et al., 2023a). For this experiment, we considered the same setup described in Appendix C.1, partitioning the data among an increasing number of $\{2, 4, 6, 8, 10\}$ identical clients, each processing a set of 1000 random sites, and training the corresponding centralized and distributed models to sample from the full posterior. The reported times were measured in a high-performance Linux machine equipped with an AMD EPYC 7V13 64-core processor, 216 GB DDR4 RAM and a NVIDIA™ A100 80 GB PCIe 4.0 GPU. Notably, we found out that training a GFlowNet in a CPU is considerably faster in this case than training it in a GPU — possibly due to the iterative nature of the generative process and the relatively small size of the policy networks, whose accelerated evaluation in a GPU does not compensate for the corresponding slow-down of the tree-building process.

Comparison between TB and CB with different learning rates. Figure 9 shows that increasing the learning rate for $\log Z_{\phi_Z}$ significantly accelerates the training convergence for the TB objective. In this experiment, the learning rate for the other parameters was fixed at 10^{-3} — following the setup of Malkin et al. (2022, Appendix B). However, CB leads to faster convergence relatively to TB for all considered learning rates. In practice, though, note that finding an adequate learning rate for $\log Z_{\phi_Z}$ may be a very difficult and computationally exhaustive endeavor that is completely avoided by implementing the CB loss.

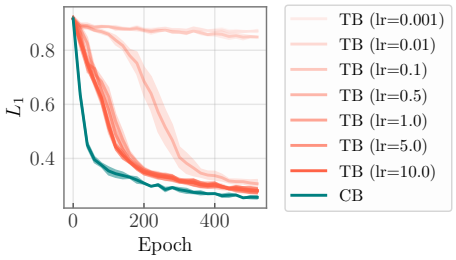

Figure 9. CB outperforms TB for different *lr*'s for $\log Z$.

Sampling from product of distributions. As a first approximation, one could employ a simple rejection sampling procedure to sample from a product of discrete distributions $q_1, \ldots, q_K$ over a shared support: draw independently a x_i from q_i and accept the resulting x_i only if it was sampled by every other model. Nonetheless, this approach scales very badly as one increases either the number of distributions or the size of their common support. Hinton (2002), in a related work, proposed an efficient algorithm to train a model to sample from a product of energy-based distributions by minimizing a contrastive divergence, acknowledging that it is not generally straightforward to sample from a product of discrete distributions. More recently, Du et al. (2023) developed

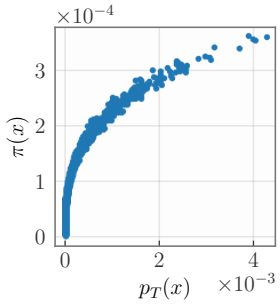

Figure 10. Sampling from the product of GFlowNets' policies isn't equivalent to sampling from the local targets' product.

[1]Note that, in Figure 12, the runtime for EP-GFlowNet equals the maximum time for training the clients' models plus the running time of our aggregation step.

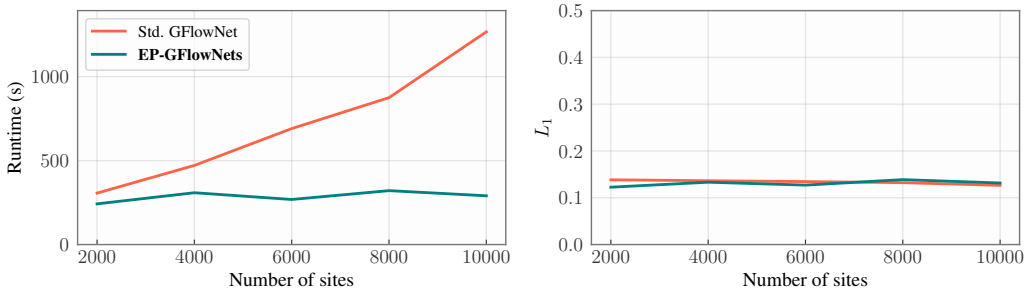

Figure 12. **EP-GFlowNets achieve a significant reduction in runtime relatively to a standard GFlowNet** in the task of Bayesian phylogenetic inference. The left plot highligths that the training time for an EP-GFlowNet remains approximately constant when an increasing set of observed samples is equally partitioned between a correspondingly increasing set of clients, each receiving 1000 sites, whereas the training time of a standard GFlowNet grows roughly linearly on the number of samples. Importantly, the right plot shows the asynchronously trained model performs comparably to the synchronously trained one for all the considered data sizes.

an MCMC-based algorithm to sample from a multiplicative composition of energy-base parameterized diffusion models, remarking the incorrectness of naive approaches based on sampling from the product of the reverse kernels. In the context of compounding GFlowNets with forward policies $(p_F^{(i)})_{1\leq i\leq N}$ to sample from the corresponding product distribution $\prod_{1\leq i\leq N} R_i$, however, one could exploit the structured nature of the model and naively attempt to use

$$p_F(\cdot|s) \propto \prod_{1\leq i\leq N} p_F^{(i)}(\cdot|s), \tag{79}$$

hoping that the resulting GFlowNet would sample from the correct product distribution. Notably, this approach fails due to the dependence of the normalizing constant of the preceding distribution on the state s — and it is unclear according to what distribution the sampled objects are distributed. Figure 10 illustrates this for the problem of generating sequences: by combining the policies of the clients accordingly to Equation (79), the culminating distribution drastically differs from the targeted one, even though the clients were almost perfectly trained. The question of which distributions one may obtain by composing the policies of expensively pre-trained GFlowNets is still open in the literature (Garipov et al., 2023).

Implementing different training objectives for the clients. Table 2 suggests that the accuracy of EP-GFlowNet's distributional approximation is mostly independent of whether the clients implemented CB or TB as training objectives. Notably, the combination phase of our algorithm is designedly agnostic to how the local models were trained — as long as they provide us with well-trained backward and forward policies. This is not constraining, however, since any practically useful training scheme for GFlowNets is explicitly based upon the estimation of such policies (Malkin et al., 2022; Pan et al., 2023a; Bengio et al., 2023; Zhang et al., 2023c).

Aggregation phase' computational cost. Figure 11 shows that the per-epoch cost of training of a naively implemented EP-GFlowNets' aggregation phase, which we measure for the task of multiset generation, increases roughly linearly as a function of the number of worker nodes. Nonetheless, we remark that, in theory, one could achieve a sublinear relationship between the computational cost and the number of clients by parallelizing the local policies' evaluation when minimizing the aggregating balance loss function $\mathcal{L}_{AB}$, which is not done by our implementation.

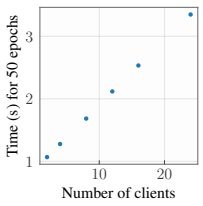

Figure 11. Aggregation phase's 50-epoch cost.

Sensibility of EP-GFlowNet to imperfectly trained clients. Theorem 3.4 ensures that EP-GFlowNets can be trained to accurately sample from the combined target when the clients are sufficiently — but maybe imperfectly — learned. However, even when clients are imperfectly trained, EP-GFlowNets may achieve a relatively good approximation to the target. To exemplify this, we consider the task of multiset generation with 4 clients trained on multiplicatively noisy targets, namely, $\log R_i'(x) = \log R_i(x) + \epsilon$ with $\epsilon \sim \mathcal{N}_{|\mathcal{X}|}(\mathbf{0}, \sigma^2\mathbf{I})$ sampled from a $|\mathcal{X}|$-dimensional normal distribution. Then, we perform EP-GFlowNets' aggregation phase and compare the learned distribution with the product $\prod_{1\leq i\leq 4} R_i$. Importantly, Figure 13 shows that the accuracy of the aggregated model is comparable and often better than that of the most inaccurate client for $\sigma^2 \in \{0.000, 0.002, 0.004, 0.006, 0.008, 0.01\}$.

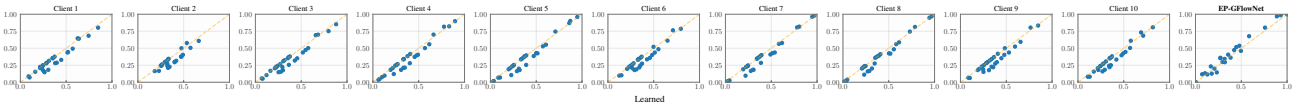

Figure 14. **Federated Bayesian network structure learning.** Each plot shows, for each pair of nodes (U, V), the expected (vertical) and learned (horizontal) probabilities of U and V being connected by an edge, $\mathbb{P}[U \to V]$, and of existing a directed path between U and V, $\mathbb{P}[U \rightsquigarrow V]$. Notably, EP-GFlowNet accurately matches the target distribution over such edges' features.

D. Federated Bayesian Network Structure Learning (BNSL)

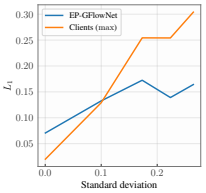

Figure 13. EP-GFlowNet's vs Clients' accuracy.

Bayesian networks are often used to describe the statistical dependencies of a set of observed variables (Bielby & Hauser, 1977; Pearl & Mackenzie, 2018; Pearl, 2009; 1988); however, learning their structure from data is very challenging due to the combinatorially many possibilities. For example, when such variables represent some form of gene expression, we may want to infer the gene regulatory network describing the genes' causal relationships (Husmeier, 2003). Conventionally, structure learning is carried out centrally — with all data gathered in a single place (Reisach et al., 2021; Lorch et al., 2021). Nonetheless, such datasets are often small and insufficiently informative about the underlying structure. On the other hand, the rapid development of information technologies has significantly lowered the barrier for multiple parties (e.g., companies) to collaboratively train a model using their collected and privacy-sensitive datasets. In this context, we extend the work of Deleu et al. (2022; 2023) and show that EP-GFlowNets can be efficiently used to learn a belief distribution over Bayesian networks in a federated setting (Ng & Zhang, 2022).

For this, let $\mathbf{X} \in \mathbb{R}^d$ be a random variable and $\mathbf{B} \in \mathbb{R}^{d \times d}$ be a sparse real-valued matrix with sparsity pattern determined by a directed acyclic graph $\mathcal{G}$ with nodes $\{1, \dots, d\}$. We assume that $\mathbf{X}$ follows a linear *structural equation model* (SEM)

$$\mathbf{X} = \mathbf{B}\mathbf{X} + \mathbf{N}, \tag{80}$$

with $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, \sigma^2 I)$ representing independent Gaussian noise (Bielby & Hauser, 1977; Deleu et al., 2022). Also, we consider K fixed clients, each owning a private dataset $\mathcal{D}_k = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, and our objective is to learn a belief distribution

$$R(G) = \prod_{1 \leq k \leq K} R_k(G) \tag{81}$$

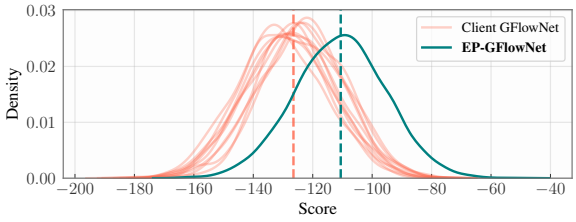

Figure 15. The distribution learned in a federated manner (green) finds significantly higher scoring graphs wrt the complete data than any of its local counterparts (red). The dashed lines represent the avg. scores for the global and local models.

over DAGs, in which $R_k(G) = p(\mathcal{D}_k|G)p(G)$ and $p(\mathcal{D}_k|G) = p(\mathcal{D}_k|\hat{\mathbf{B}}_k, G)$, with $\hat{\mathbf{B}}_k$ being client k's maximum likelihood estimate of $\mathbf{B}$ given G (Deleu et al., 2022). Intuitively, the belief R assigns high probability to DAGs which are probable under each of R_k's and may be seem as a multiplicative composition of the local models (Garipov et al., 2023; Du et al., 2023). Notably, Ng & Zhang (2022) considered a similar setup, iteratively solving a L_1-regularized continuous relaxation of the structure learning problem to find a globally optimum DAG (Zheng et al., 2018). However, in contrast to EP-GFlowNets, Ng & Zhang (2022)'s approach doesn't always guarantee that the obtained graph is acyclic (Geffner et al., 2022; Zheng et al., 2018); it doesn't provide a belief distribution over the DAGs, which could be successively refined by probing an expert (Bharti et al., 2022; da Silva et al., 2023); and it is not embarrassingly parallel, requiring many communication steps to be fulfilled.

Experimental setup. To show that EP-GFlowNets can accurately sample from the belief distribution defined by Equation (81) in a distributed setting, we first simulate a dataset of 400 independent data points according to the SEM outlined in Equation (80) with $d = 4$ variables. Next, we evenly partition this dataset between 4 clients, which then train their own DAG-GFlowNets (Deleu et al., 2022) based on their individual datasets for 10000 epochs using AdamW (Loshchilov & Hutter, 2019). Finally, the locally trained DAG-GFlowNets are aggregated by minimizing the aggregating balance and the resulting model is compared to the belief distribution in Equation (81). See Section 4.5 for further discussion on our empirical observations for this domain.

Results. Figure 14 s that EP-GFlowNet adequately learns a distribution over DAGs in a federated setting. Correspondingly, Figure 15 stresses that the collaboratively trained GFlowNet finds graphs with higher scores (graph-conditioned maximum log-likelihood) relatively to the private models. See Section 4.5 for discussion.

E. Related work

GFlowNets were originally proposed as a reinforcement learning algorithm tailored to the search of diverse and highly valuable states within a given discrete environment (Bengio et al., 2021). Recently, these algorithms were successfully applied to the discovery of biological sequences (Jain et al., 2022), robust scheduling of operations in computation graphs (Zhang et al., 2023c), Bayesian structure learning and causal discovery (Deleu et al., 2022; 2023; da Silva et al., 2023; Atanackovic et al., 2023a), combinatorial optimization (Zhang et al., 2023a), active learning (Hernandez-Garcia et al., 2023), multi-objective optimization (Jain et al., 2023), and discrete probabilistic modeling (Zhang et al., 2022a; Hu et al., 2023b; Zhang et al., 2022b). (Bengio et al., 2023) formulated the theoretical foundations of GFlowNets. Correlatively, (Lahlou et al., 2023) laid out the theory of GFlowNets defined on environments with a non-countable state space. (Pan et al., 2023c) and (Zhang et al., 2023b) extended GFlowNets to environments with stochastic transitions and rewards. Concomitantly to these advances, there is a growing literature that aims to better understand and improve this class of algorithms (Deleu & Bengio, 2023; Shen et al., 2023; Malkin et al., 2023), with an emphasis on the development of effective objectives and parametrizations to accelerate training convergence (Pan et al., 2023b;a; Malkin et al., 2022; Deleu et al., 2022). In recent work, for instance, (Pan et al., 2023a) proposed a novel residual parametrization of the state flows that achieved promising results in terms of speeding up the training convergence of GFlowNets. More specifically, the authors assumed the existence of a function $\mathcal{E}\colon \mathcal{S} \to \mathbb{R}$ such that (i) $\mathcal{E}(s_o) = 0$ and (ii) $\mathcal{E}(x) = -\log R(x)$ for each terminal state $x \in \mathcal{X}$ and reparameterized the state flows as $\log F(s, \phi_S) = -\mathcal{E}(s) + \log \tilde{F}(s, \phi_S)$. This new training scheme was named *forward looking (FL) GFlowNets* due to the inclusion of partially computed rewards in non-terminal transitions. Notably, both Malkin et al. (2023) and (Zhang et al., 2023c) proposed using the variance of the a TB-based estimate of the log partition function as a training objective based on the variance reduction method of Richter et al. (2020). It is important to note one may use stochastic rewards (see Bengio et al., 2023; Zhang et al., 2023b) for carrying out distributed inference, in the same fashion of, e.g., distributed stochastic-gradient MCMC (El Mekkaoui et al., 2021; Vono et al., 2022). Notably, stochastic rewards have also been used in the context of causal structure learning by Deleu et al. (2023). However, it would require many communication steps between clients and server to achieve convergence — which is one of the bottleneck EP-GFlowNets aim to avoid.

Distributed Bayesian inference mainly concerns the task of approximating or sampling from a posterior distribution given that data shards are spread across different machines. This comprises both federated scenarios (El Mekkaoui et al., 2021; Vono et al., 2022) or the ones in which we arbitrarily split data to speed up inference (Scott et al., 2016). Within this realm, there is a notable family of algorithms under the label of *embarrassingly parallel MCMC* (Neiswanger et al., 2014), which employ a divide-and-conquer strategy to assess the posterior. These methods sample from subposteriors (defined on each user's data) in parallel, subsequently sending results to the server for aggregation. The usual approach is to use local samples to approximate the subposteriors with some tractable form and then aggregate the approximations in a product. In this line, works vary mostly in the approximations employed. For instance, Mesquita et al. (2019) apply normalizing flows, (Nemeth & Sherlock, 2018) model the subposteriors using Gaussian processes, and (Wang et al., 2015) use hyper-histograms. It is important to note, however, that these works are mostly geared towards posteriors over continuous random variables.

Federated learning was originally motivated by the need to train machine learning models on privacy-sensitive data scattered across multiple mobile devices — linked by an unreliable communication network (McMahan et al., 2017b). While we are the first tackling FL of GFlowNets, there are works on learning other generative models in federated/distributed settings, such as for generative adversarial networks (Hong et al., 2021; Chang et al., 2020; Qu et al., 2020) and variational autoencoders (Polato, 2021). Critically, EP-GFlowNets enable the collaborative training of a global model without disclosing the data underlying the clients' reward functions (defined by a neural network (Jain et al., 2022) or a posterior distribution (Deleu et al., 2022; 2023; da Silva et al., 2023), for instance); however, it does not inherently preserve the privacy of the reward functions themselves, which may be accurately estimated from their publicly shared policy networks. Nevertheless, a future work may investigate formally to which extent imperfectly trained policy networks ensure some form of differential privacy over the clients' rewards.

Comparing EP-GFlowNets to single-round FedAVG (McMahan et al., 2017a) As EP-GFlowNets are the first method enabling embarrassingly parallel inference over discrete combinatorial spaces, there are no natural baselines to our

experiments. For completeness, however, we use a single round of FedAVG (McMahan et al., 2017a) by training local models (GFlowNets) until convergence and aggregating them in parameter space, assuming all policy networks share the same architecture. Note, however, that this baseline does not enjoy any guarantee of correctness and may lead to poor results. To validate this, we have run an experiment for the multiset generation task using FedAvg for aggregation. The resulting model attained an L_1 error of 1.32, roughly two orders of magnitude worse compared to the 0.04 of EP-GFlowNets.