A PROOFS

A.1 PROOF OF LEMMA 1

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

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

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

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 1

The proof is based on the following reasoning. We first show that, given the satisfiability of the federated 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 1. 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 federated 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 subsection A.1, notice that the contrastive nature of the federated 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

$$\begin{aligned}p_T(x)&\coloneqq\sum_{\tau\rightsquigarrow x}\prod_{s\to s'\in\tau}p_F(s\to s')&(25)\\&=c\sum_{\tau\rightsquigarrow x}\left(\prod_{1\leq i\leq N}\frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right)p_B(\tau|x)&(26)\\&=c\mathbb{E}_{\tau\sim p_B(\cdot|x)}\left[\prod_{1\leq i\leq N}\frac{p_F^{(i)}(\tau)}{p_B^{(i)}(\tau|x)}\right].&(27)\end{aligned}$$

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 federated balance condition for the distributional correctness of the global model.

A.3 Proof of Theorem 2

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 federated balance condition (see Remark 1 and subsection 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

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

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

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

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

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',

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

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

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

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

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 EXPONENTIALLY WEIGHTED DISTRIBUTIONS

This section extends our theoretical results and shows how to train a FC-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 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.

Algorithm 1 Training of Federated 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
    $L \leftarrow \frac{1}{|\mathcal{B}|} \sum_{\tau, \tau' \in \mathcal{B}} \mathcal{L}_{FB}\left(\tau, \tau'; \left\{\left(p_F^{(1)}, p_B^{(1)}\right), \ldots, \left(p_F^{(K)}, p_B^{(K)}\right)\right\}\right)$
    Update the parameters of p_F and p_B through gradient descent on L
  end for

Theorem 1′ (Federated 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* $R(x) \coloneqq \prod_{n=1}^N R(x)^{\omega_n}$ *if and only if the following condition holds for any terminal trajectories* $\tau, \tau' \in \mathcal{T}$*:*

$$\prod_{1 \leq i \leq N} \frac{\left(\prod_{s \to s' \in \tau} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)^{\omega_i}}{\left(\prod_{s \to s' \in \tau'} \frac{p_F^{(i)}(s,s')}{p_B^{(i)}(s',s)}\right)^{\omega_i}} = \frac{\left(\prod_{s \to s' \in \tau} \frac{p_F(s,s')}{p_B(s',s)}\right)}{\left(\prod_{s \to s' \in \tau'} \frac{p_F(s,s')}{p_B(s',s)}\right)}. \tag{58}$$

Secondly, Theorem 2′ 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 2′ (Influence of local failures)**.** *Let* $\pi_n \coloneqq R_n / Z_n$ *and* $p_F^{(n)}$ *and* $p_B^{(n)}$ *be the forward and backward policies of the* n*th client. 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*

$$1 - \alpha_n \leq \min_{x \in \mathcal{X}, \tau \rightsquigarrow x} \frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)\pi_n(x)} \leq \max_{x \in \mathcal{X}, \tau \rightsquigarrow x} \frac{p_F^{(n)}(\tau)}{p_B^{(n)}(\tau|x)\pi_n(x)} \leq 1 + \beta_n \tag{59}$$

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 federated balance condition in Equation 4 and* $\pi(x) \propto \prod_{n=1}^N \pi_n(x)^{\omega_n}$ *then satisfies*

$$\mathcal{D}_J(\pi, \hat{\pi}) \leq \sum_{n=1}^N \omega_n \log\left(\frac{1 + \beta_n}{1 - \alpha_n}\right). \tag{60}$$

Interestingly, one could train a *conditional* GFlowNet (Bengio et al., 2021) 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 FC-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 Federated GFlowNets.

Reviewer: wWFS

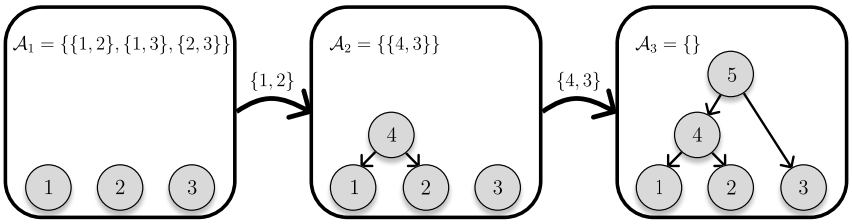

Figure 7: **An illustration of the generative process for phylogenetic trees' topologies.** We iteratively select two trees to join their roots. The final state corresponds to a single, connected graph.

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 and 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 $3 \cdot 10^{-3}$ 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 and design of sequences setups, all intermediate GFlowNet states are also terminal, since they lie on the path from the initial state to another terminal state. For the remaining setups, the intersection between terminal and intermediate states is empty.

Reviewer: rsSg

Grid world. We considered a two-dimensional grid with length size 12 as the environment for the results of both Table 1 and Figure 2. To parametrize 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) select 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 and 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 7).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.

Reviewer: Xj4h

Our implementations were based on `PyTorch` (Paszke et al., 2019) and on `PyTorch Geometric` (Fey and 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 $\mathcal{Q}$,

$$\hat{q} = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}[p_T || q] = \arg\min_{q \in \mathcal{Q}} -\mathbb{E}_{x \sim p_T}[\log q(x)], \tag{61}$$

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 subsection 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

$$\mathcal{Q} = \{(\phi, \psi) \in \Delta^{W+1} \times \Delta^{H+1} \colon q_{\phi,\psi}(x, y) = \mathrm{Cat}(x|\phi)\mathrm{Cat}(y|\psi)\}, \tag{62}$$

in which Δ^d is the d-dimensional simplex and $\mathrm{Cat}(\phi)$ ($\mathrm{Cat}(\psi)$) is a categorical distribution over $\{0, \ldots, W\}$ ($\{0, \ldots, H\}$) parametrized 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

$$\tilde{\phi} = \bigodot_{1 \le i \le N} \phi^{(i)} \text{ and } \tilde{\psi} = \bigodot_{1 \le i \le N} \psi^{(i)}. \tag{63}$$

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,

$$S \sim \mathrm{Cat}(\theta), \tag{64}$$

$$x_i \sim \mathrm{Cat}(\phi_{i,S}|S) \text{ for } i \in \{1, \ldots, S\}, \tag{65}$$

which is parametrized 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 61. 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

$$\tilde{\theta} = \bigodot_{1 \le i \le N} \theta^{(i)} \text{ and } \tilde{\phi}_{\cdot,S} = \bigodot_{1 \le i \le N} \phi^{(i)}_{\cdot,S} \text{ for } S \in \{1, \ldots, T\}, \tag{66}$$

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

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

$$\mathcal{Q} = \left\{ q(\cdot|\phi) \colon q(\mathcal{S}|\phi) = \prod_{s \in \mathcal{S}} \mathrm{Cat}(s|\phi) \right\}, \tag{67}$$

in which ϕ is the parameter of the categorical distribution over $\mathcal{W}$ estimated through Equation 61. 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

$$\tilde{\phi} = \bigodot_{1 \le i \le N} \phi^{(i)}. \tag{68}$$

Table 2: **Quality of the federated 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 $\uparrow$
FC-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$0.000)
FC-GFlowNets (TB)	0.039 ($\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)

Notably, the best known methods for carrying out Bayesian inference over the space of phylogenetic trees are either based on Bayesian networks (Zhang and Matsen IV, 2018) or MCMC, neither of which are amenable to data parallelization and decentralized distributional approximations. 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.

C.3 Comparison of different training criteria

Experimental setup. We considered the same environments and used the same neural network architectures described in subsection 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 6. In Figure 6, we observed that $\mathcal{L}_{CB}$ and $\mathcal{L}_{TB}$ perform similarly in the grid world and in design of sequences domains. 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.

C.4 Additional experiments

Comparison between TB and CB with different learning rates. Figure 8 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 *lr*'s. 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.

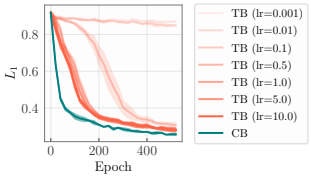

Figure 8: CB outperforms TB for different *lr*'s for $\log Z$.

Implementing different training objectives for the clients. Table 2 suggests that the accuracy of FC-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. (2023a).

Reviewer: Xj4h

D 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., 2023a), Bayesian structure learning and causal discovery (Deleu et al., 2022; 2023; da Silva et al., 2023; Atanackovic et al., 2023), combinatorial optimization (Zhang et al., 2023b), 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., 2023; 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. (2023c) 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 and 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). Notably, both Malkin et al. (2023) and (Zhang et al., 2023a) 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., 2023c) carry out federated 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., 2022) and (Deleu et al., 2023). However, it would require many communication steps between clients and server to achieve convergence — which is precisely the bottleneck FC-GFlowNets aim to avoid.

Reviewer: Xj4h

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 and 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., 2017). 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).