A Theory of Continuous Generative Flow Networks

Salem Lahlou [1 2]   **Tristan Deleu** [1 2]   **Pablo Lemos** [1 3 2]   **Dinghuai Zhang** [1 2]   **Alexandra Volokhova** [1 2]
Alex Hernández-García [1 2]   **Léna Néhale Ezzine** [1 2]   **Yoshua Bengio** [1 2 4]   **Nikolay Malkin** [1 2]

Abstract

Generative flow networks (GFlowNets) are amortized variational inference algorithms that are trained to sample from unnormalized target distributions over compositional objects. A key limitation of GFlowNets until this time has been that they are restricted to discrete spaces. We present a theory for generalized GFlowNets, which encompasses both existing discrete GFlowNets and ones with continuous or hybrid state spaces, and perform experiments with two goals in mind. First, we illustrate critical points of the theory and the importance of various assumptions. Second, we empirically demonstrate how observations about discrete GFlowNets transfer to the continuous case and show strong results compared to non-GFlowNet baselines on several previously studied tasks. This work greatly widens the perspectives for the application of GFlowNets in probabilistic inference and various modeling settings.

1. Introduction

Generative flow networks (GFlowNets; Bengio et al., 2021a) are an increasingly popular class of methods that amortize sampling from intractable distributions over spaces with a compositional structure by learning a sequential sampling policy. Their applications include the design of biological structures such as molecules (Bengio et al., 2021a; Jain et al., 2022), Bayesian structure learning (Deleu et al., 2022; Nishikawa-Toomey et al., 2022), and robust combinatorial optimization (Zhang et al., 2023b). Naturally, their development and theoretical foundations (Bengio et al., 2021b; Malkin et al., 2023; Zimmermann et al., 2022) have been geared towards environments with discrete structures.

As many probabilistic inference and modeling problems involve continuous variables, it is natural to ask whether the advantages of GFlowNets, which include stable off-policy learning and the ability to capture many modes of the target distribution, extend to general spaces. For example, molecule design implies specifying relative spatial positions of atoms and benefits from modeling continuous variables, such as torsion angles (Jing et al., 2022), and Bayesian structure learning requires the discovery of not only the structure of the graphical model, but also its parameters.

As a first attempt at using GFlowNet losses to train an amortized sampler of a unnormalized continuous density, Malkin et al. (2023) showed that the off-policy benefits of GFlowNets extend to a toy stochastic control problem. However, a theory justifying the soundness GFlowNet losses in domains with continuous actions has been still lacking. More recently, Li et al. (2023) presented an extension of the flow-matching conditions (Bengio et al., 2021a) to continuous domains; however, this extension relies upon several invalid assumptions, as we expand on in §3.1.

This paper presents a theory extending all known GFlowNet training objectives to arbitrary spaces. It relies on measurable pointed graphs, a generalization of directed acyclic graphs (DAGs) to measurable spaces, based on Markov kernels. Our main **theoretical contributions** are an extension of the flow-matching (FM; Bengio et al., 2021a), detailed balance (DB; Bengio et al., 2021b), and trajectory balance (TB; Malkin et al., 2022) conditions and a theorem proving that the learned *forward kernel* samples from the target distribution when any of these conditions is satisfied. These conditions lead to training losses involving density functions and allowing gradient-based learning. Existing losses for discrete GFlowNets are special cases of the ones we state.

Additionally, we provide **experimental results** in multiple domains with different structures, some of which include action spaces with both discrete and continuous components. These experiments serve both to validate the theoretical claims and to inform practitioners of caveats that are specific to continuous domains. Our comparative experiments confirm that the already-proven advantages of discrete GFlowNets transfer to more general state spaces.

The remainder of the paper is structured as follows:

§2 reviews GFlowNets and work on stochastic sampling;

§3 presents the theoretical results and a practical summary;

---

[1]Mila [2]Université de Montréal [3]Ciela Institute [4]CIFAR. Correspondence to: Salem Lahlou <lahlosal@mila.quebec>.

§4 is devoted to empirically validating the theory and comparing generalized GFlowNets with baselines.

2. Background and related work

2.1. Stochastic sampling in continuous spaces

Sequential sampling in continuous spaces has a long history. Specialized Markov chain Monte Carlo (MCMC) methods exist for sampling from continuous or differentiable densities, such as Langevin and Hamiltonian MCMC (Coffey et al., 2004; Neal, 2012; Hoffman & Gelman, 2011).

Another line of work considers stochastic sampling in a finite number of steps. The family of sequential Monte Carlo methods (Doucet et al., 2001) and the closely related annealed importance sampling (Neal, 2001) specify a sequence of intermediate target densities with respect to which samplers aim to approximately satisfy detailed balance, but the transition kernel is typically not a learned neural network policy. More recently, learnable-kernel sampling methods, formulated as score-based or stochastic differential equation modeling, have been used for maximum-likelihood generative modeling (e.g., Sohl-Dickstein et al., 2015; Song & Ermon, 2019; Ho et al., 2020; Dockhorn et al., 2022), as well as for learning to sample from an intractable target density (Zhang & Chen, 2022). As we show in our experiments, such algorithms can be seen as special cases of GFlowNets where the state space is of a particular form (§4.2, §4.5).

Another related direction is stochastic normalizing flows (Wu et al., 2020), which have been interpreted with a Markov chain perspective (Hagemann et al., 2022), relying on Markov kernels and Radon-Nikodym derivatives just as our theory of generalized GFlowNets.

2.2. Discrete GFlowNets

GFlowNets were first framed as a reinforcement learning (RL) algorithm (Bengio et al., 2021a), with discrete state and action spaces, that trains a sampler of a target distribution given by its unnormalized probability mass function, called reward, using a local consistency objective known as flow matching. This contrasts with usual RL algorithms that aim at maximizing a given reward function, but is equivalent to entropy-regularized RL methods in special cases. Bengio et al. (2021b) laid out the theoretical foundations of GFlowNets, based on flow networks defined on DAGs, and proposed the detailed balance loss as a more efficient alternative that bypasses the need to sum the flows over large sets of children and parents, opening the door for continuous states. The trajectory balance and subtrajectory balance losses (Malkin et al., 2022; Madan et al., 2022) have been found to be yet more efficient. Pan et al. (2022) proposed a framework to incorporate intrinsic exploration rewards when training a GFlowNet. Zhang et al.

(2023a); Malkin et al. (2023); Zimmermann et al. (2022) proved a partial equivalence between GFlowNets and hierarchical variational methods, but showed the superiority of GFlowNets when learning with trajectories sampled *off-policy*. Zhang et al. (2022) used GFlowNets for approximate maximum-likelihood training of energy-based models, bypassing the need for a given target reward.

Jain et al. (2022) used GFlowNets within an active learning loop to design biological sequences. Zhang et al. (2023b) used GFlowNets for an NP-hard combinatorial optimization problem. Other applications include Bayesian structure learning: Deleu et al. (2022); Nishikawa-Toomey et al. (2022) learn posteriors over the combinatorially large space of causal graphs, which are naturally compositional.

Review of discrete GFlowNets. Given a non-negative target reward function R on a finite space $\mathcal{X}$, which coincides with a subset of the vertices of a DAG called the terminating states, GFlowNet training objectives aim at learning transition probabilities $P_F(s' \mid s)$ along the edges of the DAG. The *(forward) action policy* P_F induces a marginal distribution $P_F^\top$ over the terminating states, the final states of trajectories that begin at the designated initial state and sample transitions according to P_F. The parameters of the forward policy P_F are sequentially updated using a stochastic gradient of the objective function applied to a trajectory, sampled either from the forward policy itself (*on-policy*), or a modified version thereof in order to incentivize exploration (*off-policy*). When the loss is at a global minimum, the forward policy is able to effectively sample from a probability distribution over $\mathcal{X}$ proportional to R.

All GFlowNet losses must introduce additional objects into the parametrization to cope with the intractable representation of $P_F^\top(x)$ as a sum of the likelihoods of all (possibly exponentially many) trajectories leading to x. For example, the DB and TB objectives use a parametric *backward policy* $P_B(s \mid s')$, which specifies a distribution over the parents of any state in the DAG.

3. A theory for generalized GFlowNets

3.1. Practical summary

A summary of the key differences and analogies between discrete and generalized GFlowNets is provided in Table 1, and the precise way in which discrete GFlowNets are special cases of generalized GFlowNets is stated in §B.

The theory we develop in this section shows that the main losses used to train discrete GFlowNets, namely the detailed balance (Bengio et al., 2021b) and the trajectory balance (Malkin et al., 2022) losses, *naturally* extend to generalized GFlowNets, simply by replacing probability mass functions with probability density functions. Most impor-

Table 1. Dictionary between discrete and generalized GFlowNets

Discrete GFlowNet	Generalized GFlowNet	Reference
The state space is a finite set with distinguished source and sink states	The state space is a topological space with distinguished source and sink states, and may consist of both continuous and discrete components	Def. 1
Children and parents of a state s	Supports of the measures $\kappa(s,-)$, $\kappa^b(s,-)$	Def. 1
All states are reachable from s_0	All open sets are reachable from s_0 with nonzero likelihood	(1)
The state $\perp$ has no outgoing edges	The state $\perp$ is absorbing	(2)
The state graph is acyclic ($\Rightarrow$ trajectory lengths are bounded)	The measurable pointed graph is finitely absorbing	(7)
State flow F, forward policy P_F, backward policy P_B	Flow measure μ, forward kernel P_F, backward kernel P_B	Defs. 3 and 5
Transition likelihoods $P_F(-\vert s)$ only positive along edges	Transition kernels $P_F(s,-)$ absolutely continuous w.r.t. κ	Def. 3
Terminating distribution $P_F^\top$	Terminating state measure $P_\top$	Def. 9 and Thm. 1
Flow-matching implies sampler matches reward function R	Flow-matching implies sampler matches reward measure R	Def. 3 and Thm. 1
Detailed balance: $F(s)P_F(s'\vert s) = F(s')P_B(s\vert s')$	Detailed balance: $\mu(ds)P_F(s,ds') = \mu(ds')P_B(s',ds)$	Def. 5
Trajectory balance: $ZP_F(\tau) = R(x_\tau)P_B(\tau\mid x_\tau)$	Trajectory balance: $ZP_F(s_0,ds_1)\ldots P_F(s_n,\{\perp\}) = R(ds_n)P_B(s_n,ds_{n-1})\ldots P_B(s_1,\{s_0\})$	Def. 6

tantly, however, the soundness of the theory relies upon the following assumptions, which need to be carefully verified when training a GFlowNet in an infinite space:

(1) The structure of the state space must allow all states to be reachable from the source state s_0 (1);
(2) The structure must ensure that the number of steps required to reach any state from s_0 is bounded (7);
(3) The learned probability measures need to be expressed through densities over states, rather than over actions.

These assumptions are naturally verified in discrete domains, as long as the state space is described by a pointed directed acyclic graph (Bengio et al., 2021b).

On previous attempts to train continuous GFlowNets. While Li et al. (2023) proposed to train continuous GFlowNets by writing the flow matching conditions as integrals rather than sums, assumptions (1) and (3) are violated in a critical way. First, the environments considered in Li et al. (2023)'s experiments violate assumption (1), without which the main GFlowNet training theorems do not hold. Second, regarding (3), Li et al. (2023) implicitly assumes that for a state s and flow function $F(s \to s)$,

$$\int_{s':s\to s'} F(s \to s')\, ds' = \int_a F(s \to T(s,a))\, da,$$

where the second integral is taken over actions and $T(s,a)$ is the state reached by taking action a from s. This change of variables is invalid in general: *the integrand on the right side is missing the Jacobian term* $\frac{dT(s,a)}{da}$, which need not equal 1. In particular, it does not equal 1 in the environments studied by Li et al. (2023) (although it may hold in special cases, such as sampling in Euclidean spaces where $T(s,a) = s + a$). These issues are concerning for the scope of that

method's applicability.

3.2. Structured state space

Note. To help the reader form a mental picture, we list the concepts introduced and their discrete analogues in Table 1 and formally state the connection in §B. Paragraphs marked (⋆) explain the meaning of the technical results.

(⋆) **How could one describe a structure in general spaces, similar to DAGs on finite sets?** In finite sets, it would suffice to enumerate the child sets and parent sets of all states, with the constraint that s' is a child of s if and only if s is a parent of s'. In general state spaces, however, enumeration is replaced by measure. One could thus define, for each state, a measure on the state space describing what states can be accessed in one step.

The structured state spaces we consider will be called *measurable pointed graphs* and rely on *transition kernels* (Nummelin, 2004; Cappé et al., 2009; Petritis, 2015), of which we recall the definition in §A.

Definition 1 (Measurable pointed graph). *A measurable pointed graph* $G = (\bar{\mathcal{S}}, \mathcal{T}, \Sigma, s_0, \perp, \kappa, \kappa^b, \nu)$ *consists of:*

- *A topological space* $(\bar{\mathcal{S}}, \mathcal{T})$, *where* $\mathcal{T}$ *is the set of open subsets of* $\bar{\mathcal{S}}$ *and* Σ *is the Borel σ-algebra associated to the topology on* $\bar{\mathcal{S}}$;
- *A pair of distinct distinguished states* $s_0 \in \bar{\mathcal{S}}$ *and* $\perp \in \bar{\mathcal{S}}$, *called the* source state *and* sink state, *such that* $\{s_0\}$ *and* $\{\perp\}$ *are both open and closed sets. We define* $\mathcal{S} = \bar{\mathcal{S}} \setminus \{\perp\}$ *and* $\mathcal{S}^\circ = \mathcal{S} \setminus \{s_0\}$, *so the topology on* $\bar{\mathcal{S}}$ *is the disjoint union topology on* $\{s_0\}$, $\{\perp\}$, *and* $\mathcal{S}^\circ$.
- *A σ-finite transition kernel* κ *on* $(\bar{\mathcal{S}}, \Sigma)$, *called the* reference kernel,

- A σ-finite transition kernel κ^b on $(\bar{S}, \Sigma)$, called the backward reference kernel,
- A strictly positive σ-finite measure ν on $(\bar{S}, \Sigma)$, called the reference measure,

such that the following conditions hold:

$$\forall B \in \mathcal{T} \setminus \{\emptyset\} \quad \exists n \geq 0 : \kappa^n(s_0, B) > 0, \tag{1}$$

$$\kappa(\perp, -) = \delta_\perp, \tag{2}$$

$$\forall B \in \Sigma, \ s \mapsto \kappa(s, B) \text{ is continuous}, \tag{3}$$

$$\forall B \in \Sigma \otimes \Sigma, \ ((s_0, s_0) \notin B, \ (\perp, \perp) \notin B) \Rightarrow$$
$$\nu \otimes \kappa(B) = \nu \otimes \kappa^b(B), \tag{4}$$

$$\forall B \in \Sigma, \ \kappa^b(s_0, B) = 0, \tag{5}$$

$$\forall s \in \mathcal{S}, \ \kappa(s, \{\perp\}) > 0 \Rightarrow \kappa(s, \{\perp\}) = 1. \tag{6}$$

The measurable pointed graph is called finitely absorbing if

$$\exists N > 0 : \mathrm{supp}(\kappa^N(s_0, -)) = \{\perp\}, \tag{7}$$

in which case the minimal such N is called the maximal trajectory length.

($\star$) The reference transition kernel κ provides a notion of "structure" of the state space. The support of $\kappa(s, -)$ (resp. $\kappa^b(s, -)$) can be thought of as the child set (resp. parent set) of the state s. For example, in a discrete graph, $\kappa(s, -)$ could be uniform over the children of s. The reference kernel is not a policy to be sampled, but an object needed to define probability densities of policies. The measure ν, the reference with respect to which flows and rewards are defined, is typically a simple measure, such as the counting measure on a discrete set or the standard Lebesgue measure on a Euclidean space.

In practice, if the structure is only defined by the reference kernel κ, then ν and κ^b satisfying the conditions of Def. 1 can be defined from κ under some mild assumptions, as we discuss in Prop. 3 in §A.5.

From now on, we fix a finitely absorbing measurable pointed graph $G = (\bar{S}, \mathcal{T}, \Sigma, s_0, \perp, \kappa, \kappa^b, \nu)$ with maximal trajectory length N.

Definition 2 (Terminating states). *The set of* terminating states $\mathcal{X}$ *is defined by:*

$$\mathcal{X} = \{s \in \mathcal{S} \ : \ \kappa(s, \{\perp\}) > 0\}. \tag{8}$$

($\star$) Terminating states are ones from which one can transition to $\perp$ with positive probability. Any transition kernel can be sampled for n steps, yielding a measure over n-step trajectories and a marginal measure over states reached after n steps. As described in the Appendix (§A.4), this can be used to define the marginal terminating measure $P_\top$ of a transition kernel P_F, used in the next section.

3.3. Flows

Definition 3 (Flows and flow-matching conditions). *Given a σ-finite measure μ over $(\bar{S}, \Sigma)$ that is absolutely continuous w.r.t. ν (we write $\mu \ll \nu$), and a σ-finite Markov kernel P_F on $(\bar{S}, \Sigma)$ (i.e. a transition kernel such that each $P_F(s, -)$ is a probability measure) satisfying:*

(1) $P_F(s, -) \ll \kappa(s, -)$ for every $s \in \bar{S}$,
(2) $s \mapsto P_F(s, B)$ is continuous for every $B \in \Sigma$,

*P_F is said to be a **forward kernel** over G. We say that the tuple $F = (\mu, P_F)$ satisfies the **flow-matching (FM) conditions** if for any bounded measurable function $f : \bar{S} \to \mathbb{R}$ satisfying $f(s_0) = 0$, we have*

$$\int_{\bar{S}} f(s')\mu(ds') = \iint_{S \times \bar{S}} f(s')\mu(ds)P_F(s, ds'). \tag{9}$$

*In which case, we say that F is a **flow** over G.*

($\star$) The condition of absolute continuity w.r.t. the reference kernel κ indicates that the flow F must follow the "structure" of the measurable pointed graph, by assigning positive measure only to parts of the space where the measure induced by κ is also positive. The kernel P_F can be represented through a density function with respect to κ, which represents a probability *mass* (if the action space is discrete) or a probability *density* (if it is continuous). This allows to write conditions such as (9) using densities (Radon-Nikodym derivatives), thus providing practical loss functions to train GFlowNets. We expand on this point in §3.5.

Definition 4 (Reward-matching conditions). *Let $F = (\mu, P_F)$ be a flow over G. Given a positive and finite measure R over $\mathcal{X}$, called the reward measure, satisfying $R \ll \nu$, the flow F is said to satisfy the* reward-matching condition *w.r.t. R if we have:*

$$R(dx) = \mu(dx)P_F(x, \{\perp\}). \tag{10}$$

The following theorem, proved in §E, ascertains that, similar to discrete GFlowNets, when the flow and reward matching conditions are satisfied, then recursively sampling from the Markov kernel P_F starting from s_0 (until reaching $\perp$) yields samples from the normalized reward.

Theorem 1. *If $F = (\mu, P_F)$ is a flow over G, that satisfies the reward matching conditions (10) w.r.t. a measure R, then the corresponding terminating state measure $P_\top$ (Def. 10) is a probability measure and satisfies for all $B \in \Sigma_{|\mathcal{X}}$:*

$$P_\top(B) = \frac{1}{R(\mathcal{X})} R(B). \tag{11}$$

($\star$) $R(\mathcal{X})$, the reward measure taken over the set of all terminating states $\mathcal{X}$, corresponds to the total reward or partition function Z of GFlowNets. Certain conditions (Def. 3 and 4) on μ, which represents a state flow, and P_F, which

represents a policy, imply that the marginal terminating distribution of the policy is proportional to the reward. These conditions correspond to the "flow in = flow out" condition at vertices of a discrete DAG.

3.4. Detailed balance and trajectory balance

In finite GFlowNets, the detailed balance conditions (Bengio et al., 2021b) and the trajectory balance conditions (Malkin et al., 2022) were converted into training objectives in order to sample from a target unnormalized distribution. In this section, we present analogous conditions for general measurable pointed graphs.

Definition 5. *Let μ be a σ-finite measure over $(\bar{S}, \Sigma)$ such that $\mu \ll \nu$, P_F a forward kernel over G, and P_B a transition kernel on $(\bar{S}, \Sigma)$ such that:*

(1) $P_B(s, -) \ll \kappa^b(s, -)$ for every $s \in \bar{S}$,
(2) $s \mapsto P_B(s, B)$ is continuous for every $B \in \Sigma$,
(3) $P_B(s, -)$ is a probability measure for every $s \neq s_0$,

*P_B is then said to be a **backward kernel** over G. We say that (μ, P_F, P_B) satisfy the **detailed balance (DB) conditions** if for any bounded measurable function $f : S \times \bar{S} \to \mathbb{R}$ satisfying $f(s, s_0) = 0$ for every $s \in S$, we have*

$$\iint_{S \times \bar{S}} f(s, s')\mu(ds)P_F(s, ds') \qquad (12)$$
$$= \iint_{S \times \bar{S}} f(s, s')\mu(ds')P_B(s', ds).$$

The following proposition, proved in §E, shows an equivalence between the DB and FM conditions.

Proposition 1. *If (μ, P_F, P_B) satisfy the detailed balance conditions in Def. 5, then $F = (\mu, P_F)$ satisfies the flow-matching conditions in Def. 3 and is thus a flow.*

Definition 6. *Let P_F be a forward kernel over G, P_B a backward kernel over G, and $Z \in \mathbb{R}_+$. Let R be a positive finite measure on X. The triple (Z, P_F, P_B) satisfies the **trajectory balance (TB) conditions** w.r.t. R if for any $n \geq 0$ and any bounded measurable function $f : \bar{S}^{n+2} \to \mathbb{R}$:*

$$\int_{\bar{S}^{n+2}} Z f(s, \overrightarrow{s_{1:n+1}}) \mathbb{1}_{s_n \neq \perp, s_{n+1} = \perp} P_F^{\otimes n+1}(s_0, ds \overrightarrow{ds_{1:n+1}}) \quad (13)$$
$$= \int_{\bar{S}^{n+1}} \mathbb{1}_{s = s_0} f(s, \overrightarrow{s_{1:n}}, \perp) R(ds_n) P_B^{\otimes n}(s_n, ds' \overrightarrow{ds_{n-1:1}} ds),$$

where $\overrightarrow{s_{1:n}}$ denotes $(s_1, \ldots, s_n)$ and $\overrightarrow{ds_{1:n}}$ denotes $ds_1 \ldots ds_n$.

The following proposition, proved in §E, shows an equivalence between the TB and both the FM and reward matching conditions.

Proposition 2. *If (Z, P_F, P_B) satisfy the TB conditions (13) w.r.t. a measure R, then $F = (\mu, P_B)$, where μ is defined by:*

(1) $\mu(\{\perp\}) = \mu(\{s_0\}) = Z$
(2) $\forall B \in \Sigma_{|S}: \mu(B) = \mu(\{s_0\}) \sum_{n=0}^{\infty} P_F^n(s_0, B)$

satisfies both the flow-matching conditions (9) and the reward matching conditions (10) w.r.t. R.

($\star$) Analogues of the DB and TB conditions for discrete GFlowNets were stated and shown to imply the FM conditions. In the next section, they will be used to construct training objectives for parametric policies.

3.5. Training losses for GFlowNets

Above, we have presented three conditions under which a sampler based on a Markov kernel P_F samples from the normalized version of a given reward measure. In practice, similar to discrete GFlowNets, the objects of interest (μ, P_F, P_B, Z) are parametrized by a vector θ, and the goal is to learn θ using gradient-based learning. In this section, we derive losses corresponding to the previous objectives.

We recall the Radon-Nikodym theorem that states that for any two given σ-finite measures p and q on a measurable space $(U, \mathcal{U})$ satisfying $p \ll q$, there exists a measurable function $f : U \to \mathbb{R}_+$, which is unique up to a set of measure zero under q, called the density or the Radon-Nikodym derivative of p w.r.t. q, such that:

$$\forall A \in \mathcal{U}, \; p(A) = \int_A f(u)q(du). \qquad (14)$$

This theorem is convenient as it allows to bypass the need to define the measures μ, $P_F(s, -)$, $P_B(s, -)$ on every measurable set, and only requires parametrizing the corresponding densities (w.r.t. ν, $\kappa(s, -)$, and $\kappa^b(s, -)$ respectively).

Definition 7 (Losses). *Let $u : S \to \mathbb{R}_+$, $p_F : S \times \bar{S} \to \mathbb{R}_+$, and $p_B : S \times S \to \mathbb{R}_+$ be three functions, and $Z \in \mathbb{R}_+$ a scalar, all parametrized by a vector θ, and satisfying for every θ:*

$$\forall s \in \bar{S}, \; \int_{\bar{S}} p_F(s, s'; \theta)\kappa(s, ds') = 1 \qquad (15)$$

$$\forall s' \in \bar{S}, \; \int_{\bar{S}} p_B(s', s; \theta)\kappa^b(s', ds) = 1 \qquad (16)$$

*The **flow-matching** (FM) loss is defined for every $s' \in S$ as:*

$$L_{FM}(s'; \theta) = \left(\log \frac{\int_S u(s; \theta)p_F(s, s'; \theta)\kappa^b(s', ds)}{u(s'; \theta)} \right)^2$$

*The **detailed balance** (DB) loss is defined for every $(s, s') \in S \times S$ as:*

$$L_{DB}(s, s'; \theta) = \left(\log \frac{u(s; \theta)p_F(s, s'; \theta)}{u(s'; \theta)p_B(s', s; \theta)} \right)^2$$

*Denoting by r the density of the reward measure R w.r.t. the reference measure ν, the **reward-matching** (RM) loss is defined for any $x \in X$ as:*

5

$$L_{RM}(x;\theta) = \left(\log \frac{u(x;\theta)p_F(x,\perp;\theta)}{r(x)}\right)^2$$

*Finally, the **trajectory balance** (TB) loss is defined for every complete trajectory $\tau = (s_0, s_1, \ldots, s_n, s_{n+1}) \in \{s_0\} \times \mathcal{S}^n \times \{\perp\}$ (also denoted $\overrightarrow{s_{0:n+1}}$) where $s_n \in \mathcal{X}$ and $s_{n+1} = \perp$ as:*

$$L_{TB}^n(\tau;\theta) = \left(\log \frac{Z(\theta)\prod_{t=0}^n p_F(s_t, s_{t+1};\theta)}{r(s_n)\prod_{t=0}^{n-1} p_B(s_{t+1}, s_t;\theta)}\right)^2.$$

Note that one could derive in a similar fashion a subtrajectory balance loss, similar to the one used in discrete GFlowNets (Madan et al., 2022).

(★) The above losses resemble discrete GFlowNet losses. When the action space is discrete, and the reference measures are the counting measures over vertices of a DAG, $p_F(s, s')$ is a transition probability $P_F(s'|s)$. When it is continuous, it represents a conditional probability density over s', given s.

Conversely, from functions $u(-;\theta), p_F(-;\theta), p_B(-;\theta)$, we can define a measure $\mu(-;\theta)$ on $(\bar{\mathcal{S}}, \Sigma)$ whose density w.r.t. ν is u and forward and backward kernels $P_F(-;\theta), P_B(-;\theta)$ such that $p_F(s,-;\theta)$ and $p_B(s',-;\theta)$ are their densities of w.r.t. $\kappa(s,-)$ and $\kappa^b(s,-)$, respectively[1].

(★) The following theorem, proved in §E, ensures that, similar to the discrete case, minimizing the losses above leads to samplers of the right probability measure.

Theorem 2. *(1) If $L_{FM}(-;\theta) = 0$ ν-almost surely, then $F = (\mu, P_F)$ is a flow (i.e. satisfies the flow-matching conditions in Def. 3).*
(2) If $L_{DB}(-;\theta) = 0$ $\nu \otimes \kappa$-almost surely, then (μ, P_F, P_B) satisfy the detailed balance conditions in Def. 5.
(3) If $L_{RM}(-;\theta) = 0$ $\nu_{|\mathcal{X}}$-almost surely, then (μ, P_F) satisfies the reward matching conditions in (10).
(4) If $L_{TB}^n(-;\theta) = 0$ $((\nu \otimes \kappa^{\otimes n+1})_{|\{s_0\} \times \mathcal{S}^n \times \{\perp\}})$-almost surely for every $n \geq 0$, then $(Z\nu(\{s_0\}), P_F, P_B)$ satisfy the trajectory balance condition in Def. 6.

An important consequence of Thm. 2 is that if we can find density functions that achieve zero loss using any of the above objectives almost surely, in addition to the reward-matching loss, then we obtain a way to sample terminating states (i.e., elements of $\mathcal{X}$) proportionally to the reward measure R, according to Thm. 1.

Training generalized GFlowNets. The FM, DB, and TB losses can be minimized using states (resp. pairs of subsequent states, trajectories) obtained from trajectories sampled from a training policy π, which can be P_F itself (*on-policy*), or a modification of it to encourage exploration (*off-policy*). Thm. 2 suggests that the parameters θ could be updated

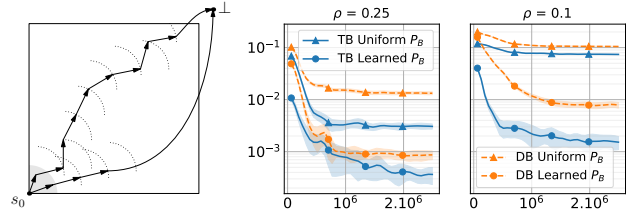[1]The measures at $\perp$ are irrelevant.

Figure 1. **(a)** Measurable pointed graph structure of the environment in §4.1: starting at s_0, the first action makes a step within the grey quarter-disc, and subsequent actions make steps of a fixed size or terminate. **(b)** Evolution of the JSD during training of TB and DB, with both a uniform P_B and a learned P_B, for $\rho = 0.25$; **(c)** $\rho = 0.1$. x-axis is the number of sampled trajectories. Shaded areas represent standard deviations across 6 runs.

with stochastic gradients $\mathbb{E}_{\tau = \overrightarrow{s_{0:n+1}} \sim \pi}[\nabla_\theta \mathcal{L}]$, where $\mathcal{L}$ is $\sum_{t=1}^n L_{FM}(s_t;\theta) + \alpha L_{RM}(s_n;\theta)$, or $\sum_{t=0}^n L_{DB}(s_t, s_{t+1};\theta) + \alpha L_{RM}(s_n;\theta)$ or $L_{TB}(\tau;\theta)$.

4. Experiments

4.1. A synthetic continuous environment

In this section, we study a synthetic environment inspired by the hypergrid environment (Bengio et al., 2021a; Malkin et al., 2022; 2023), with varying trajectory lengths and a pointed graph structure imposing a mixed discrete and continuous probability measure for the policy P_F. Code for these experiments can be found at https://github.com/saleml/continuous-gfn.

Structure of the state space. The measurable pointed graph is specified by $\mathcal{S} = [0, 1]^2$, and $s_0 = (0, 0)$. A hyperparameter ρ, called the step size, controls the maximal trajectory length. $\kappa(s_0, -)$ is the Lebesgue measure on D_0, the northeastern quarter disk of radius ρ centered at s_0. When $s \neq s_0$, and $||s|| < 1 - \rho$, $\kappa(s, -)$ is the sum of the one-dimensional Lebesgue arclength measure on C_s^+ (the intersection of the northeastern quarter circle of radius ρ centered at s and $\mathcal{S}$) and the Dirac measure $\delta_\perp$. Finally, when $||s|| > 1 - \rho$, $\kappa(s, -) = \delta_\perp$. The forward structure is depicted in Fig. 1(a). The backward reference kernel κ^b is defined similarly.

The reference measure ν is the sum of the Lebesgue measure on $\mathcal{S}$, δ_{s_0}, and $\delta_\perp$. All states besides s_0 are terminating.

The reward measure R on $\mathcal{X}$ is specified by a density function r depicted in Fig. 2(a). The densities p_F and p_B are parametrized with mixtures of Beta distributions for the continuous components.

In Fig. 1(b,c), we compare DB and TB on two versions of the environment ($\rho \in \{0.1, 0.25\}$), with both a uniform and a learned P_B, using the Jensen-Shannon divergence (JSD, §C.1) between the learned terminating state distribution and the target distribution as an evaluation metric. The results confirm the findings of Malkin et al. (2022) on
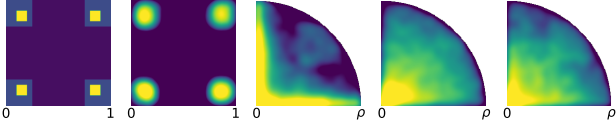
Figure 2. **(a)** Reward density in $[0, 1]^2$. **(b)** KDE fit on terminating states of the models trained with TB, $\rho = 0.25$. **(c)** KDE fit on samples from the reward, brought back to D_0 using a **uniform** P_B, corresponding to what $P_F(s_0, -)$ needs to be in order to satisfy DB or TB. A richer search space for the densities $p_F(s, -)$ is required to fit this distribution. **(d)** $P_F(s_0, -)$ for a trained model with learnable P_B. **(e)** The measure induced by a trained P_B on D_0, which matches the learned $P_F(s_0, -)$ in (d).
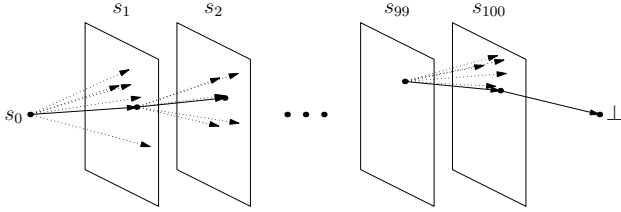

Figure 3. The GFlowNet state space for stochastic control tasks. The solid arrows show a possible sampling trajectory and the dashed arrows show other possible actions, i.e., point to other states in the support of the reference kernel κ.

the discrete grid domain: the TB loss is more efficient in terms of credit assignment, as it learns to model the target distribution faster and more precisely than DB, and the environment with longer trajectory lengths is harder to model. Additionally, learning a backward policy significantly improves the learning curves of both methods. A justification of the importance of learning in a backward policy is provided in Fig. 2(c,d,e). Fig. 2(b) shows a KDE plot fit on terminating states sampled from the model trained with TB on the $\rho = 0.25$ domain. We provide more details in §C.2.

4.2. Low-dimensional stochastic control

In this section, we show how generalized GFlowNets with a state space of a particular form can be used to learn (discretizations of) stochastic differential equations so as to sample from a black-box target density. We bridge two recent works: Zhang & Chen (2022), from which we borrow the datasets and many parts of the experimental setup, and Malkin et al. (2023), where various algorithms for training stochastic samplers in discrete spaces were considered and whose claims we validate in the continuous case.

We restate the problem considered by Zhang & Chen (2022) in GFlowNet terms. A reward density is given on a Euclidean space $\mathbb{R}^n$ (e.g., the plane in Fig. 3). The state space is $\mathcal{S} = \{s_0\} \cup (\mathbb{R}^n \times \{1, 2, \ldots, T\})$, where T is the number of moves an agent will make before terminating (here, $T = 100$). Thus the noninitial states are pairs $(\mathbf{x}_t, t)$ where

Table 2. Log-partition function estimation bias using importance-weighted bound B_{RW} (mean and standard deviation over 10 runs). The **bold** value in each column shows the best result and all those statistically equivalent to it ($p > 0.1$ under a Welch's t-test). Algorithms assuming access to the gradient of the reward (non-black-box) are shown for comparison. Rows marked with $*$ require importance weighting for gradient estimation. Cells with – were unstable to optimize. Last three rows from Zhang & Chen (2022).

Black box?			Gaussian mixture	Funnel
✓	Off-policy	GFlowNet TB	$-\mathbf{0.003} \pm 0.011$	$-\mathbf{0.026} \pm 0.020$
✓	Off-policy	Reverse KL*	-1.609 ± 0.546	–
✓	Off-policy	Forward KL*	$-\mathbf{0.001} \pm 0.013$	-0.087 ± 0.081
✓	On-policy	GFlowNet TB	-1.301 ± 0.434	$-\mathbf{0.012} \pm 0.108$
✓	On-policy	Reverse KL	-1.237 ± 0.413	-0.040 ± 0.023
✓	On-policy	Forward KL*	-0.007 ± 0.023	-0.034 ± 0.143
✓	Non-SDE	SMC	-0.362 ± 0.293	-0.216 ± 0.157
✗	On-policy	PIS-NN	-1.192 ± 0.482	-0.018 ± 0.020
✗	Non-SDE	HMC	-1.876 ± 0.527	-0.835 ± 0.257

$\mathbf{x}_t \in \mathbb{R}^n$ and $1 \leq t \leq T$; we identify s_0 with $(\mathbf{0}, 0)$. Trajectories begin at s_0 and make successive steps through the copies of $\mathbb{R}^n$ until reaching the sink state.[2] Learning a forward policy amounts to learning a conditional probability density $p(\mathbf{x}_{t+1}|\mathbf{x}_t, t)$ over $\mathbb{R}^n$. In particular, if this density is Gaussian, then the policy is the T-step Euler-Maruyama discretization of an Itô stochastic differential equation (SDE).

Zhang & Chen (2022) studied this problem in the case where the backward policy is fixed to be the discretization of a Brownian motion with fixed variance $\frac{\sigma}{T}$ pinned at $(\mathbf{0}, 0)$, and the forward policy is constrained to be Gaussian with the same variance $\frac{\sigma}{T}$ but with learned mean. (The theory of SDEs implies that in the $T \to \infty$ limit, the forward policy P_F that minimizes the GFlowNet loss is indeed Gaussian with the same variance as the fixed P_B.) We thus aim to learn a function $\mu(\mathbf{x}_t, t)$, the mean of the forward policy, so as to make the policy sample from the target reward density.

The path integral sampler (PIS) training objective proposed by Zhang & Chen (2022) minimizes the reverse KL divergence between two measures over trajectories: that defined by P_F and that defined by R and P_B. By Theorem 1 of Malkin et al. (2023) (the proof of which trivially generalizes to the continuous case), the gradient of this objective with respect to the parameters of P_F is proportional, in expectation, to that of the TB gradient when trained on-policy. A key difference between PIS and the on-policy TB objective is that the latter does not require access to the gradient of the reward distribution, but treats it as a black box.

Datasets, algorithms, and baselines. We evaluate

---

[2]To be precise, the reference measure ν is the Lebesgue measure on each copy of $\mathbb{R}^n$ and the counting measure on $\{s_0\}$. If s_i is a state in the i-th copy of $\mathbb{R}^n$ (if $i > 0$) or the initial state s_0 (if $i = 0$), the reference kernel $\kappa(s_i, -)$ is Lebesgue on the $(i + 1)$-st copy of $\mathbb{R}^n$ if $i < T$ and $\delta_\perp$ if $i = T$; κ^b is defined similarly.

GFlowNets and baselines on two synthetic densities: a 2-dimensional mixture of 9 Gaussians and the 10-dimensional funnel from MCMC literature (Hoffman & Gelman, 2011).

In addition to GFlowNet TB, we evaluate the two algorithms for minimizing divergences between trajectory measures studied by Malkin et al. (2023): the reverse KL optimized via policy gradient – equivalent in expectation to TB – and the forward KL, for which gradient estimation requires importance weighting. We also evaluate the algorithms in an off-policy setting, where the training trajectories are sampled with additional variance injected into the policy to encourage exploration (see §C for details). We include baselines from Zhang & Chen (2022) as well.

All algorithms use the same model architecture as the PIS baseline for $\mu(\mathbf{x}_t, t)$ and are evaluated using two metrics as defined in Zhang & Chen (2022): the log-partition function estimation bias using simple and importance-weighted variational bounds, as defined in §C.3.

Results and discussion. From the results in Table 2, and the extended results in Table C.1, we conclude that the two main observations of Malkin et al. (2023) continue to hold in this continuous setting. First, as expected, on-policy TB and reverse KL perform similarly when both can be stably optimized. Second, in settings where off-policy exploration is important, TB is more stable and achieves a better fit to the target than the other objectives, which require importance weighting for gradient estimation. Fig. C.1 shows that exploration is necessary to discover modes. Finally, we note that TB is competitive with the PIS objective despite not having access to gradients of the reward density.

4.3. Stochastic control on a torus

We consider a variant of the samplers discussed in §4.2 to model reward densities on the surface of a 2D torus. Distributions over tori are useful to model torsion angles in molecular conformations, as we illustrate in §C.4 and Fig. C.2 with the alanine dipeptide molecule.

To model the surface of a torus, the measurable pointed graph is defined by $\mathcal{S} = \{s_0\} \cup [0, 2\pi)^2 \times \{t \in \mathbb{N}, 1 \le t \le T\}$, where t denotes the step number and T the trajectory length, and $s_0 = ((0, 0), 0)$. Note that here $[0, 2\pi)$ has the topology of the circle, not that induced from the real line.

We consider two reward densities: a synthetic multimodal density, and a density based on the energy $\mathcal{E}$ of the alanine dipeptide molecule as a function of two of the angles defining the conformation of the molecule. More details are provided in §C.4. We provide a visual representation of learned and reward distributions in Fig. 4.
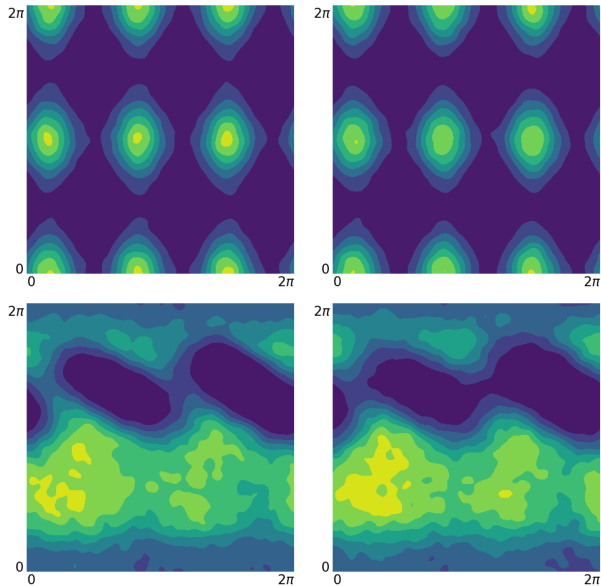

Figure 4. KDEs fit on samples from the reward functions (**a**: synthetic multimodal reward function, **c**: Boltzmann distribution of principal torsion angles of the alanine dipeptide molecule – details in §C.4) and on samples from the corresponding trained GFlowNets (**b, d**). The topology of the torus imposes periodic boundary conditions on $[0, 2\pi)^2$.

4.4. Posterior over continuous parameters in Bayesian structure learning

To show the capacity of GFlowNets to model a distribution over a mixed space of discrete and continuous quantities, we study here the problem of learning the structure of a Bayesian network and its parameters, from a Bayesian perspective. Extending the work of Deleu et al. (2022), our goal here is to approximate the (joint) posterior distribution $P(G, \theta \mid \mathcal{D})$ over the directed acyclic graph (DAG) structure G of the Bayesian Network (discrete component) and the parameters θ of its conditional probability distributions (continuous component), given a dataset of observations $\mathcal{D}$.

We use a GFlowNet that is structured as follows: starting from the empty graph, the DAG G is first generated one edge at a time, following the structure of DAG-GFlowNet (Deleu et al., 2022). Once the graph G has been completely generated, we then sample the parameters θ associated to it, in order to reach a valid terminating state (G, θ). Details about the state space, and the forward transition probability are given in §C.5. We use the subtrajectory balance loss (Madan et al., 2022) to train the GFlowNet with $R(G, \theta) = P(\mathcal{D} \mid \theta, G)P(\theta, G)$ as a reward function.

In order to evaluate our approximation against the target distribution, we consider problems where the true posterior $P(G, \theta \mid \mathcal{D})$ may be computed in closed form. More precisely, we assume that the Bayesian network follows

Table 3. Comparison between GFlowNet and other methods based on variational inference on the Bayesian structure learning task. (Graphs) RMSE between the estimated edge marginals and the exact edge marginals. (Params.) Average negative log-probability of the parameter samples under the exact posterior $P(\theta \mid G, \mathcal{D})$.

| | | Number of variables (d) | | |
		3	4	5
Graphs	BCD Nets	–	2.13×10^{-1}	2.61×10^{-1}
	DiBS	3.28×10^{-1}	2.95×10^{-1}	3.15×10^{-1}
	GFlowNet	$\mathbf{1.50 \times 10^{-2}}$	$\mathbf{1.61 \times 10^{-2}}$	$\mathbf{1.80 \times 10^{-2}}$
Params.	BCD Nets	–	2.17×10^{2}	2.63×10^{2}
	DiBS	5.87×10^{2}	1.12×10^{3}	2.12×10^{3}
	GFlowNet	$\mathbf{-1.75 \times 10^{0}}$	$\mathbf{-3.06 \times 10^{0}}$	$\mathbf{-5.17 \times 10^{0}}$

a linear-Gaussian model and that the number of random variables $d \leq 5$. Additional details about the experimental settings and metrics are available in §C.5. In Table 3, we compare the performance of the GFlowNet with two baseline methods based on variational inference: DiBS (Lorch et al., 2021) and BCD Nets (Cundy et al., 2021). In Table 3 (top), we report the root mean-square error (RMSE) between the edge marginals computed with the approximation and the exact posterior $P(G \mid \mathcal{D})$; we observe that the model learned by the GFlowNet is significantly more accurate on the discrete component, supporting the observation made in Malkin et al. (2023). Moreover, in Table 3 (bottom), we observe that the sampled θ from the GFlowNet are significantly more likely under the exact posterior $P(\theta \mid G, \mathcal{D})$, suggesting that the GFlowNet's approximation of the continuous component is also more accurate.

4.5. Connections with diffusion models

We show how the generalized GFlowNet framework can be applied *beyond* the setting of fitting a sampler to a target reward function. As

Table 4. ImageNet-32 results.

Method	FID↓	NLL↓
Baseline	17.65	4.57
MLE-GFN	16.36	4.47

shown in Zhang et al. (2023a), GFlowNets can also be trained to *maximize likelihood* of a given set of terminating states with an algorithm called MLE-GFN. Here we apply MLE-GFN to generalized GFlowNets to improve denoising diffusion probabilistic models (DDPMs; Ho et al., 2020).

Sampling process. The generative process in a DDPM can be seen as a special case of the sampling process in a generalized GFlowNet of the same form as in §4.2 and Fig. 3. A fixed number of steps T is made through a sequence of copies of a high-dimensional space $\mathbb{R}^n$ (with the i-th state in the trajectory representing, e.g., an image at noise level $T - i$). The policy at the first step, from s_0 to $(\mathbf{x}_1, 1)$, is constrained to be unit Gaussian, while subsequent steps are

conditional Gaussians with a known variance.

More specifically, recall that diffusion models begin with a sample $\mathbf{x}_1$ from a noise distribution and transform it through a sequence of conditional Gaussian steps $x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_T$ (note the unconventional reversed and one-based indexing). Viewing the intermediate samples $\mathbf{x}_t$ as states $(\mathbf{x}_t, t)$, we can cast sampling from the diffusion model as sampling from a GFlowNet, where the first action samples $\mathbf{x}_1$ by transitioning from the abstract initial state s_0 and subsequent actions follow a forward kernel $P_F(- \mid (\mathbf{x}_t, t))$ whose support is $\{(\mathbf{x}, t+1) : \mathbf{x} \in \mathbb{R}^n\}$ and whose density is a Gaussian conditioned on $\mathbf{x}_t$ and t.

Noising process. While DDPMs typically fix the noising process – corresponding to the backward policy P_B in the GFlowNet – and learn only the denoiser (forward process), MLE-GFN allows learning both P_F and P_B as Gaussian policies. The description and proof of soundness of MLE-GFN, as well as details of the parametrization of means and variances, can be found in Zhang et al. (2023a).

Experimental result. We train a GFlowNet as described above on the ImageNet-32 dataset (treated as a set of terminating states) with $T = 100$ steps. Table 4 demonstrates the efficacy of our method compared to the DDPM baseline in terms of both the sample quality (FID) and density modeling (NLL). We defer other details and example images to §C.6.

5. Conclusion

We have developed a theory for generalized GFlowNets and illustrated it through experiments. Future work will exploit this theory and scale the experiments up to more complex and high-dimensional spaces where generation includes both discrete and continuous choices. Possible application areas include estimation of Bayesian neural network posteriors, molecular conformer generation (discussed in §C.4), and simulation-based inference for inverse problems in the natural sciences.

Acknowledgments

Author contributions

S.L., T.D., N.M. developed the theory and L.N.E. contributed some proofs. Experiments in §4 were done by S.L. (synthetic grid), N.M. (stochastic control), A.V. and A.H. (torus), T.D. (Bayesian structure learning), D.Z. (diffusion). Experiments on the importance of modeling assumptions were also done by P.L. Y.B. conceived and guided the project. All authors contributed to designing the experiments and writing the paper.

References

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

Bengio, Y., Lahlou, S., Deleu, T., Hu, E., Tiwari, M., and Bengio, E. GFlowNet foundations. *arXiv preprint 2111.09266*, 2021b.

Bergwerf, H. Molview, 2014. URL https://molview.org.

Cappé, O., Moulines, E., and Rydén, T. Inference in hidden markov models. In *Proceedings of EUSFLAT conference*, 2009.

Coffey, W., Kalmykov, Y., and Waldron, J. *The Langevin Equation: With Applications to Stochastic Problems in Physics, Chemistry and Electrical Engineering*. 09 2004.

Cundy, C., Grover, A., and Ermon, S. BCD Nets: Scalable variational approaches for Bayesian causal discovery. *Neural Information Processing Systems (NeurIPS)*, 2021.

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

Dockhorn, T., Vahdat, A., and Kreis, K. Score-based generative modeling with critically-damped langevin diffusion. *International Conference on Learning Representations (ICLR)*, 2022.

Doucet, A., Freitas, N., and Gordon, N. *An Introduction to Sequential Monte Carlo Methods*, pp. 3–14. Springer New York, 2001.

Hagemann, P., Hertrich, J., and Steidl, G. Stochastic normalizing flows for inverse problems: A Markov chains viewpoint. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3):1162–1190, 2022.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Neural Information Processing Systems (NeurIPS)*, 2020.

Hoffman, M. D. and Gelman, A. The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research (JMLR)*, 15:1593–1623, 2011.

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

Jing, B., Corso, G., Jeffrey Chang, R. B., and Jaakkola, T. Torsional diffusion for molecular conformer generation. *Neural Information Processing Systems (NeurIPS)*, 2022.

Li, Y., Luo, S., Wang, H., and Hao, J. CFlowNets: Continuous control with generative flow networks. *International Conference on Learning Representations (ICLR)*, 2023.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. *arXiv preprint 2210.02747*, 2022.

Lorch, L., Rothfuss, J., Schölkopf, B., and Krause, A. DiBS: Differentiable Bayesian structure learning. *Neural Information Processing Systems (NeurIPS)*, 2021.

Madan, K., Rector-Brooks, J., Korablyov, M., Bengio, E., Jain, M., Nica, A., Bosc, T., Bengio, Y., and Malkin, N. Learning GFlowNets from partial episodes for improved convergence and stability. *arXiv preprint 2209.12782*, 2022.

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

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

Mironov, V., Alexeev, Y., Mulligan, V. K., and Fedorov, D. G. A systematic study of minima in alanine dipeptide. *Journal of Computational Chemistry*, 40(2):297–309, 2018.

Neal, R. M. Annealed importance sampling. *Statistics and Computing*, 11(2):125–139, 2001.

Neal, R. M. MCMC using Hamiltonian dynamics. *arXiv preprint 1206.1901*, 2012.

Nishikawa-Toomey, M., Deleu, T., Subramanian, J., Bengio, Y., and Charlin, L. Bayesian learning of causal structure and mechanisms with GFlowNets and variational bayes. *arXiv preprint 2211.02763*, 2022.

Nummelin, E. *General irreducible Markov chains and non-negative operators*. Cambridge University Press, 2004.

Pan, L., Zhang, D., Courville, A., Huang, L., and Bengio, Y. Generative augmented flow networks. *International Conference on Learning Representations (ICLR)*, 2022.

Petritis, D. *Markov chains on measurable spaces*. 2015. URL http://perso.univ-rennes1.fr/dimitri.petritis/enseignement/markov/markov.pdf.

Sohl-Dickstein, J. N., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. *International Conference on Machine Learning (ICML)*, 2015.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Neural Information Processing Systems (NeurIPS)*, 2019.

Thiede, L., Miret, S., Sadowski, K., Xu, H., Phielipp, M., and Aspuru-Guzik, A. Conformer search using SE3-transformers and imitation learning. *NeurIPS 2022 AI for Accelerated Materials Design Workshop*, 2022.

Wu, H., Köhler, J., and Noé, F. Stochastic normalizing flows. *Neural Information Processing Systems (NeurIPS)*, 2020.

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

Zhang, D., Chen, R. T. Q., Malkin, N., and Bengio, Y. Unifying generative models with GFlowNets and beyond. *arXiv preprint 2209.02606*, 2023a.

Zhang, D., Rainone, C., Peschl, M., and Bondesan, R. Robust scheduling with GFlowNets. *International Conference on Learning Representations (ICLR)*, 2023b. To appear.

Zhang, Q. and Chen, Y. Path integral sampler: a stochastic control approach for sampling. *International Conference on Learning Representations (ICLR)*, 2022.

Zimmermann, H., Lindsten, F., van de Meent, J.-W., and Naesseth, C. A. A variational perspective on generative flow networks. *arXiv preprint 2210.07992*, 2022.

A. Transition kernels: Additional notations and definitions

We first recall the definition of transition kernels and Markov kernels

Definition 8 (Transition kernel). *Let $(\bar{S}, \Sigma)$ be a measurable (state) space. A function $\kappa : \bar{S} \times \Sigma \to [0, +\infty)$ is called a positive σ-finite transition kernel if*

1. *For any $B \in \Sigma$, the mapping $s \mapsto \kappa(s, B)$ is measurable, where the space $[0, +\infty)$ is associated with the Borel σ-algebra $\mathcal{B}([0, +\infty))$;*

2. *For any $s \in \mathcal{S}$, the mapping $B \mapsto \kappa(s, B)$ is a positive σ-finite measure on $(\bar{S}, \Sigma)$.*

A transition kernel such that the mappings $B \mapsto \kappa(s, B)$ are probability measures is called a Markov *kernel.*

Notations. Given a measurable space $(\bar{S}, \Sigma)$, we denote by $\Sigma_{|\mathcal{U}}$ the restriction of Σ to any subset $\mathcal{U}$ of $\bar{S}$.

A.1. Products of kernels.

Given a measurable space $(\bar{S}, \Sigma)$, a positive measure ν on $(\bar{S}, \Sigma)$, and a transition kernel κ on $(\bar{S}, \Sigma)$, we denote by $\nu\kappa$ (resp. $\nu \otimes \kappa$) the measure on $(\bar{S}, \Sigma)$ (resp. $(\bar{S} \times \bar{S}, \Sigma \otimes \Sigma)$) defined for $B \in \Sigma$ (resp. $B \in \Sigma \otimes \Sigma$) as:

$$\nu\kappa(B) = \int_{\bar{S}} \nu(ds)\kappa(s, B). \tag{17}$$

$$\nu \otimes \kappa(B) = \iint_{\bar{S}^2} \mathbb{1}_B(s, s')\nu(ds)\kappa(s, ds') \tag{18}$$

In particular, for any state $s \in \bar{S}$, the *n-step measure* $\kappa^n(s, -)$ is recursively defined by $\kappa^0(s, -) = \delta_s$, the Dirac at s, and:

$$\kappa^{n+1}(s, -) = \kappa^n(s, -)\kappa. \tag{19}$$

The following lemma, proved in §F ensures that absolute continuity between transition kernels transfers to n-step measures

Lemma 1. *Let P_F be a transition kernel on $(\bar{S}, \Sigma)$ such that $P_F(s, -) \ll \kappa(s, -)$ for every $s \in \bar{S}$. Then for every $n \geq 1$, $P_F^n(s, -)$ and $s \in \mathcal{S}$ is absolutely continuous wrt. $\kappa^n(s, -)$.*

A.2. Equality between measures

Given two measures p and q on $(\bar{S}, \Sigma)$, and a function $g : \bar{S} \to \mathbb{R}$, we use the notation:

$$p(ds) = g(s)q(ds)$$

to say that for any measurable bounded function $f : \bar{S} \to \mathbb{R}$:

$$\int_{\bar{S}} f(s)p(ds) = \int_{\bar{S}} f(s)g(s)q(ds). \tag{20}$$

Throughout the paper, we use the two notations interchangeable when the context allows it. Our proofs rely mostly on writing the equality with measurable bounded functions.

Equalities between product measures require a special care, especially when using both the kernel κ and the backward reference kernel κ^b. For example (4) means that for any measurable bounded function $f : \bar{S} \times \bar{S} \to \mathbb{R}$ satisfying $f(s_0, s_0) = f(\perp, \perp) = 0$:

$$\iint_{\bar{S} \times \bar{S}} f(s, s')\nu(ds)\kappa(s, ds') = \iint_{\bar{S} \times \bar{S}} f(s, s')\nu(ds')\kappa^b(s', ds).$$

We choose not to write (4) with the notation $\nu(ds)\kappa(s, ds') = \nu(ds')\kappa(s', ds)$ as it does not convey any information about when "ds" and "ds'" represent $\{s_0\}, \{s_0\}$ or $\{\perp\}, \{\perp\}$.

A.3. Trajectory measures

Definition 9. *Let P_F be a transition kernel on $(\bar{\mathcal{S}}, \Sigma)$. For any $n \geq 0$ and $s \in \bar{\mathcal{S}}$, P_F induces a measure $P_F^{\otimes n}(s, -)$ over the product space $(\bar{\mathcal{S}}^{n+1}, \Sigma^{\otimes (n+1)})$. $P_F^{\otimes n}(s, -)$, called the n-step trajectory measure at s recursively defined by*

$$P_F^{\otimes 0}(s, -) = \delta_s, \tag{21}$$

$$P_F^{\otimes n+1}(s, -) = P_F^{\otimes n}(s, -) \otimes P_F. \tag{22}$$

Notation. We use $\overrightarrow{s_{1:n}}$ to denote $(s_1, \ldots, s_n)$ and $\overrightarrow{ds_{1:n}}$ to denote $ds_1 \ldots ds_n$.

We can write for example: $P_F^{\otimes 1}(s, ds'\, ds_1) = \delta_s(ds')P_F(s', ds_1)$ and $P_F^{\otimes 2}(s, ds'\, ds_1\, ds_2) = \delta_s(ds')P_F(s', ds_1)P_F(s_1, ds_2)$, and more generally:

$$P_F^{\otimes n}(s, ds'\, \overrightarrow{ds_{1:n}}) = P_F^{\otimes n-1}(s, ds'\, \overrightarrow{ds_{1:n-1}})P_F(s_{n-1}, ds_n)$$

A.4. Terminating state measure

Given a measurable pointed DAG $G = (\bar{\mathcal{S}}, \mathcal{T}, \Sigma, s_0, \perp, \kappa, \kappa^b, \nu)$, any transition kernel P_F on $(\bar{\mathcal{S}}, \Sigma)$ induces a terminating state measure $P_\top$, which is the sum of the n-step terminating state measures defined as follows:

Definition 10. *Let P_F be a transition kernel on $(\bar{\mathcal{S}}, \Sigma)$. For any $n \geq 0$ we define the n-step terminating state measure $P_\top^n$ over $(\mathcal{X}, \Sigma_{|\mathcal{X}})$, for any $B \in \Sigma_{|\mathcal{X}}$ as:*

$$P_\top^n(B) = \int_{\bar{\mathcal{S}}^{n+1}} P_F^{\otimes n}(s_0, ds_1 \ldots ds_{n+1}) \mathbb{1}_B(s_n) \mathbb{1}_{s_{n+1} = \perp}. \tag{23}$$

The terminating state measure is defined as:

$$P_\top : B \in \Sigma_{|\mathcal{X}} \mapsto \sum_{n=1}^{\infty} P_\top^n(B) \tag{24}$$

The following lemma, proved in §F, relates the n-step terminating measures to the n-step measures $P_F^n(s_0, -)$:

Lemma 2. *Let P_F be a transition kernel on $(\bar{\mathcal{S}}, \Sigma)$. For every $n \geq 1$, we have:*

$$P_\top^n(dx) = P_F(x, \{\perp\})P_F^{n-1}(s_0, dx) \tag{25}$$

A.5. Backward reference kernels

Given a reference kernel κ, designing a backward reference kernel κ^b can be done using reverse kernels (Cappé et al., 2009), which we redefine below:

Definition 11 (Reverse kernel). *Let $(\bar{\mathcal{S}}, \Sigma)$ be a measurable space, κ be a transition kernel on $(\bar{\mathcal{S}}, \Sigma)$, and ν be a positive measure on $(\bar{\mathcal{S}}, \Sigma)$. A reverse kernel κ^r_ν associated to ν and κ is a transition kernel over $(\bar{\mathcal{S}}, \Sigma)$ such that:*

$$\nu \otimes \kappa = (\nu\kappa) \otimes \kappa^r_\nu \tag{26}$$

Note how (26) is different from the condition of the backward reference kernel (4). Conveniently, there is a reference measure ν for which the two conditions are equivalent, meaning that the backward reference kernel can be defined as the reverse kernel associated to ν and κ. While there is no guarantee that the reverse kernel exists or is unique in general, existence is guaranteed if $(\bar{\mathcal{S}}, \mathcal{T})$ is a Polish space (e.g. a discrete space, the Euclidean space $\mathbb{R}^n$, hyperrectangles or balls in $\mathbb{R}^n$, or products or disjoint unions of countable families thereof) (Cappé et al., 2009). The following proposition, proved in §F shows that.

Proposition 3. *Given a Polish space $(\bar{\mathcal{S}}, \mathcal{T})$, with source and sink states $s_0, \perp \in \bar{\mathcal{S}}$ such that $\{s_0\}$ and $\{\perp\}$ are both open and closed sets, and a transition kernel κ on $(\bar{\mathcal{S}}, \Sigma)$ satisfying (1), (2) and (7). Let ν be the measure defined by:*

$$\nu = \sum_{n=0}^{N} \kappa^n(s_0, -), \tag{27}$$

and let κ^r_ν be any reverse kernel associated to κ and ν that satisfies the following two conditions:

$$\kappa^r_\nu(s_0, -) = 0 \quad \text{i.e. it's the trivial measure} \tag{28}$$

$$\forall s \in \mathcal{S}, \ \kappa^b(s, \{\bot\}) = 0. \tag{29}$$

Let κ^b be a transition kernel on $(\bar{\mathcal{S}}, \Sigma)$ defined by:

$$\forall s \neq \bot, \ \kappa^b(s, -) = \kappa^r_\nu(s, -),$$

$$\forall B \in \Sigma_{|\mathcal{S}}, \ \kappa^b(\bot, B) = \frac{1}{\nu(\{\bot\})} \nu \otimes \kappa(B \times \{\bot\}),$$

$$\kappa^b(\bot, \{\bot\}) = 0.$$

ν is strictly positive and κ^b satisfies (4). Note that the existence of a reverse kernel satisfying (28) and (29) is guaranteed by Lemma 3 below, proved in §F.

Lemma 3. *Given a Polish space $(\bar{\mathcal{S}}, \mathcal{T})$, with source and sink states $s_0, \bot \in \bar{\mathcal{S}}$ such that $\{s_0\}$ and $\{\bot\}$ are both open and closed sets, and a transition kernel κ on $(\bar{\mathcal{S}}, \Sigma)$ satisfying (1), (2) and (7). Then the measure ν defined by:*

$$\nu = \sum_{n=0}^{N} \kappa^n(s_0, -), \tag{30}$$

If κ^b is a reverse kernel associated to ν and κ, then the kernel κ' defined by:

$$\kappa'(s_0, -) = 0, \tag{31}$$

$$\kappa'(\bot, -) = \kappa^b(\bot, -), \tag{32}$$

$$\forall s' \in \mathcal{S} \setminus \{s_0\}, \ \kappa'(s', \{\bot\}) = 0 \tag{33}$$

$$\forall s' \in \mathcal{S} \setminus \{s_0\}, \ \forall B \in \Sigma, \ \bot \notin B \Rightarrow \kappa'(s', B) = \kappa^b(s, B), \tag{34}$$

is also a reverse kernel associated to ν and κ.

B. Pointed DAGs as measurable pointed graphs

The following example shows that pointed directed acyclic graphs (Bengio et al., 2021b) are a special case of finitely absorbing measurable pointed graphs.

Example 1. *Finite state spaces are special cases of measurable pointed graphs. Let $G = (V, E, s_0, \bot)$ be a pointed directed acyclic graph, where V is the finite set of vertices, $E \subset V \times V$ is the set of directed edges, $s_0 \in V$ is the initial state, and $\bot \in V$ is the sink state.*

The set of vertices V with the discrete topology corresponds to the state space. We can define a transition kernel κ such that for any vertex $s \in V$, and any $B \in \mathcal{P}(V)$, with $\mathcal{P}(V)$ the power set of V, containing all subsets of V:

$$\kappa(s, B) = \sum_{s' \in B} \mathbb{1}_{s \to s' \in E} + \mathbb{1}_{s = \bot, \bot \in B}$$

Using this transition kernel, the measure $B \mapsto \kappa^n(s, B)$ over $(V, \mathcal{P}(V))$ counts the number of trajectories of length n starting at s that ends at a vertex in B in the pointed graph G.

The reverse kernel can be defined for any vertex $s' \in V$ and any $B \in \mathcal{P}(V)$ as:

$$\kappa^b(s', B) = \sum_{s \in B} \mathbb{1}_{s \to s' \in E},$$

and the reference measure ν can be defined as the counting measure (that counts the number of elements in any $B \in \mathcal{P}(V)$).

Since $(V, \mathcal{P}(V))$ is a discrete space, the condition of accessibility in (1) can be verified for only singletons $B = \{s\}$. This condition then corresponds to having a positive number of trajectories of any length $n > 0$ starting at s_0 and ending in s, which is exactly the notion of accessibility in G. The continuity condition is trivially satisfied because the topology is discrete, and (4) is trivially satisfied. Finally, (7) is satisfied given the acyclicity of G.

C. Experimental details

C.1. Approximating the Jensen-Shannon Divergence

Given an unnormalized target reward measure r with respect to the Lebesgue measure on a bounded space $\mathcal{X}$, and a GFlowNet sampler $P_\top$ of terminating states, we approximate the JSD between the learned sampler and the normalized distribution $R(dx) = \frac{1}{\int_\mathcal{X} r(x')dx'} r(x)dx$ as follows:

(1) We sample N points from the the target distribution using rejection sampling, with a uniform distribution as a proposal,
(2) We fit a kernel density estimator (KDE) on the above samples,
(3) We fit a second KDE on N samples from $P_\top$
(4) We use both KDEs to score a fixed set of points defining a discretization of the sample space $\mathcal{X}$,
(5) We normalize both sets of scores in order to obtain valid probability mass functions on the grid,
(6) We evaluate the JSD between the two probability mass functions.

C.2. A synthetic continuous environment

The forward and backward kernels P_F, P_B are defined by their densities p_F and p_B wrt. the reference kernels κ, κ^b.

κ, ν, κ^b satisfy the requirements of a finitely absorbing measurable pointed graph. More notably, all states can be reached from s_0 within $1 + \left\lceil \frac{\sqrt{2}}{\rho} \right\rceil$ steps.

The topology $\mathcal{T}$ on $\bar{\mathcal{S}} = \mathcal{S} \cup \{\bot\}$ is the disjoint union topology on $\{s_0, \bot\}$ and $\mathcal{S}$.

We parametrized $p_F(s_0, -)$ using a mixture of four Beta distributions for both the radius $r \in (0, \rho)$ and the angle $\theta \in (0, \frac{\pi}{2})$. We used a mixture of two Beta distributions for the angle $\theta \in (\theta_{min}(s), \theta_{max}(s))$ when modeling $p_F(s, -)$ and $p_B(s, -)$. The forward policy neural network has an extra output head corresponding to the probability of terminating the trajectory, i.e. $p_F(s_0, \bot)$. The learned probabilities were effectively multiplied by the right Jacobians to account for the support of the Beta distributions ($[0, 1]$) being different from that of θ or r.

Reward density.

The reward measure R was specified using a density r wrt. the Lebesgue measure λ on $\mathcal{X} = (0, 1)^2$. Following Bengio et al. (2021a) and Malkin et al. (2022), the (unnormalized) density is defined for every $x = (x_1, x_2) \in \mathcal{X}$ as:

$$r(x) = 0.1 + 0.5\mathbb{1}_{|x_1 - 0.5| \in (0.25, 5]}\mathbb{1}_{|x_2 - 0.5| \in (0.25, 5]} + 2\mathbb{1}_{|x_1 - 0.5| \in (0.25, 5]}\mathbb{1}_{|x_2 - 0.5| \in (0.3, 4)}$$

Hyperparameters.

We learned the concentration parameters of the Beta distributions, which were restrained to the interval $[0.1, 5.1]$, using a three-layered neural network with 128 units per layer, and leaky ReLU activation for $s \neq s_0$. The parameters corresponding to $p_F(s_0, -)$ were learned separately.

Each iteration consisted of sampling 128 trajectories from the forward policy, and evaluating the TB or the DB loss (with $\alpha = 1$), before taking a gradient step on the learned parameters. We trained the models for 20,000 iterations.

For both the DB and TB losses, we used a learning rate of 10^{-3} for the parameters of $p_F, p_B, p_F(s_0, -)$ (and $\log Z$ for TB, u for DB). The learning rate was annealed using a discount factor of 0.5 every 2500 iterations.

In experiments with learned p_B, both p_F and p_B shared parameters except in the output layer. In DB, p_F and u shared parameters except in the output layer.

Evaluation metric. We approximated the JSD between the learned the terminating state distribution and the target distribution following the scheme described in §C.1.

C.3. Low-dimensional stochastic control

The neural network computing $\mu(\mathbf{x}_t, t)$ had the same architecture as in Zhang & Chen (2022): a pair of 2-layer MLP processing $\mathbf{x}_t$ and a 128-dimensional Fourier feature representation of t, followed by a 3-layer MLP on the concatenation of the features derived from $\mathbf{x}_t$ and from t. We set $\sigma = 5$ for the Gaussians density and $\sigma = 1$ for funnel density. Exploration algorithms added a constant $\frac{\epsilon^2}{T}$ to the sampling policy variance at each step; we used a value of $\epsilon = 0.1$ linearly annealed to
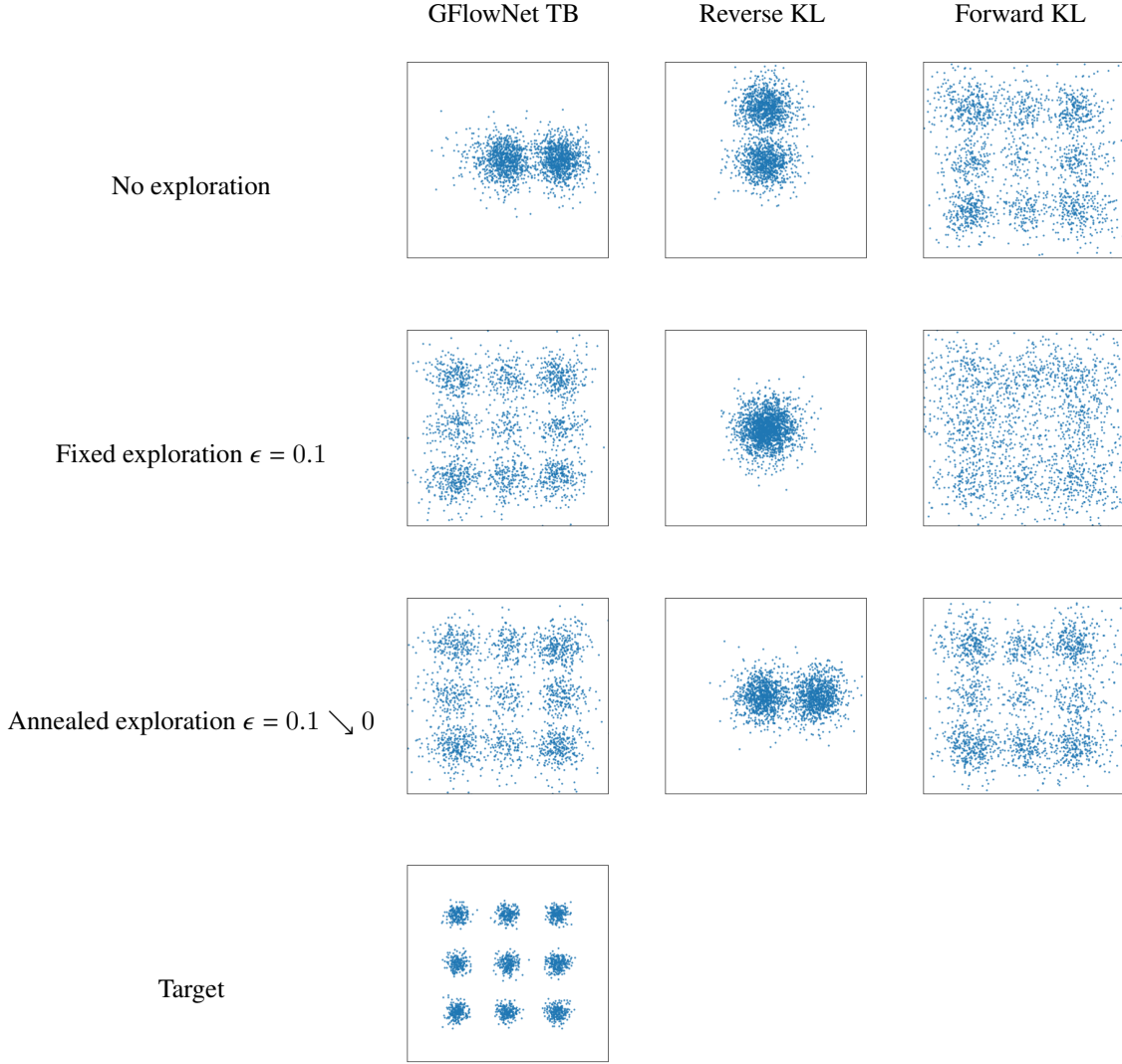
Figure C.1. The target density for 9 Gaussians and samples from models trained with various algorithms trained for 1500 batches. (When trained longer, GFlowNet TB policies with exploration learn to model the modes with higher precision.)

0 over the course of training. All models are trained for 1500 batches of 300 samples with a learning rate of 10^{-2} for the policy and 10^{-1} for $\log Z$ (in the case of GFlowNet algorithms); we found that higher learning rates made optimization unstable. We also observed that the off-policy forward KL and TB algorithms continue to improve with longer training, unlike the on-policy algorithms, which experience mode collapse and cease to discover new areas of the density landscape. Fig. C.1 shows samples from models trained with various algorithms and highlights the importance of exploration.

The simple and importance-weighted estimates of the log-partition function from Zhang & Chen (2022) are defined, in GFlowNet terms, as

$$B = \frac{1}{K} \sum_{i=1}^{K} \log \frac{R(x_T^{(i)}) p_B^{\otimes T}(\tau^{(i)}|x_T^{(i)})}{p_F^{\otimes T}(\tau^{(i)})},$$

$$B_{\mathrm{RW}} = \log \frac{1}{K} \sum_{i=1}^{K} \frac{R(x_T^{(i)}) p_B^{\otimes T}(\tau^{(i)}|x_T^{(i)})}{p_F^{\otimes T}(\tau^{(i)})},$$

where the $\tau^{(i)}$ are K trajectories sampled from P_F, the $x^{(i)}$ are their terminating states, and $p_F^{\otimes T}(\tau^{(i)}), p_B^{\otimes T}(\tau^{(i)}|x_T^{(i)})$ are

Table C.1. Estimation bias of the log-partition function using simple (B) and importance-weighted (B_{RW}) bounds (mean and standard deviation over 10 runs). The **bold** value in each column shows the best result and all those statistically equivalent to it ($p > 0.1$ under a Welch's t-test). Algorithms assuming access to the gradient of the reward (non-black-box) are shown for comparison. Rows marked with [*] require importance weighting for gradient estimation. Cells with – were unstable to optimize. Last three rows from Zhang & Chen (2022).

Black box?			Gaussian mixture ($d = 2$)		Funnel ($d = 10$)	
			B	B_{RW}	B	B_{RW}
✓	Off-policy	GFlowNet TB	$\mathbf{-0.150} \pm 0.019$	$\mathbf{-0.003} \pm 0.011$	$\mathbf{-0.219} \pm 0.020$	$\mathbf{-0.026} \pm 0.020$
✓	Off-policy	Reverse KL[*]	-1.706 ± 0.537	-1.609 ± 0.546	–	–
✓	Off-policy	Forward KL[*]	-0.306 ± 0.036	$\mathbf{-0.001} \pm 0.013$	-2.822 ± 0.576	-0.087 ± 0.081
✓	On-policy	GFlowNet TB	-1.409 ± 0.427	-1.301 ± 0.434	-0.265 ± 0.026	$\mathbf{-0.012} \pm 0.108$
✓	On-policy	Reverse KL	-1.348 ± 0.397	-1.237 ± 0.413	-0.259 ± 0.018	-0.040 ± 0.023
✓	On-policy	Forward KL[*]	-0.254 ± 0.032	-0.007 ± 0.023	-1.384 ± 0.284	-0.034 ± 0.143
✓	Non-SDE	SMC	-0.362 ± 0.293		-0.216 ± 0.157	
✗	On-policy	PIS-NN	-1.691 ± 0.370	-1.192 ± 0.482	-0.098 ± 0.005	-0.018 ± 0.020
✗	Non-SDE	HMC	-1.876 ± 0.527		-0.835 ± 0.257	

the products of forward (resp. backward) Gaussian policy densities along the trajectories. Note that both estimates would equal the true integral of the reward density for a perfect sampler. Identically to Zhang & Chen (2022), we use $K = 2000$ for the 2-dimensional Gaussian mixture dataset and $K = 6000$ for the 10-dimensional funnel dataset.

We show extended results, including both simple and importance-weighted variational bounds, in Table C.1.

C.4. Stochastic control on a torus environment

The transition kernel $\kappa(s, -)$ for any $s = (\tilde{s}, t)$ when $t < T$ is the product of the Lebesgue measure on $[0, 2\pi]^2$ with the Dirac measure at $t + 1$, and $\kappa((\tilde{s}, T), -) = \delta_\perp$ for every $\tilde{s} \in [0, 2\pi]^2$. Similar to §4.2, the reference measure is $\nu = \delta_{s_0} + \sum_{t=1}^{T} \lambda \otimes \delta_t$, where λ is the Lebesgue measure on each copy of the torus $[0, 2\pi]^2$.

We parameterized the densities p_F and p_B with mixtures of independent von Mises distributions defined by a measure of location μ and a measure of concentration κ. We considered two tasks in the torus environment defined by different reward functions.

Synthetic multimodal task. For this task, we designed a reward density with six modes on the torus surface:

$$R_6(\psi, \varphi) = (\sin(3\psi) + \cos(2\varphi) + 2)^3.$$

Molecule conformation task. In this task, we define the reward function using the energy $\mathcal{E}$ of an alanine dipeptide molecule, which depends on the conformation of the molecule C (spatial arrangement of its atoms). This conformation can be efficiently parametrizeded using internal coordinates: bond lengths, bond angles, and torsion angles (Jing et al., 2022; Thiede et al., 2022). For alanine dipeptide, there are four torsion angles largely influencing the energy (see Fig. C.2). In our experiments, the GFlowNet generates values for the angles ψ and φ while keeping all other coordinates fixed. In this way, the support of the reward function remains a torus, and its values are proportional to the Boltzmann distribution with energy $\mathcal{E}$:

$$R_{AD}(\psi, \varphi) = \exp(-\mathcal{E}(C(\psi, \varphi))),$$

The plots in Fig. 4 show the results of training a GFlowNet on a toroidal space with a continuous synthetic multimodal reward function R_6 (see text) and a reward function defined by the Boltzmann distribution of the alanine dipeptide molecule R_{AD}. The images represent the density over a discretization of the space $[0, 2\pi]^2$, obtained after fitting KDE with 100,000 samples. The samples to fit the reward densities (Fig. 4(**a-c**)) were obtained via rejection sampling, and the GFlowNet densities (Fig. 4**b-d**) use GFlowNet samples from the learned distribution over terminating states $P_\top$.

Results. To evaluate the performance of the GFlowNet trained with the TB loss, we calculated the Jensen-Shannon divergence
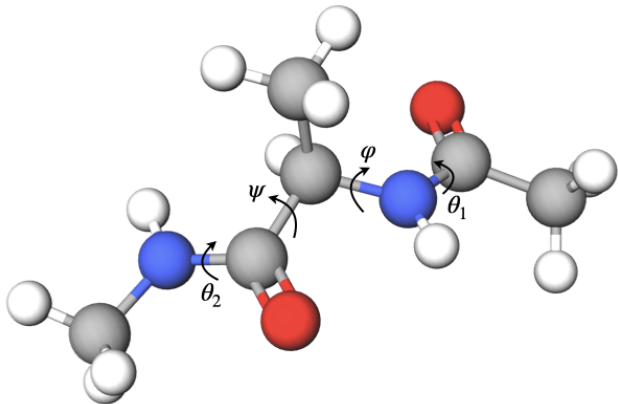
Figure C.2. Alanine dipeptide 3D structure. Torsion angles ψ, φ, θ_1, θ_2 have the biggest impact on the energy of the molecule. A pair of torsion angles φ and ψ can take any values $\in [0, 2\pi]$, while θ_1 and θ_2 can be either close to 0 or π due to energy barriers (Mironov et al., 2018). The image is rendered using MolView (Bergwerf, 2014)

(see §C.1 for details about its estimation) between the learned terminating state distribution $P_\top$ and the normalized reward distribution. We provide a visual representation of learned and reward distributions in Fig. 4. Quantitatively, the GFlowNet achieved a JSD of 0.063 for the synthetic multimodal task and 0.009 for the molecule conformation task. These results show that the generalized GFlowNet can model probability densities over non-Euclidean spaces.

Hyperparameters. We modeled both p_F and p_B with 5-layer perceptrons with 512 hidden units per layer, training the full set of parameters of each model separately. These models output, for each angle ψ and φ, the location μ_i and concentration κ_i of 5 independent von Mises distributions, mixed with learned weights w_i. To take into account the topology of the torus, we encoded input angles with trigonometric transformations $(\sin(k\psi), \cos(k\psi), k = 1, \ldots, 5$, for both angles ψ and φ). We used a learning rate of 10^{-5} for the model parameters and 10^{-2} for $\log Z$, updating the parameters with batches of 100 trajectories of length $T = 10$. With the synthetic reward, the model converged in about 5,000 iterations; in the molecular conformation task, we trained for 40k iterations.

C.5. Posterior over continuous parameters in Bayesian structure learning

Bayesian Networks. Recall that a Bayesian Network is a probabilistic model, where the joint distribution over d random variables $\{X_1, \ldots, X_d\}$ factorizes according to a directed acyclic graph (DAG) G as:

$$P(X_1, \ldots, X_d; \theta, G) = \prod_{i=1}^{d} P(X_i \mid \mathrm{Pa}_G(X_i); \theta_i),$$

where $\mathrm{Pa}_G(X_i)$ is the set of parents of X_i in G, and θ_i is the set of parameters for the conditional probability distribution of X_i. We denote by $\theta = \{\theta_1, \ldots, \theta_d\}$ the set of all the parameters of this model.

We assume that $\theta \in \Theta_G$, where Θ_G is the space of all parameters for the Bayesian Network. Note that this space of parameters depends on the structure G of the Bayesian Network. For example, the Bayesian Network where all the random variables are mutually independent (corresponding to G being empty) has fewer parameters than another Bayesian Network that encodes dependencies between those random variables. We will also denote by $\mathcal{G}$ the space of DAG over d nodes; the number of elements in this space grows super-exponentially with d.

Linear Gaussian model. In order to compute the exact posterior distribution $P(G, \theta \mid \mathcal{D})$ in closed-form, we consider here a linear-Gaussian model for the parametrization of the conditional probability distributions appearing in the Bayesian Network. More precisely, the conditional probability distribution is given by

$$P(X_i \mid \mathrm{Pa}_G(X_i); \theta_i) = \mathcal{N}(X_i \mid \mu_i, \sigma^2) \qquad \text{where} \qquad \mu_i = \sum_{X_j \in \mathrm{Pa}_G(X_i)} \theta_{ij} X_j.$$

In other words, X_i follows a Normal distribution, whose mean μ_i is given by a linear combination of its parents, and with fixed variance σ^2. For this class of models,

$$\Theta_G \simeq \bigtimes_{i=1}^{d} \mathbb{R}^{|\mathrm{Pa}_G(X_i)|}$$

18

GFlowNet over a mixed state space. We are using the GFlowNet in order to approximate the joint posterior distribution $P(G, \theta \mid \mathcal{D})$, and therefore its terminating states have the form (G, θ), where $G \in \mathcal{G}$ is a DAG, and $\theta \in \Theta_G$ are the associated parameters. Unlike Nishikawa-Toomey et al. (2022), which uses Variational Bayes to update the distribution over parameters θ, we model the distribution over both the graphs and parameters using a single GFlowNet.

The generation of a terminating state follows 2 phases: during the first phase, the DAG G is constructed by adding one edge at a time, starting from the empty graph, following the structure of DAG-GFlowNet (Deleu et al., 2022). To reach a graph G with k edges, we therefore are taking k steps in the GFlowNet. The states traversed during this first phase have no parameters associated to them; we denote by $(G, \sharp) \in \mathcal{S}$ such an (intermediate) state, where $\sharp \notin \Theta_{G'}$ for any $G' \in \mathcal{G}$ is a symbol indicating that G has no corresponding parameters.

Then once we have finished adding edges (in practice, this decision is made by selecting a special "stop" action), we sample the parameters $\theta \in \Theta_G$ associated to G by taking a final step in the GFlowNet to reach the terminating state $(G, \theta) \in \mathcal{X}$. Since the space of parameters depends on the graph G, we define the state space of the GFlowNet as

$$\mathcal{S} = \bigcup_{G \in \mathcal{G}} \{G\} \times \bar{\Theta}_G \qquad \text{and} \qquad \mathcal{X} = \bigcup_{G \in \mathcal{G}} \{G\} \times \Theta_G,$$

where $\bar{\Theta}_G = \Theta_G \cup \{\sharp\}$ indicates the space of parameters, augmented with the special symbol $\sharp$. All the states in this state space are guaranteed to be accessible from the initial state $(G_0, \sharp)$, where G_0 is the empty graph.

Reference kernel. Given a DAG G, the measure $\kappa((G, \sharp), -)$ is the sum of a discrete measure (to transition to another intermediate state $(G', \sharp)$) and a continuous measure (to transition to a terminating state (G, θ)). We can write this measure as

$$\kappa((G, \sharp), -) = \sum_{G' \in \mathrm{Ch}(G)} \delta_{(G', \sharp)} + (\delta_G \otimes \lambda_{\Theta_G}),$$

where $\mathrm{Ch}(G)$ represents the children of G in DAG-GFlowNet (Deleu et al., 2022), i.e. the graphs G' obtained by adding an edge to G, and λ_{Θ_G} is the Lebesgue measure over Θ_G. Moreover, we also have $\kappa((G, \theta), -) = \delta_\perp$ for all terminating state $(G, \theta) \in \mathcal{X}$; in other words, there is no transition from a terminating state other than to the sink state.

The backward reference kernel κ^b on the other hand is simpler: it is always a discrete transition kernel, regardless of the state. We have

$$\kappa^b((G, \theta), -) = \delta_{(G, \sharp)} \qquad \text{and} \qquad \kappa^b((G', \sharp), -) = \sum_{G \in \mathrm{Pa}(G')} \delta_{(G, \sharp)},$$

where $\mathrm{Pa}(G')$ are the parents of G' in DAG-GFlowNet, i.e. they are the graphs obtained by removing a single edge from G'.

Forward transition probability. In order to define the P_F, we consider 2 cases: either we have a distribution of the form $P_F(G' \mid G)$, where G' is the result of adding an edge to G, or a distribution of the form $P_F(\theta \mid G)$. Note that here we are using a slight abuse of notation, where $P_F(G' \mid G)$ (resp. $P_F(\theta \mid G)$) represents $P_F((G', \sharp) \mid (G, \sharp))$ (resp. $P_F((G, \theta) \mid (G, \sharp))$). The distribution $P_F(\theta \mid G)$ is parametrized by a Normal distribution, whose mean and (diagonal) covariance are returned by a neural network. Similar to (Deleu et al., 2022), P_B is fixed to the uniform distribution.

Data generation. We sampled a dataset $\mathcal{D}$ as follows: (1) we first generated a DAG G^* from an Erdös-Renyi model, then (2) we sampled the parameters θ^* of the conditional probability distributions from a Normal distribution, each edge having a weight $\theta_{ij}^* \sim \mathcal{N}(0, 1)$, and finally (3) we sampled $N = 100$ datapoints from the Bayesian Network described above with (G^*, θ^*), using ancestral sampling. Note that the ground-truths G^* and θ^* are unknown to the GFlowNet, and it only uses the observations from $\mathcal{D}$.

Evaluations. To evaluate the quality of the approximation learned by the GFlowNet, and to compare it against the baseline methods based on variational inference (Lorch et al., 2021; Cundy et al., 2021), we study the distribution over graphs (discrete component) and the distribution over graphs (continuous component) separately. Recall that since we assume that our model is linear-Gaussian over small graphs ($d \leq 5$), we can compute the exact posterior distribution $P(G, \theta \mid \mathcal{D})$ in closed form.

In Table 3, we compared the edge marginals estimated using the posterior approximations to the exact edge marginals. For

Table C.2. Comparison between GFlowNet and other methods based on variational inference on the Bayesian structure learning task, for different marginals of interest of the distribution over graphs $P(G \mid \mathcal{D})$.

		RMSE			Pearson's r		
Number of variables (d)		3	4	5	3	4	5
Edges	BCD Nets	–	2.13×10^{-1}	2.61×10^{-1}	–	0.8578	0.7886
	DiBS	3.28×10^{-1}	2.95×10^{-1}	3.15×10^{-1}	0.6903	0.7085	0.7170
	GFlowNet	$\mathbf{1.50 \times 10^{-2}}$	$\mathbf{1.61 \times 10^{-2}}$	$\mathbf{1.80 \times 10^{-2}}$	$\mathbf{0.9993}$	$\mathbf{0.9990}$	$\mathbf{0.9990}$
Paths	BCD Nets	–	2.59×10^{-1}	3.08×10^{-1}	–	0.8378	0.7500
	DiBS	3.50×10^{-1}	3.35×10^{-1}	3.48×10^{-1}	0.6951	0.7080	0.7020
	GFlowNet	$\mathbf{3.39 \times 10^{-3}}$	$\mathbf{1.07 \times 10^{-2}}$	$\mathbf{1.99 \times 10^{-2}}$	$\mathbf{1.0000}$	$\mathbf{0.9996}$	$\mathbf{0.9989}$
Markov blanket	BCD Nets	–	3.02×10^{-1}	3.49×10^{-1}	–	0.8831	0.7864
	DiBS	3.88×10^{-1}	3.80×10^{-1}	4.45×10^{-1}	0.7840	0.7892	0.6888
	GFlowNet	$\mathbf{2.14 \times 10^{-2}}$	$\mathbf{2.38 \times 10^{-2}}$	$\mathbf{2.83 \times 10^{-2}}$	$\mathbf{0.9986}$	$\mathbf{0.9982}$	$\mathbf{0.9980}$

any pair of random variables (X_i, X_j), this means evaluating the following marginals

$$P(X_i \to X_j \mid \mathcal{D}) = \sum_{G \mid X_i \in \mathrm{Pa}_G(X_j)} P(G \mid \mathcal{D}).$$

To estimate this marginal using samples $\{(G_k, \theta_k)\}_{k=1}^{K}$ from the GFlowNet (or from the variational inference methods), we can simply use the sample graphs G_k in order to get an empirical approximation of the maginal posterior $P(G \mid \mathcal{D})$. In other words,

$$\hat{P}(X_i \to X_j) = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}(X_i \to X_j \in G_k)$$

We then report the root mean-square error (RMSE) between the edge marginals estimated using the posterior approximations, and those computed using the exact posterior:

$$\mathrm{RMSE}(\hat{P}, P) = \left(\frac{1}{d(d-1)} \sum_{i \neq j} \left(\hat{P}(X_i \to X_j) - P(X_i \to X_j \mid \mathcal{D}) \right)^2 \right)^{1/2}.$$

In Table C.2, we also report the RMSE for other marginals: the marginal of having a directed path between two nodes $P(X_i \rightsquigarrow X_j \mid \mathcal{D})$, as well as the marginal of node X_i being in the Markov blanket of X_j $P(X_i \in \mathrm{MarkovBlanket}(X_j) \mid \mathcal{D})$. In addition to the RMSE between those marginals, we also report the Pearson correlation coefficient, as in (Deleu et al., 2022). Note that no metric is reported on graphs over $d = 3$ nodes for BCD Nets (Cundy et al., 2021) due to technical reasons (the method is not applicable for graphs smaller than 4 nodes).

To evaluate the accuracy of the approximation on the continuous part of the distribution, we report in Table 3 the (average) negative log-probability of the sampled parameters θ_k from the different approximations (GFlowNet, DiBS (Lorch et al., 2021), and BCD Nets (Cundy et al., 2021)) against the exact posterior distribution $P(\theta \mid G, \mathcal{D})$. More precisely, we compute

$$\mathrm{Measure}_{\theta}(\hat{P}, P) = -\frac{1}{K|\mathcal{D}|} \sum_{k=1}^{K} \log P(\theta_k \mid G_k, \mathcal{D})$$

In other words, the lower this metric is, the more likely the samples θ_k from those approximations are under the exact posterior distribution over parameters.

C.6. Connections with diffusion models

We train a diffusion model-specified GFlowNet with $T = 100$ for $200,000$ steps. This is much shorter than other state-of-the-art work (such as Lipman et al. (2022)) and takes less than 3 days on a single V100 GPU. All NLL results are computed
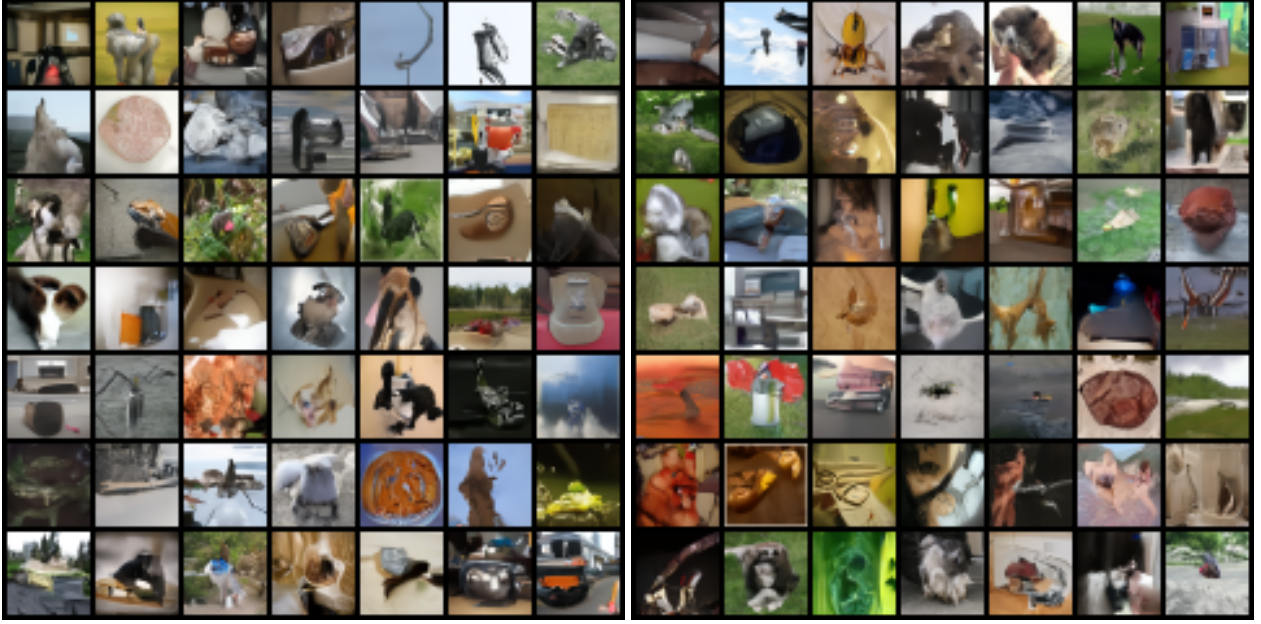
Figure C.3. Generated samples from MLE-GFN on ImageNet-32 dataset.

in bits per dimension (BPD). We use $50,000$ generated samples to compute the FID score for evaluating the sample quality. The Adam learning rate is 2×10^{-4} for the forward policy and 2×10^{-5} for the backward policy. The parameter of backward policy is $\{\phi_i\}_{i=1}^T$, where the variance coefficient β_i satisfies $\beta_i = \bar{\beta}_i \cdot \exp(\phi_i)$ and $\bar{\beta}_i$ is the original variance coefficient used in Ho et al. (2020). For details about the MLE-GFN algorithm, we refer to Zhang et al. (2023a). Fig. C.3 shows examples of images generated by the algorithm.

D. Additional lemmas and propositions

In this section, we will write and prove lemmas and propositions that would help explain and shorten the proofs of the main results presented in the main text and in §A, which we provide in §E.

The following lemma ensures that in a measurable pointed graph, $\{s_0\}$ is not accessible.

Lemma 4. $\forall s \in \mathcal{S}, \; \kappa(s, \{s_0\}) = 0$

Proof We will present a proof by contradiction.

Let $\mathcal{N} = \{s \in \bar{\mathcal{S}}, \; \kappa(s, \{s_0\}) > 0\}$, and assume that $\mathcal{N} \neq \emptyset$. $(0, \infty)$ being an open set, and $s \mapsto \kappa(s, \{s_0\})$ continuous, this means that $\mathcal{N} \in \bar{\mathcal{T}}$ (i.e. it is open). From (1), it follows that there is some $n \geq 0$ such that $\kappa^n(s_0, \mathcal{N}) > 0$.

Applying (19), we obtain:

$$\kappa^{n+1}(s_0, \{s_0\}) = \int_{\bar{\mathcal{S}}} \kappa^n(s_0, ds')\kappa(s', \{s_0\}) \geq \int_{\mathcal{N}} \kappa^n(s_0, ds')\kappa(s', \{s_0\}) > 0.$$

Writing, for all $m > 1$:

$$\kappa^{m(n+1)}(s_0, \{s_0\}) = \int_{\bar{\mathcal{S}}} \kappa^{(m-1)(n+1)}(s_0, ds')\kappa^{n+1}(s', \{s_0\}) \geq \int_{\{s_0\}} \kappa^{(m-1)(n+1)}(s_0, ds')\kappa^{n+1}(s', \{s_0\})$$
$$= \kappa^{(m-1)(n+1)}(s_0, \{s_0\})\kappa^{n+1}(s_0, \{s_0\}),$$

it follows from a simple induction that $\forall m \geq 1, \; \kappa^{m(n+1)}(s_0, \{s_0\}) > 0$, which contradicts (7).

$\mathcal{N}$ is thus necessarily empty. ∎

The following lemma ensures a compatibility between the definition of the terminating states X and $\kappa^b(\perp, -)$:

Lemma 5. *The support of $\kappa^b(\perp, -)$ is the closure of X.*

Proof Let s be an element in the support of $\kappa^b(\perp, -)$. By definition of the support, it means that for any $B \in \mathcal{T}$ containing s, there is some $B' \subseteq B$ such that $B' \subseteq X$. In particular, $B \cap X \neq \emptyset$. This means that s is a point of closure of X.

Conversely, let s be a point of closure of X. Given any open set $B \in \mathcal{T}$ containing s, s being a closure point means that $B \cap X$ (which is measurable) is non-empty. Following (4), we get:

$$\nu(\{\perp\})\kappa^b(\perp, B \cap X) = \int_{\bar{S}} \mathbb{1}_{B \cap X}(s')\nu(ds')\kappa(s', \{\perp\})$$

The RHS of the previous equality is positive because ν is a strictly positive measure and $\kappa(s', \{\perp\}) > 0$ for every $s' \in X$, following Def. 2. Hence $\kappa^b(\perp, B \cap X) > 0$. It follows that $\kappa^b(\perp, B) \geq \kappa^b(\perp, B \cap X) > 0$. Meaning that s is indeed within the support of $\kappa^b(\perp, -)$. ∎

The proof of Lemma 2 relies on an the following intermediary lemma, which relates the n-step trajectory measures $P_F^{\otimes n}(s, -)$ to the n-step measures $P_F^n(s, -)$ defined by (19).

Lemma 6. *Let P_F be a transition kernel on $(\bar{S}, \Sigma)$. For every $s \in \bar{S}$, $n \geq 0$, and for any bounded measurable function $f : \bar{S} \to \mathbb{R}$, we have:*

$$\int_{\bar{S}^{n+1}} f(s')P_F^{\otimes n}(s, ds_1 \ldots ds_n ds') = \int_{\bar{S}} f(s')P_F^n(s, ds') \tag{35}$$

Proof We prove the lemma by induction on n. First, for $n = 0$, using (21)

$$\int_{\bar{S}} f(s')P_F^{\otimes 0}(s, ds') = f(s) = \int_{\bar{S}} f(s')P_F^0(s, ds').$$

Then, assuming that (35) is satisfied for some $n \geq 0$, we get:

$$\begin{aligned}
\int_{\bar{S}^{n+2}} f(s')P_F^{\otimes n+1}(s, ds_1 \ldots ds_{n+1}ds') &= \int_{\bar{S}^{n+2}} f(s')P_F^{\otimes n}(s, ds_1 \ldots ds_{n+1})P_F(s_{n+1}, ds') \\
&= \int_{\bar{S}^{n+1}} \underbrace{\int_{\bar{S}} f(s')P_F(s_{n+1}, ds')}_{\triangleq g(s_{n+1})} P_F^{\otimes n}(s, ds_1 \ldots ds_{n+1}) \\
&= \int_{\bar{S}} g(s_{n+1})P_F^n(s, ds_{n+1}) = \iint_{\bar{S} \times \bar{S}} f(s')P_F(s_{n+1}, ds')P_F^n(s, ds_{n+1}) \\
&= \int_{\bar{S}} f(s')P_F^{n+1}(s, ds_{n+1}),
\end{aligned}$$

where we applied the inductive hypothesis to a new bounded and measurable[3] function g, and applied the recursive definition of P_F^{n+1}. ∎

The next proposition is crucial in proving Thm. 1.

Proposition 4. *Let $F = (\mu, P_F)$ be a flow over G (i.e. F satisfies the flow-matching conditions (9)), then the measure u defined by:*

$$u : B \in \Sigma_{|S} \mapsto \sum_{n=0}^{\infty} P_F^n(s_0, B), \tag{36}$$

is finite, and satisfies for all $B \in \Sigma_{|S}$

$$\mu(\{s_0\})u(B) = \mu(B) \tag{37}$$

---

[3]This can be seen by writing $f = f^+ - f^-$, where f^+, f^- are non-negative, writing each of f^+, f^- as a limit of step functions, and using the monotone convergence theorem.

Proof First, using a simple recursion, we show that $\forall n \geq N$, $P_F^n(s_0, -) = \delta_\perp$. The base case ($n = N$) is satisfied as a consequence of Lemma 1, and the fact that the measurable pointed graph is finitely absorbing. Assuming it holds for some $n \geq N$, going back to the definition of the n-step measure, we have for every $B \in \Sigma$:

$$P_F^{n+1}(s_0, B) = \int_{\bar{S}} P_F^n(s_0, ds) P_F(s, B) = \int_{\bar{S}} \delta_\perp(ds) P_F(s, B) = P_F(\perp, B) = \delta_\perp(B),$$

where the last equality stems from the absolute continuity of $P_F(\perp, -)$ wrt. $\kappa(\perp, -)$ and (7).

This shows that for every $B \in \Sigma_{|S}$:

$$u(B) = \sum_{n=0}^{N-1} P_F^n(s_0, B).$$

Which shows the measure u is finite.

Next, we partition S into N disjoint sets $S_0, \ldots, S_{N-1}$, where:

$$s \in S_n \Leftrightarrow n = \max\{m \in \mathbb{N}_0 : \forall B \in \mathcal{T}, s \in B \Rightarrow P_F^m(s_0, B) > 0\}$$

$S_n \in \Sigma$ given that $S_n = S_n' \setminus \bigcup_{k=1}^\infty S_{n+k}'$, where S_n' is the support of $P_F^n(s_0, -)$, which is known to be a closed set, and hence measurable.

Writing any $B \in \Sigma_{|S}$ as:

$$B = \bigcup_{n=0}^{N-1} B \cap S_n,$$

and using the additivity property of the measures u and μ, then proving (37) for all $B \in \Sigma_{|S}$, amounts to proving it for all $B \in \Sigma_{|S_n}$ for all $n \in \{0, \ldots, N-1\}$, given that the sets S_i are themselves measurable. We prove this by strong induction on n.

Base case: For $n = 0$, $S_0 = \{s_0\}$, and $\Sigma_{|S_0} = \{\{s_0\}\}$.

$$u(\{s_0\}) = P_F^0(s_0, \{s_0\}) = \delta_{s_0}(\{s_0\}) = 1,$$

Hence (37) is satisfied for $B = \{s_0\}$.

Induction step: Assume that for some $n \geq 0$, (37) is satisfied for all $B \in \Sigma_{|S_m}$ for all $m \leq n$, and let $B \in \Sigma_{|S_{n+1}}$.

Define $B' = \{s' \in S : P_F(s', B) > 0\}$. We first show that $B' \subseteq \bigcup_{m=0}^n S_m$. If there is $s' \in B'$ and $n' > n$ such that $s' \in S_{n'}$, then for any open set $\tilde{B}$ (i.e. $\tilde{B} \in \mathcal{T}$) containing s', $P_F^{n'}(s_0, \tilde{B}) > 0$. $\tilde{s} \mapsto P_F(\tilde{s}, B)$ being a continuous function, it follows that B' itself is open. Hence $P_F^{n'}(s_0, B') > 0$. And noting that:

$$P_F^{n'+1}(s_0, B) = \int_{\bar{S}} P_F^{n'}(s_0, ds') P_F(s', B) \geq \int_{B'} P_F^{n'}(s_0, ds') P_F(s', B) > 0,$$

which would contradict the fact that $B \subseteq S_{n+1}$, given that $n' + 1 > n + 1$. This shows that $B' \subseteq \bigcup_{m=0}^n S_m$.

Hence:

$$\mu(\{s_0\}) u(B) = \mu(\{s_0\}) \sum_{m=0}^{n+1} P_F^m(s_0, B) = \mu(\{s_0\}) \underbrace{\delta_{s_0}(B)}_{=0} + \mu(\{s_0\}) \sum_{m=0}^n P_F^{m+1}(s_0, B)$$

$$= \mu(\{s_0\}) \sum_{m=0}^n \int_{B'} P_F^m(s_0, ds') P_F(s', B) = \int_{B'} \mu(\{s_0\}) u(ds') P_F(s', B)$$

$$= \int_{B'} \mu(ds') P_F(s', B) = \mu(B),$$

where the last two equalities stem from the induction hypothesis and the flow matching conditions respectively. ∎

The following lemma, relates the flow measures at the source and sink states to the reward measure and shows that the source and sink flows correspond to the "total reward" $R(\mathcal{X})$ (called the partition function with discrete GFlowNets):

Lemma 7. *Let $F = (\mu, P_F)$ be a flow over G satisfying reward-matching conditions in (10) w.r.t. a measure R, then*

$$\mu(\{s_0\}) = \mu(\{\perp\}) = R(\mathcal{X}). \tag{38}$$

Proof Applying (10) to the function $f : x \in \mathcal{X} \mapsto 1$, we get:

$$\int_{\mathcal{X}} R(dx) = \int_{\mathcal{X}} \mu(dx) P_F(x, \{\perp\}). \tag{39}$$

Additionally, from (8), we get $\forall s \in \mathcal{S} \setminus \mathcal{X}, \ \kappa(s, \{\perp\}) = 0$. It follows from the absolute continuity requirements of P_F that $\forall s \in \mathcal{S} \setminus \mathcal{X}, \ P_F(s, \{\perp\}) = 0$. Hence:

$$\int_{\mathcal{X}} R(dx) = \int_{\mathcal{S}} \mu(ds) P_F(s, \{\perp\}) = \iint_{\mathcal{S} \times \bar{\mathcal{S}}} \mathbb{1}_{s'=\perp} \mu(ds) P_F(s, ds') = \int_{\bar{\mathcal{S}}} \mathbb{1}_{s'=\perp} \mu(ds') = \mu(\{\perp\}),$$

where the last line follows from (9). This shows that $\mu(\{\perp\}) = R(\mathcal{X})$.

Note that as a consequence of Lemma 4 and the absolute continuity requirement, a P_F satisfies:

$$\forall s \in \bar{\mathcal{S}}, P_F(s, \{s_0\}) = 0 \tag{40}$$

Next, following (9) and (40), we have:

$$\iint_{\mathcal{S} \times \bar{\mathcal{S}}} \mu(ds) P_F(s, ds') = \iint_{\mathcal{S} \times \bar{\mathcal{S}}} \mathbb{1}_{s' \neq s_0} \mu(ds) P_F(s, ds') = \int_{\bar{\mathcal{S}}} \mathbb{1}_{s' \neq s_0} \mu(ds') = \mu(\bar{\mathcal{S}}) - \mu(\{s_0\}). \tag{41}$$

On the other hand, because each $P_F(s, -)$ is a probability measure on $\bar{\mathcal{S}}$:

$$\iint_{\mathcal{S} \times \bar{\mathcal{S}}} \mu(ds) P_F(s, ds') = \int_{\mathcal{S}} \left(\int_{\bar{\mathcal{S}}} P_F(s, ds') \right) \mu(ds) = \int_{\mathcal{S}} \mu(ds) = \mu(\mathcal{S}) \tag{42}$$

Subtracting (41) from (42), we get:

$$\mu(\{s_0\}) = \mu(\bar{\mathcal{S}}) - \mu(\mathcal{S}) = \mu(\{\perp\}) = R(\mathcal{X})$$

$\blacksquare$

The following lemma is crucial to prove Prop. 2.

Lemma 8. *If (P_F, P_B, Z) satisfy the trajectory balance conditions wrt. R, then for any $n \in \{0, ..., N\}$, and for any measurable bounded function $f : \mathcal{S} \to \mathbb{R}$:*

$$Z \int_{\mathcal{S}} f(s) P_F^n(s_0, ds) P_F(s, \{\perp\}) = \int_{\mathcal{S}} f(s) P_B^n(s, \{s_0\}) R(ds) \tag{43}$$

Proof Using Lemma 6, we have:

$$
\begin{aligned}
Z \int_{\mathcal{S}} f(s) P_F^n(s_0, ds) P_F(s, \{\perp\}) &= Z \int_{\bar{\mathcal{S}}^{n+1}} \mathbb{1}_{s \neq \perp} f(s) P_F(s, \{\perp\}) P_F^{\otimes n}(s_0, ds' ds_1 \ldots ds_{n-1} ds) \\
&= Z \int_{\bar{\mathcal{S}}^{n+2}} \mathbb{1}_{s \neq \perp, s_{n+1}=\perp} f(s) P_F^{\otimes n+1}(s_0, ds' ds_1 \ldots ds_{n-1} ds ds_{n+1}) \\
&= \int_{\bar{\mathcal{S}}^{n+2}} \mathbb{1}_{s \neq \perp} \mathbb{1}_{s''=s_0} f(s) R(ds) P_B^{\otimes n}(s, ds' ds_{n-1} \ldots ds_1, ds'') \\
&= \int_{\bar{\mathcal{S}}} f(s) \mathbb{1}_{s \neq \perp} R(ds) \int_{\bar{\mathcal{S}}^{n+1}} \mathbb{1}_{s''=s_0} P_B^{\otimes n}(s, ds' ds_{n-1} \ldots ds_1, ds'') \\
&= \int_{\bar{\mathcal{S}}} f(s) \mathbb{1}_{s \neq \perp} R(ds) \int_{\bar{\mathcal{S}}} \mathbb{1}_{s''=s_0} P_B^n(s, ds'') \\
&= \int_{\mathcal{S}} f(s) R(ds) P_B^n(s, \{s_0\})
\end{aligned}
$$

■

The following proposition generalizes Lemma 5 of (Bengio et al., 2021b) to measurable pointed graphs, and is crucial in proving Prop. 2

Proposition 5. *Let P_B be a backward kernel over G. Let $P_{B,T}$ be the measure defined by:*

$$P_{B,T}(s) = \sum_{n=0}^{\infty} P_B^n(s, \{s_0\}) \tag{44}$$

We have $\forall s \in \mathcal{S}$:

$$P_{B,T}(s) = 1 \tag{45}$$

Proof First, using a simple recursion, we show that $\forall n \geq N$, $P_B^n(s, \{s_0\}) = 0$. The base case $(n = N)$ is trivially satisfied because the measurable pointed graph has maximal trajectory length N and $P_B(s_0, -)$ is the trivial measure (i.e. it assigns zero to every measurable set), given that it is absolutely continuous wrt. $\kappa^b(s_0, -)$. Assuming it holds for some $n \geq N$, we have:

$$P_B^{n+1}(s, s_0) = \int_{\bar{\mathcal{S}}} P_B(s, ds') P_B^n(s', \{s_0\}) = \int_{\bar{\mathcal{S}}} P_B(s, ds').0 = 0$$

This shows that for every $s \in \mathcal{S}$:

$$P_{B,T}(s) = \sum_{n=0}^{N-1} P_B^n(s, ds_0).$$

Which shows the measure $P_{B,T}$ is finite.

Next, we partition $\mathcal{S}$ into N disjoint sets $\mathcal{S}_0, \ldots, \mathcal{S}_{N-1}$, where:

$$s \in \mathcal{S}_n \Leftrightarrow n = \max\{m \in \mathbb{N}_0 : P_B^m(s, \{s_0\}) > 0\}$$

$\mathcal{S}_n \in \Sigma$ given that $\mathcal{S}_n = \mathcal{S}_n' \setminus \bigcup_{k=0}^{\infty} \mathcal{S}_{n+k}'$, where $\mathcal{S}_n'$ is the support of $P_B^n(-, \{s_0\})$, which is known to be a closed set, and hence measurable.

Writing :

$$\mathcal{S} = \bigcup_{n=0}^{N-1} \mathcal{S}_n,$$

Then proving (45) for all $s \in \mathcal{S}$, amounts to proving it for all $s \in \mathcal{S}_n$ for all $n \in \{0, \ldots, N-1\}$. We prove this by strong induction on n.

Base case: For $n = 0$, $\mathcal{S}_0 = \{s_0\}$, and

$$P_{B,T}(s_0) = P_B^0(s_0, \{s_0\}) = \delta_{s_0}(\{s_0\}) = 1,$$

Hence it is satisfied for n = 0.

Induction step: Assume that for some $n \geq 0$, (45) is satisfied for all $s \in \mathcal{S}_m$ for all $m \leq n$, and let $s \in \mathcal{S}_{n+1}$.

Define $B_s = \{s' \in \mathcal{S} : \forall B \in \mathcal{T}, s' \in B => P_B(s, B) > 0\}$. We first show by contradiction that $B_s \subseteq \bigcup_{m=0}^{n} \mathcal{S}_m$. If there is $s' \in B_s$ and $n' > n$ such that $s' \in \mathcal{S}_{n'}$, then $P_B^{n'}(s', \{s_0\}) > 0$, and by continuity of $\tilde{s} \mapsto P_B^{n'}(\tilde{s}, \{s_0\})$, there exists an open set $\tilde{B}$ (i.e. $\tilde{B} \in \mathcal{T}$) containing s' such that $P_B^{n'}(\tilde{s}, \{s_0\}) > 0$ for all $\tilde{s} \in \tilde{B}$. Hence:

$$P_B^{n'+1}(s, \{s_0\}) = \int_{\bar{\mathcal{S}}} P_B(s, ds') P_B^{n'}(s', \{s_0\}) \geq \int_{\tilde{B}} P_B(s, ds') P_B^{n'}(s', \{s_0\}) > 0,$$

which would contradict the fact that $s \in \mathcal{S}_{n+1}$, given that $n' + 1 > n + 1$.

Hence:

$$P_{B,T}(s) = \sum_{m=0}^{n+1} P_B^m(s, \{s_0\}) = \sum_{m=1}^{n+1} P_B^m(s, \{s_0\}) \quad \text{given that } s \neq s_0$$

$$= \sum_{m=1}^{n+1} \int_{s' \in B_s} P_B(s, ds') P_B^{m-1}(s', \{s_0\}) = \int_{s' \in B_s} P_B(s, ds') \underbrace{\sum_{m=0}^{n} P_B^m(s', \{s_0\})}_{=1} = \int_{s' \in B_s} P_B(s, ds') = 1$$

∎

E. Proofs of results in the main text

We are now ready to prove the main theorem of the paper, which we restate here:

Theorem 1. *If $F = (\mu, P_F)$ is a flow over G, that satisfies the reward matching conditions* (10) *wrt. a measure R, then the terminating state measure:*

$$P_T : B \in \Sigma_{|X} \mapsto \sum_{n=1}^{\infty} P_T^n(B) \tag{46}$$

is a probability measure and satisfies for all $B \in \Sigma_{|X}$:

$$P_T(B) = \frac{1}{R(X)} R(B) \tag{47}$$

Proof Using Lemma 2, the terminating state measure P_T satisfies for any bounded measurable function $f : X \to \mathbb{R}$:

$$\int_X f(x) P_T(dx) = \int_X f(x) P_F(x, \perp) \sum_{n=0}^{\infty} P_F^n(s_0, dx).$$

It follows from Prop. 4 that

$$\mu(\{s_0\}) \int_X f(x) P_T(dx) = \int_X f(x) P_F(x, \perp) \mu(dx).$$

Following Lemma 7, and the positivity assumption on R that:

$$\int_X f(x) P_T(dx) = \frac{1}{R(X)} \int_X f(x) P_F(x, \perp) \mu(dx).$$

Finally, using (10), we obtain:

$$\int_X f(x) P_T(dx) = \frac{1}{R(X)} \int_X f(x) R(dx).$$

P_T being a probability measure follows by applying the last equality to the function $f : x \mapsto 1$. ∎

Next, we will prove Prop. 1.

Proposition 1. *If (μ, P_F, P_B) satisfy the detailed balance conditions in Def. 5, then $F = (\mu, P_F)$ satisfies the flow-matching conditions in Def. 3 and is thus a flow.*

Proof For any bounded measurable function $f : \bar{S} \to \mathbb{R}$ satisfying $f(s_0) = 0$, we can define a function $g : S \times \bar{S} \to \mathbb{R}$ such that for all $(s, s') \in S \times \bar{S}$, $g(s, s') = f(s')$. Note that g satisfies $g(s, s_0) = 0$ for every $s \in S$. Applying the detailed balance conditions to the function g, we have

$$\iint_{S \times \bar{S}} g(s, s') \mu(ds) P_F(s, ds') = \iint_{S \times \bar{S}} f(s') \mu(ds) P_F(s, ds')$$

26

On the other hand, using the RHS of (12) in the detailed balance conditions, we get

$$\iint_{S \times \bar{S}} g(s, s') \mu(ds') P_B(s', ds) = \iint_{S \times \bar{S}} f(s') \mu(ds') P_B(s', ds) = \int_{\bar{S}} f(s') \mu(ds') \int_S P_B(s', ds),$$

Following (2) and the the absolute continuity conditions of P_B with respect to κ^b, we have:

$$\forall s' \in S, \ P_B(s', \{\perp\}) = 0,$$

from which it follows that:

$$\iint_{S \times \bar{S}} g(s, s') \mu(ds') P_B(s', ds) = \int_{\bar{S}} f(s') \mu(ds') \underbrace{\int_{\bar{S}} P_B(s', ds)}_{=1}$$

This shows that (μ, P_F) satisfy the flow matching conditions. $\blacksquare$

Next, we will prove Prop. 2, which we restate here:

Proposition 2. *If (Z, P_F, P_B) satisfy the trajectory balance condition (13) wrt. a measure R, then $F = (\mu, P_B)$, where μ is defined by*

(1) $\mu(\{\perp\}) = \mu(\{s_0\}) = Z$
(2) $\forall B \in \Sigma_{|S}: \mu(B) = \mu(\{s_0\}) \sum_{n=0}^{\infty} P_F^n(s_0, B)$

satisfies both the flow-matching conditions (9) and the reward matching conditions (10) wrt. R.

Proof First we show that μ satisfies the flow-matching condition. for any bounded measurable function $f : \bar{S} \to \mathbb{R}$ satisfying $f(s_0) = 0$, we have :

$$\iint_{S \times \bar{S}} f(s') \mu(ds) P_F(s, ds') = \iint_{S \times \bar{S}} f(s') \mu(s_0) \sum_{n=0}^{\infty} P_F^n(s_0, ds) P_F(s, ds')$$

$$= \int_{\bar{S}} f(s') \mu(s_0) \sum_{n=0}^{\infty} P_F^{n+1}(s_0, ds')$$

$$= \int_{\bar{S}} f(s') \mu(s_0) \sum_{n=0}^{\infty} P_F^n(s_0, ds') \quad \text{(because } f(s_0) = 0\text{)}$$

$$= \int_{\bar{S}} f(s') \mu(ds')$$

Now, we will show the reward matching condition. For any bounded measurable function $f : X \to \mathbb{R}$

$$\int_X f(s) \mu(ds) P_F(s, \perp) = \int_S \mathbb{1}_X(s) f(s) \underbrace{\mu(\{s_0\})}_{=Z} \sum_{n=0}^{\infty} P_F^n(s_0, ds) P_F(s, \perp)$$

$$= \sum_{n=0}^{\infty} \int_S \mathbb{1}_X(s) f(s) R(ds) P_B^n(s, \{s_0\})$$

$$= \int_S \mathbb{1}_X(s) f(s) R(ds) \sum_{n=0}^{\infty} P_B^n(s, \{s_0\}) = \int_X f(s) R(ds)$$

where we used Prop. 5 and Lemma 8. $\blacksquare$

The following is the proof of Thm. 2, which we restate here:

Theorem 2. *(1) If $L_{FM}(-;\theta) = 0$ ν-almost surely, then $F = (\mu, P_F)$ is a flow (i.e. satisfies the flow-matching conditions in Def. 3).*
(2) If $L_{DB}(-;\theta) = 0$ $\nu \otimes \kappa$-almost surely, then (μ, P_F, P_B) satisfy the detailed balance conditions in Def. 5.
(3) If $L_{RM}(-;\theta) = 0$ $\nu_{|X}$-almost surely, then (μ, P_F) satisfies the reward matching conditions in (10).
(4) If $L_{TB}^n(-;\theta) = 0$ $((\nu \otimes \kappa^{\otimes n+1})_{|\{s_0\} \times \mathcal{S}^n \times \{\perp\}})$-almost surely for every $n \geq 0$, then $(Z\nu(\{s_0\}), P_F, P_B)$ satisfy the trajectory balance condition in Def. 6.

Proof We will first show the result for the flow-matching loss. Let the function $v : \bar{\mathcal{S}} \to \mathbb{R}_+$ be the function, depending on the parameter θ, defined by $\forall s' \in \bar{\mathcal{S}}$:

$$v(s';\theta) := \int_{\mathcal{S}} u(s;\theta) p_F(s,s';\theta) \kappa^b(s',ds).$$

If we assume that $L_{FM}(-;\theta) = 0$ ν-almost surely, then we have equivalently $u(-;\theta) = v(-;\theta)$ ν-almost surely. Let $f : \bar{\mathcal{S}} \to \mathbb{R}$ be a bounded measurable function such that $f(s_0) = 0$, we then have

$$\int_{\bar{\mathcal{S}}} f(s') u(s';\theta) \nu(ds') = \int_{\bar{\mathcal{S}}} f(s') v(s';\theta) \nu(ds')$$

$$= \int_{\bar{\mathcal{S}}} f(s') \int_{\mathcal{S}} u(s;\theta) p_F(s,s';\theta) \kappa^b(s',ds) \nu(ds')$$

$$= \iint_{\bar{\mathcal{S}} \times \mathcal{S}} f(s') u(s;\theta) p_F(s,s';\theta) \kappa(s,ds') \nu(ds),$$

where we used the fact that $\nu \otimes \kappa = \nu \otimes \kappa^b$ (from (4) in Def. 1) in the last equality. Replacing the densities (and reference measures) in the equality above with their corresponding measures μ and P_F, we get

$$\int_{\bar{\mathcal{S}}} f(s') \mu(ds') = \iint_{\mathcal{S} \times \bar{\mathcal{S}}} f(s') \mu(ds) P_F(s,ds').$$

Since this equality is valid for any bounded measurable function f satisfying $f(s_0) = 0$, this is the definition of $F = (\mu, P_F)$ satisfying the flow-matching conditions (Def. 3).

The proof for the detailed balance loss is similar. Let the functions $g : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to \mathbb{R}_+$ and $h : \bar{\mathcal{S}} \times \bar{\mathcal{S}} \to \mathbb{R}_+$ defined as

$$g(s,s';\theta) := u(s;\theta) p_F(s,s';\theta)$$
$$h(s,s';\theta) := u(s';\theta) p_B(s',s;\theta).$$

If $L_{DB}(-;\theta) = 0$ $\nu \otimes \kappa$-almost surely, then we have equivalently $g(-;\theta) = h(-,\theta)$. Let $f : \mathcal{S} \times \bar{\mathcal{S}} \to \mathbb{R}$ be a bounded measurable function such that $f(s,s_0) = 0$ for all $s \in \mathcal{S}$. We have

$$\iint_{\mathcal{S} \times \bar{\mathcal{S}}} f(s,s') g(s,s';\theta) (\nu \otimes \kappa)(ds,ds')$$

$$= \iint_{\mathcal{S} \times \bar{\mathcal{S}}} f(s,s') u(s;\theta) p_F(s,s';\theta) (\nu \otimes \kappa)(ds\, ds')$$

$$= \iint_{\mathcal{S} \times \bar{\mathcal{S}}} f(s,s') h(s,s';\theta) (\nu \otimes \kappa)(ds\, ds')$$

$$= \iint_{\mathcal{S} \times \bar{\mathcal{S}}} f(s,s') h(s,s';\theta) (\nu \otimes \kappa^b)(ds'\, ds)$$

$$= \iint_{\mathcal{S} \times \bar{\mathcal{S}}} f(s,s') u(s';\theta) p_B(s',s;\theta) (\nu \otimes \kappa^b)(ds'\, ds),$$

where we used $\nu \otimes \kappa = \nu \otimes \kappa^b$ in the 3rd inequality. Note that while the equalities between functions are valid $\nu \otimes \kappa$-almost surely over the whole space $\bar{\mathcal{S}} \times \bar{\mathcal{S}}$, we only used the equality restricted to $\mathcal{S} \times \bar{\mathcal{S}}$. Moreover, since u, p_F, and p_B are the densities of the respective measures μ, P_F, and P_B (wrt. the appropriate reference measures), we know that for $B \in \bar{\Sigma} \otimes \bar{\Sigma}$

$$(\mu \otimes P_F)(B) = \iint_B u(s;\theta) p_F(s,ds')(\nu \otimes \kappa)(ds\, ds')$$

$$(\mu \otimes P_B)(B) = \iint_B u(s';\theta) p_B(s',ds)(\nu \otimes \kappa^b)(ds'\, ds).$$

Replacing these measures in the equality above, we obtain

$$\iint_{S \times \bar{S}} f(s, s') \mu(ds) P_F(s, ds') = \iint_{S \times \bar{S}} f(s, s') \mu(ds') P_B(s', ds).$$

Since this equality is valid for any bounded measurable function f such that $f(s, s_0) = 0$ for all $s \in \mathcal{S}$, this corresponds to (μ, P_F, P_B) satisfying the detailed balance conditions (Def. 5).

Now, for the trajectory balance loss: for a trajectory $(s, s_1, .., s_{n+1}) \in \mathcal{S}^{n+2}$, we define:

$$p_F^{\otimes n+1}(s, \overrightarrow{s_{1:n+1}}) = p_F(s, s_1, \theta) \prod_{t=1}^{n} p_F(s_t, s_{t+1}, \theta)$$

$$p_B^{\otimes n}(\overrightarrow{s_{n:1}}, s) = p_B(s_1, s, \theta) \prod_{t=1}^{n-1} p_B(s_{t+1}, s_t, \theta)$$

For any bounded measurable function $f : \bar{\mathcal{S}}^{n+2} \to \mathbb{R}$, assuming $L_{TB}^n = 0$ almost surely for every $n \geq 0$:

$$\int_{\mathcal{S}^{n+2}} Z(\theta) f(s, \overrightarrow{s_{1:n+1}}) \mathbb{1}_{s=s_0, s_n \neq \perp, s_{n+1}=\perp} p_F^{\otimes n+1}(s, \overrightarrow{s_{1:n+1}}) \nu \otimes \kappa^{\otimes n+1}(ds \, \overrightarrow{ds_{1:n+1}})$$

$$= \int_{\{s_0\} \times \mathcal{S}^{n-1} \times \mathcal{X} \times \{\perp\}} Z(\theta) f(s, \overrightarrow{s_{1:n+1}}) p_F^{\otimes n+1}(s, \overrightarrow{s_{1:n+1}}) \nu \otimes \kappa^{\otimes n+1}(ds \, \overrightarrow{ds_{1:n+1}})$$

$$= \int_{\{s_0\} \times \mathcal{S}^n \times \{\perp\}} f(s, \overrightarrow{s_{1:n+1}}) r(s_n) p_B^{\otimes n}(\overrightarrow{s_{n:1}}, s) \nu \otimes \kappa^{\otimes n+1}(ds \, \overrightarrow{ds_{1:n+1}})$$

$$= \int_{\{s_0\} \times \mathcal{S}^n} f(s, \overrightarrow{s_{1:n}}, \perp) r(s_n) \underbrace{\kappa(s_n, \{\perp\})}_{=1 \text{ (see (6))}} p_B^{\otimes n}(\overrightarrow{s_{n:1}}, s) \nu \otimes \kappa^{\otimes n}(ds \, \overrightarrow{ds_{1:n}})$$

$$= \int_{\{s_0\} \times \mathcal{S}^n} f(s, \overrightarrow{s_{1:n}}, \perp) r(s_n) p_B^{\otimes n}(\overrightarrow{s_{n:1}}, s) \nu \otimes \kappa^{b, \otimes n}(\overrightarrow{ds_{n:1}} \, ds)$$

$$= \int_{\mathcal{S}^{n+1}} f(s, \overrightarrow{s_{1:n}}, \perp) \mathbb{1}_{s=s_0} r(s_n) \nu(ds_n) p_B^{\otimes n}(\overrightarrow{s_{n:1}}, s) \kappa^{b, \otimes n}(s_n, \overrightarrow{ds_{n-1:1}} \, ds)$$

Replacing the measures in the last equality obtained, we recover the TB condition in Def. 6 with $(Z\nu(\{s_0\}), P_F, P_B)$.

Here, we used $\nu \otimes \kappa^{\otimes n} = \nu \otimes \kappa^{b, \otimes n}, \forall n \in \{0, ..N\}$. We can show this by simple induction : for $n = 0$, it is trivially satisfied . Now suppose it is true for a given $n \leq N - 1$, using (4). We have :

$$\nu \otimes \kappa^{\otimes n+1}(ds \, \overrightarrow{ds_{1:n+1}}) = \nu \otimes \kappa^{\otimes n}(ds \, \overrightarrow{ds_{1:n}}) \kappa(s_n, ds_{n+1}) = \nu \otimes \kappa^{b, \otimes n}(\overrightarrow{ds_{n:1}} \, ds) \kappa(s_n, ds_{n+1})$$

$$= \nu(ds_n) \kappa(s_n, ds_{n+1}) \kappa^{b, \otimes n}(s_n, \overrightarrow{ds_{n:1}} \, ds) = \nu(ds_{n+1}) \kappa^b(s_{n+1}, ds_n) \kappa^{b, \otimes n}(s_n, \overrightarrow{ds_{n:1}} \, ds)$$

$$= \nu(ds_{n+1}) \kappa^{b, \otimes n+1}(s_{n+1}, \overrightarrow{ds_{n:1}} \, ds) = \nu \otimes \kappa^{b, \otimes n+1}(\overrightarrow{ds_{n+1:1}} \, ds)$$

Which proves the claim above.

$\blacksquare$

F. Proofs of lemmas and propositions in §A

Lemma 1.

Proof We prove the lemma by induction on n. The base case ($n = 1$) is trivially satisfied. Assuming the property holds for some $n \geq 0$, let $s \in \bar{\mathcal{S}}$ and $B \in \Sigma$ such that $\kappa^{n+1}(s, B) = 0$.

$$P_F^{n+1}(s, B) = \int_{\bar{\mathcal{S}}} P_F^n(s, ds') P_F(s', B).$$

If $P_F^{n+1}(s, B) > 0$, that would mean there exists an open set $B' \in \mathcal{T}$ such that $P_F^n(s, B') > 0$ and $P_F(s', B) > 0$ for all $s' \in B'$. From the induction hypothesis, it would follow that $\kappa^n(s, B') > 0$ and $\kappa(s', B) > 0$ for all $s' \in B'$, meaning that:

$$\kappa^{n+1}(s, B) = \int_{\bar{S}} \kappa^n(s, ds')\kappa(s', B) \geq \int_{B'} \kappa^n(s, ds')\kappa(s', B) > 0.$$

A contradiction ! Hence, $P_F^{n+1}(s, B) = 0$ ∎

Lemma 2.

Proof Starting from the definition of $P_\top^n$:

$$\int_{\mathcal{X}} f(x)P_\top^n(dx) = \int_{\bar{S}} f(s_n)\mathbb{1}_{\mathcal{X}}(s_n)P_\top^n(dx) = \int_{\bar{S}^{n+1}} \mathbb{1}_{\mathcal{X}}(s_n)f(s_n)P_F^{\otimes n}(s_0, ds_1 \ldots ds_{n+1})\mathbb{1}_{s_{n+1}=\perp}$$

Hence, using the recursive definition of $P_F^{\otimes n}$ in (22):

$$\int_{\mathcal{X}} f(x)P_\top^n(dx) = \int_{\bar{S}^{n+1}} \mathbb{1}_{\mathcal{X}}(s_n)f(s_n)P_F^{\otimes n-1}(s_0, ds_1 \ldots ds_n)P_F(s_n, ds_{n+1})\mathbb{1}_{s_{n+1}=\perp}$$

$$= \int_{\bar{S}^n} \underbrace{\mathbb{1}_{\mathcal{X}}(s_n)f(s_n)P_F(s_n, \{\perp\})}_{g(s_n)} P_F^{\otimes n-1}(s_0, ds_1 \ldots ds_n)$$

$$= \int_{\bar{S}^n} g(s_n)P_F^{\otimes n-1}(s_0, ds_1 \ldots ds_n)$$

$$= \int_{\bar{S}} g(s_n)P_F^{n-1}(s_0, ds_n) = \int_{\mathcal{X}} f(x)P_F(x, \{\perp\}P_F^{n-1}(s_0, ds_n),$$

where we applied Lemma 6 to the bounded and measurable function g. ∎

Lemma 3.

Proof Let $f : \bar{S}^2 \to \mathbb{R}$ be a bounded measurable function.

$$\int_{\bar{S}^2} f(s, s')\nu\kappa(ds')\kappa'(s', ds) = \int_{\bar{S}} f(s, s_0) \underbrace{\nu\kappa(\{s_0\})}_{=0} \kappa'(s_0, ds) + \int_{S} \int_{\bar{S}\backslash\{s_0\}} f(s, s')\nu\kappa(ds')\kappa'(s', ds)$$

$$+ \int_{S\backslash\{s_0\}} f(\perp, s')\nu\kappa(ds') \underbrace{\kappa'(s', \{\perp\})}_{=0} + f(\perp, \perp)\nu\kappa(\{\perp\})\kappa'(\perp, \{\perp\})$$

$$= \int_{S} \int_{\bar{S}\backslash\{s_0\}} f(s, s')\nu\kappa(ds')\kappa^b(s', ds) + \int_{\bar{S}} f(s, s_0) \underbrace{\nu\kappa(\{s_0\})}_{=0} \kappa^b(s_0, ds)$$

$$+ f(\perp, \perp)\nu\kappa(\{\perp\})\kappa^b(\perp, \{\perp\}) \tag{48}$$

On the other hand, let B be the largest open set within $\mathcal{S}$ such that $\forall s' \in B$, $\kappa^b(s', \{\perp\}) > 0$. Applying the definition of the reverse kernel (26) to the function $f : (s, s') \mapsto \mathbb{1}_{s=\perp}\mathbb{1}_B(s')$, we get:

$$\int_{\bar{S}} \mathbb{1}_B(s')\nu(\{\perp\})\kappa(\perp, s') = \int_{\bar{S}} \mathbb{1}_B(s')\nu\kappa(s')\kappa^b(s', \{\perp\})$$

The LHS of the previous equality is 0, following (2). It follows from the assumption that $\forall s' \in B$, $\kappa^b(s', \{\perp\}) > 0$ that

$\nu\kappa(B) = 0$. Hence:

$$\int_{\mathcal{S}\setminus\{s_0\}} f(\perp, s')\nu\kappa(ds')\kappa^b(s', \{\perp\})$$

$$= \int_{\bar{\mathcal{S}}\setminus\{s_0\}} \mathbb{1}_{\mathcal{S}\setminus B}(s')f(\perp, s')\nu\kappa(ds')\kappa^b(s', \{\perp\})$$

$$+ \int_{\bar{\mathcal{S}}\setminus\{s_0\}} \mathbb{1}_B(s')f(\perp, s')\nu\kappa(ds')\kappa^b(s', \{\perp\})$$

The first summand of the RHS of the last equality is zero by the definition of B. The second summand is zero because $\nu\kappa(B) = 0$. Going back to (48), we obtain:

$$\int_{\bar{\mathcal{S}}^2} f(s, s')\nu\kappa(ds')\kappa'(s', ds) = \int_{\bar{\mathcal{S}}^2} f(s, s')\nu\kappa(ds')\kappa^b(s', ds) = \int_{\bar{\mathcal{S}}^2} f(s, s')\kappa(ds)\kappa(s, ds')$$

∎

Proposition 3.

Proof First, note that (2) and (7) imply that $\forall n \geq N$, $\kappa^n(s_0, -) = \delta_\perp$. This can be shown by a simple induction on n, writing for any $B \in \Sigma$:

$$\kappa^{n+1}(s_0, B) = \int_{\bar{\mathcal{S}}} \kappa^n(s_0, ds)\kappa(ds, B).$$

This entails that (1) could be rewritten as:

$$\forall B \in \mathcal{T} \setminus \{\emptyset\}, \ \exists n \in \{0, \dots, N\} \ : \ \kappa^n(s_0, B) > 0.$$

Hence,

$$\forall B \in \mathcal{T} \setminus \{\emptyset\}, \ \nu(B) > 0.$$

The measure ν is thus strictly positive.

Note that for any B in $\Sigma_{|\mathcal{S}}$ such that $s_0 \notin B$.

$$\nu\kappa(B) = \int_{\bar{\mathcal{S}}} \nu(ds)\kappa(s, B)$$

$$= \int_{\bar{\mathcal{S}}} \sum_{n=0}^{N} \kappa^n(s_0, ds)\kappa(s, B) = \sum_{n=0}^{N} \int_{\bar{\mathcal{S}}} \kappa^n(s_0, ds)\kappa(s, B)$$

$$= \sum_{n=0}^{N} \kappa^{n+1}(s_0, B) = \sum_{n=1}^{N+1} \kappa^n(s_0, B).$$

Because $B \subseteq \mathcal{S}$, then $\kappa^{N+1}(s_0, B) = \delta_\perp(B) = 0$. And because $s_0 \notin B$, $\kappa^0(s_0, B) = \delta_{s_0}(B) = 0$. From this it follows that:

$$\nu\kappa(B) = \nu(B)$$

Then, let $B \in \Sigma_{|\mathcal{S}} \otimes \Sigma_{|\mathcal{S}}$ such that $(s_0, s_0) \notin B$. Using the definition of the reverse kernel, we obtain:

$$
\begin{aligned}
\nu \otimes \kappa(B) = (\nu\kappa) \otimes \kappa_\nu^r(B) &= \int_{\bar{\mathcal{S}} \times \bar{\mathcal{S}}} \mathbb{1}_B(s, s') \nu\kappa(ds') \kappa_\nu^r(ds', ds) \\
&= \int_{\mathcal{S} \times (\mathcal{S} \setminus \{s_0\})} \mathbb{1}_B(s, s') \underbrace{\nu\kappa(ds')}_{=\nu(ds)} \kappa_\nu^r(ds', ds) + \int_{\mathcal{S}} \mathbb{1}_B(s, s_0) \underbrace{\nu\kappa(s_0)}_{=0} \kappa_\nu^r(s_0, ds) \\
&= \int_{\mathcal{S} \times (\mathcal{S} \setminus \{s_0\})} \mathbb{1}_B(s, s') \nu(ds') \kappa_\nu^r(ds', ds) \\
&= \int_{\mathcal{S} \times (\mathcal{S} \setminus \{s_0\})} \mathbb{1}_B(s, s') \nu(ds') \kappa_\nu^r(ds', ds) + \int_{\mathcal{S} \setminus \{s_0\}} \mathbb{1}_B(s, s_0) \kappa(s_0) \underbrace{\kappa_\nu^r(s_0, ds)}_{=0} \\
&= \int_{\bar{\mathcal{S}} \times \bar{\mathcal{S}}} \mathbb{1}_B(s, s') \nu(ds') \kappa_\nu^r(ds', ds) \\
&= \nu \otimes \kappa_\nu^r(B)
\end{aligned}
$$

Finally, if $B \in \Sigma_{|\mathcal{S}}$:

$$
\nu \otimes \kappa(B \times \{\perp\}) = \int_{\mathcal{S}} \mathbb{1}_B(ds) \nu(ds) \kappa(s, \{\perp\}) = \int_{\mathcal{S}} \mathbb{1}_B(ds) \nu(\{\perp\}) \kappa^b(\perp, ds) = \nu \otimes \kappa^b(B \times \{\perp\})
$$

$$
\nu \otimes \kappa(\{\perp\} \times B) = \nu(\{\perp\}) \underbrace{\kappa(\perp, B)}_{=0} = \int_{\mathcal{S}} \mathbb{1}_B(s') \nu(ds') \underbrace{\kappa^b(s', \{\perp\})}_{=0} = \nu \otimes \kappa^b(\{\perp\} \times B)
$$

∎