GFlowNets for Causal Discovery: an Overview

Cristian Dragos Manta
Mila
Université de Montréal
cristian-dragos.manta@mila.quebec

Edward J. Hu
Mila
Université de Montréal
edward.hu@mila.quebec

Yoshua Bengio
Mila
Université de Montréal
CIFAR Senior Fellow, IVADO
yoshua.bengio@mila.quebec

Abstract

Causal relationships underpin modern science and our ability to reason. Automatically discovering useful causal relationships can greatly accelerate scientific progress and facilitate the creation of machines that can reason like we do. Traditionally, the dominant approaches to causal discovery are statistical, such as the PC algorithm. A new area of research is integrating recent advancement in machine learning with causal discovery. We focus on a series of recent work that leverages new algorithms in deep learning for causal discovery – notably, generative flow networks (GFlowNets). We discuss the unique perspectives GFlowNets bring to causal discovery.

1 Introduction

Causal discovery from data is challenging because of the discrete and combinatorial nature of the space of DAGs, which grows super-exponentially with the number of nodes (Cundy et al., 2021). Crucially, it has been shown that structure learning from data is NP-complete (Chickering, 1996). In addition, even if we had an oracle that could search over the whole space of DAGs efficiently, the limited amount of observational data reduces the certainty about the "best" graph candidate. Worse yet, even if we had infinite observational data, in most cases there are multiple causal models that are compatible with it. We discuss these concept more rigorously in §2.

A number of methods tackle the problem of structure learning. However, most of them attempt to learn a single point estimate of the DAG that best fits the data (Cundy et al., 2021; Zheng et al., 2018). In this overview, we cover a novel probabilistic inference tool called generative flow networks (GFlowNets; Bengio et al., 2021a,b) and review its successful applications in causal discovery relative to the alternatives. It is a form of score-based method (Malinsky, 2022), but it learns a multimodal generative model estimating the Bayesian posterior distribution over DAGs rather than returning a single DAG. Table 1 shows a summary of some key differences between the various lines of work compared in this paper. Since GFlowNets are used to learn Bayesian posteriors, we found it useful to include other works based on variational inference, such as (Lorch et al., 2021), in §5.

ICML 2023 Workshop: Sampling and Optimization in Discrete Space (SODS 2023).

Table 1: Comparison of different causal discovery approaches covered in our overview. More detailed descriptions of each column are provided below.

Method	Learning Target	Posterior	Mechanisms	Temporal	Amortized
NOTEARS	MLE	struct.(D) mech.(D)	linear		
DiBS	SVGD	struct.(B) mech.(B)	arbitrary		
BCDNet	ELBO	struct.(B) mech.(B)	linear-Gaussian		✓
DAG-GFlowNet	DB	struct.(B)	linear-Gaussian and categorical		✓
VB-GFlowNet	DB	struct.(B) mech.(D)	arbitrary		✓
JSP-GFlowNet	SubTB	struct.(B) mech.(B)	arbitrary		✓
DynGFlowNet	DB	struct.(B) mech.(B)	arbitrary	✓	✓
CSIvA	MLE	struct.(B)	/		✓
AVICI	MLE	struct (B)	/		✓

Learning Target. The appeared learning targets are maximum-likelihood estimation (MLE), Stein variational gradient descent (SVGD; Liu & Wang, 2016), evidence lower bound for variational inference (ELBO; Blei et al., 2017), detailed balance (DB; Bengio et al., 2021b), and sub-trajectory balance (SubTB; Madan et al., 2022).

Posterior. The posterior over structure (struct.) and mechanisms (mech.) is either Bayesian (B) or Dirac (D).

Mechanisms. The class of mechanisms, including function and noise, used to parametrize the relationships between a child node and its parents in the SCM.

Temporal. Whether the framework supports cyclic causal relationships that can be unrolled through time.

Amortized. Whether the cost of learning the causal structure is transferred to the task of training a neural network, which can approximate the exact answer at a cheaper cost.

2 Why Bayesian Causal Discovery?

Causal discovery assumes there is a unique ground-truth causal model that describes the data-generating process (Pearl, 2009). The goal is then to learn the ground-truth causal model. In principle, if the ground-truth model is included in the search space and that it is distinct in some ways, i.e., identifiable (Pearl, 2009), causal discovery is reduced to an optimization problem with a unique global minimum. This point-estimate perspective is exemplified by Zheng et al. (2018).

In practice, the ground-truth model is often only identifiable up to the Markov equivalence class (MEC) given observational data alone (Lorch et al., 2021). This is in particular the case for the linear-Gaussian and multinomial model classes (Eberhardt, 2017), which poses a problem because the MEC might be exponentially large. As a result, identifiability became a property often discussed in new causal models. With certain interventional data or different parametric assumptions, such as linear non-Gaussian model classes, we *sometimes* obtain identifiability (Eberhardt, 2017); however, interventional data is not always easy to obtain (or in the quantity and diversity that would make a single graph dominate) and a substantial body of work focuses on either multinomial or linear-Gaussian parametrizations due to the computational challenges involved (Deleu et al., 2022; Cundy et al., 2021). More fatally, even if we have identifiability of a given class of causal models, it is not clear at all if the ground-truth model belongs to that class. Identifiability becomes less relevant here because the learned model cannot be the true one. In fact, identifiability makes the problem worse because we confidently converge to an incorrect single point estimate when the causal model class is misspecified.

The solution we favor is anchored in the notion of Bayesian uncertainty, which obviates the issue of identifiability. That is, instead of working with a point estimate of a causal model, we work with a distribution over causal models. This allows us to specify a prior over such models, e.g., one which exponentially favors simpler models, and derive a posterior over them as we observe more data. Identifiability becomes a special case where the posterior collapses to a Dirac delta, but the approach works even if it doesn't. This raises the question of how tractable it may be to learn rich approximations of the Bayesian posterior and how a generative policy for sampling from

such a posterior may be exploited in a computationally tractable way. It also raises the question of computational asymptotics: if we dedicate more computational resources to form the approximate posterior, will we converge to the true posterior? The answer is affirmative for GFlowNets, which are described below.

3 GFlowNets and Bayesian Posteriors over Directed Acyclic Graphs

A distribution over causal models can be difficult to represent. We use the common assumption which decomposes causal models into directed acyclic graphs (DAGs), which encode the conditional independencies of the causal relationships, and parametrized mechanisms, which encode the conditional probability distributions (CPDs) of these relationships. The DAG and the mechanisms together define a structural causal model or SCM (Pearl, 2009). A distribution over causal models then at least describes a distribution over DAGs, whose support is super-exponential in the number of nodes (Cundy et al., 2021)). As a result, prior work has relied on either factorized variational approximations (Lopez et al., 2022) or expensive Markov Chain Monte Carlo (MCMC) simulations (Eaton & Murphy, 2012; Jain et al., 2023) to sample from posterior distributions over causal models.

GFlowNets (Bengio et al., 2021a,b) are a novel framework for training amortized samplers of compositional objects, such as DAGs. They learn to construct such objects step-by-step and are trained to sample them according to an energy function using self-consistency objectives (Bengio et al., 2021b) similar to ones used in reinforcement learning (Sutton & Barto, 2018). The key insight is that one can turn the difficulty of approximating an intractable distribution, defined by an energy function, into that of optimizing a large neural network. This allows us to leverage the advancement of deep learning, i.e., the training of large neural networks, in the last decade (Bengio et al., 2021b; Hu et al., 2023).

3.1 GFlowNet Training Objectives

The key objective of GFlowNets is to sample objects x according to the probability distribution $p(x) = \frac{\exp[-\mathcal{E}(x)]}{\sum_{x'} \exp[-\mathcal{E}(x')]}$, where $\mathcal{E}(\cdot)$ is an energy function. We shall use τ to denote a trajectory, a sequence of states that results in a complete object x; we also use $\tau \ni x$ to denote all trajectories ending in object x. Thus, a GFlowNet $q(\cdot)$, defined as a distribution over trajectories, trained to completion should exhibit

$$\sum_{\tau \ni x} q(\tau) \propto \exp[-\mathcal{E}(x)] \tag{1}$$

It is often convenient to parametrize $q(\cdot)$ using a forward policy $p_F(s' \mid s)$ where s and s' are adjacent intermediary states. Thus we have

$$\sum_{\tau \ni x} \prod_{(s,s') \in \tau} p_F(s' \mid s) \propto \exp[-\mathcal{E}(x)] \tag{2}$$

where $(s, s') \in \tau$ iterates over all consecutive states in a trajectory.

The main result of GFlowNet is that Eq. 2 is satisfied when one of the several training objectives are minimized globally. We discuss two such objectives most relevant to our use case; they are the flow matching objective (Bengio et al., 2021a) and the detailed balance objective (Bengio et al., 2021b)). We refer interested readers to Malkin et al. (2022a) and Madan et al. (2022) for other objectives.

Flow-matching Objective For every adjacent state-pairs (s, s'), meaning we can add an edge to DAG s to obtain DAG s', we define a scalar flow $F(s \to s') = \sum_{\tau \ni (s,s')} F(\tau) = \sum_{\tau \ni (s,s')} Zq(\tau)$. Here, we parametrize with $F(\cdot)$ to avoid the intractable partition function $Z = \sum_{\tau} F(\tau)$. The flow-matching objective is motivated by the observation that the "flow" coming into a state s equals the amount leaving it for all states when a GFlowNet is trained to completion. This means

$$\forall s, \sum_{s' \in Child(s)} F(s \to s') = \sum_{s'' \in Par(s)} F(s'' \to s) \tag{3}$$

where $Child(s)$ and $Par(s)$ denote the set of children and parent nodes of s. Bengio et al. (2021a) turned this into a practical objective function

$$\mathcal{L}_{FM} = \sum_s \left[\log \frac{\sum_{s' \in Child(s)} F(s \to s')}{\sum_{s'' \in Par(s)} F(s'' \to s)} \right]^2 \tag{4}$$

Detailed-balance Objective The summation over the set of parents or children can be expensive in Eq. 4. In addition, sampling from the GFlowNet requires the evaluation of a policy $\pi(s''|s) = \frac{F(s \to s'')}{\sum_{s' \in Child(s)} F(s \to s')}$, which involves many calls to the flow function. It is sometimes more convenient to parametrize with the policy directly. Bengio et al. (2021b) used both a forward policy $p_F(s'|s)$ and a backward policy $p_B(s|s')$ in addition to a scalar flow $F(s)$ for every state s. They observed that, when trained to completion, the flow associated with every transition $s \to s'$ is the same from the perspective of both the forward and the backward policy analogous to the detailed-balance condition in Markov chains

$$\forall (s, s'), F(s) p_F(s'|s) = F(s') p_B(s|s') \tag{5}$$

This can in turn be turned into a practical objective function

$$\mathcal{L}_{DB} = \sum_{(s,s')} \left[\log \frac{F(s) p_F(s'|s)}{F(s') p_B(s|s')} \right]^2 \tag{6}$$

We will see in Section 4.1 how Eq. 6 can be modified to adapt to model a distribution over DAGs for causal discovery.

So far, we have focused on how GFlowNets are trained to sample DAGs. On-going work leveraging a recent generalization of GFlowNets to the continuous domain has attempted to train and sample parameters for mechanisms using the same GFlowNet. We will briefly present that work in §4.4.

3.2 Advantages of the GFlowNet Framework

GFlowNets can be trained to sample objects step-by-step with probability proportional to a given reward function. Alternatives approaches are max-entropy reinforcement learning (RL) (Haarnoja et al., 2018) and hierarchical variational inference (HVI) (Sønderby et al., 2016). As elaborated in Bengio et al. (2021a), policies trained with max-entropy RL do not sample according to the reward function, i.e., exponentiated negative energy, even when trained to completion, if there exists multiple trajectories leading to the same final object, e.g., in causal discovery, we might build the same DAG by placing edges in different orders. Extensive experimentation in Malkin et al. (2022b) compare GFlowNets with existing amortized variational inference and shows that GFlowNets find a better tradeoff between the mode-seeking behavior of reverse-KL and the zero-avoiding behavior of forward-KL; the simulations show that this allows GFlowNets to better capture multiple modes, e.g., different DAGs in the same MEC. In addition, GFlowNets are able to be trained off-policy without resorting to importance sampling. In practice, this leads to more stable gradients and better optimization. A non-machine-learning alternative is to run MCMC in the DAG space (Lorch et al., 2021). This unamortized approach is prohibitively expensive due to the well-known issue of mode mixing (Jain et al., 2023; Bengio et al., 2021a).

4 Applications of GFlowNets in Causal Discovery

We compare works that apply GFlowNets to causal discovery in various setups. Overall, the structure of the GFlowNet is very similar across different papers, with the reward function being the biggest difference in most cases, which highlights the framework's great flexibility.

4.1 Bayesian Structure Learning with Generative Flow Networks (DAG-GFlowNet)

Deleu et al. (2022) is the first to use GFlowNets for Bayesian causal discovery. In their formulation, they assume access to N samples of a *static* and *single* dataset $\mathcal{D}$ of d observed *causal* variables $X_1, ..., X_d$. Their goal is to train a sampler over Bayesian networks from the posterior distribution $p(G|\mathcal{D})$ over DAGs G. We note that this framework and most of the other GFlowNets works are

agnostic to the causality formalism, since the learned distribution over graphs captures the ambiguity from having multiple graphs in the same MEC, as discussed in §2. The estimated posterior converges to the true one as the capacity of the GFlowNet parametrization and the training time are increased, i.e., as the training loss is brought to its global minimum of 0. The posterior collapses to the true causal graph in the limit as $N \to \infty$ when presented with enough appropriate interventional data. The framework is thus very flexible and remains principled regardless of the availability of interventional data and N.

GFlowNets sequentially construct a solution (e.g. G) by stochastically modifying a state variable (a partial construction of the object to generate). These works define the state space as the set $\mathcal{S}$ of all partially constructed DAGs. The action space $\mathcal{A}$ is the set of new edges that can be introduced to a given state $s \in \mathcal{S}$ while not introducing any cycles. Let us see how to define a reward function $R(s)$ for all terminating states s such that we obtain a sampler of $p(G|\mathcal{D})$ when the GFlowNet is trained to completion (see §3.1), i.e., when the GFlowNet samples G with probability proportional to $R(G)$:

$$p(G|\mathcal{D}) = \frac{p(\mathcal{D}|G)p(G)}{p(\mathcal{D})} \propto p(\mathcal{D}|G)p(G). \tag{7}$$

Thus, it suffices to define $R(G) := p(G)p(\mathcal{D}|G)$ for any terminating G. The prior over graphs could simply be the uniform distribution if we don't want to incorporate any prior knowledge, but one can also define a preference for sparser graphs. One limiting difficulty *that is alleviated in the next papers* consists of computing the marginal likelihood $p(\mathcal{D}|G)$, which is in principle given by

$$p(\mathcal{D}|G) = \int_\theta p(\mathcal{D}|G,\theta)p(\theta|G)d\theta, \tag{8}$$

where θ has the parameters of the causal mechanisms. $p(\mathcal{D}|G,\theta)$ is the likelihood of the data under the fully specified SCM and $p(\theta|G)$ is the prior over parameters. To the best of our knowledge, it is only possible to efficiently compute this integral (Deleu et al., 2023) when we restrict the class of causal mechanisms to be linear-Gaussian (Geiger & Heckerman, 1994), discrete categorical with Dirichlet prior (Heckerman & Geiger, 2013), or when we use a Gaussian process parametrization of non-linear mechanisms (von Kügelgen et al., 2019). Deleu et al. (2022) is restricted to the first two classes of models, which is very limiting.

We now turn our attention to how the reward is incorporated in the GFlowNet loss. The flow-matching objective is challenging to parametrize since flows near the initial state tend to be much larger than the ones further away (Deleu et al., 2022). The alternative used is the detailed-balance objective (see §3.1). Since each state is a valid DAG (and thus a terminating state), a modification is made in order to incorporate the probability of stopping at each state. To this end, let s_t, s_{t+1} be two consecutive (DAG) states, and let s_f be the (universal) final state that is reached once the terminating action is selected. Then by definition,

$$P_F(s_f|s_t) = \frac{F(s_t \to s_f)}{F(s_t)} = \frac{R(s_t)}{F(s_t)}, \tag{9}$$

where the second equality assumes that the GFlowNet is trained to completion. This implies that $F(s_t) = \frac{R(s_t)}{P_F(s_f|s_t)}$. Then, substituting this into Eq. 6 yields the following modified loss, which can be approximated by an empirical average over randomly drawn transitions from an exploratory policy to avoid the large summation:

$$\mathcal{L}_{DB} = \sum_{(s,s')} \left[\log \frac{R(s_t)P_F(s_{t+1}|s_t)P_F(s_f|s_{t+1})}{R(s_{t+1})P_B(s_t|s_{t+1})P_F(s_f|s_t)} \right]^2. \tag{10}$$

One advantage of the GFlowNets framework is that the graphs sampled from the GFlowNet posterior are always valid DAGs, unlike DiBS (Lorch et al., 2021). Another advantage over DiBS is that, once trained, we can efficiently[1] sample from the posterior because of amortization. In contrast, DiBS requires picking a specific number of samples ahead of time and requires re-running the optimization algorithm to construct new samples.

[1]To sample G, GFlowNets only require forward passes through the pretrained policy network.

The main drawback of this work is that they only learn the structure, not the parameters, of Bayesian networks. As a result, they make strong parametric assumptions about the causal mechanisms to compute the marginal likelihoods efficiently, which makes the method less practical for real data with complicated ground-truth mechanisms. The next two lines of work address this issue.

4.2 Bayesian learning of Causal Structure and Mechanisms with GFlowNets and Variational Bayes (VB-GFlowNet)

Nishikawa-Toomey et al. (2022) model the joint posterior $p(G, \theta|\mathcal{D})$, as opposed to only the marginal distribution over graphs $p(G|\mathcal{D})$, which allows one to not only use weaker parametric assumptions than Deleu et al. (2022), but also to answer causal queries with such a trained model via the predictive posterior (Jain et al., 2023; Toth et al., 2022). To learn $P(G, \theta|\mathcal{D})$, they assume the following factorization:

$$P(G, \theta|\mathcal{D}) \approx q_\phi(G)q_\lambda(\theta|G) = q_\phi(G)\prod_{i=1}^{d} q_\lambda(\theta_i|G)$$

where θ_i has the parameters of the i-th conditional. We show here that the last equality only relies on the weak assumption of a factorized prior, i.e., $P(\theta|G) = \prod_i P(\theta_i|G)$:

$$\begin{aligned}
P(\theta|D, G) &= \frac{P(\theta, D|G)}{P(D|G)} && (11)\\
&= \frac{\prod_i P(\theta_i|G)P(D_i|D_1^{i-1}, \theta_i, G)}{\prod_i P(D_i|D_1^{i-1}, G)} && (12)\\
&= \prod_i \frac{P(\theta_i, D_i|D_1^{i-1}, G)}{P(D_i|D_1^{i-1}, G)} && (13)\\
&= \prod_i P(\theta_i|D_1^i, G) && (14)
\end{aligned}$$

where D_i is the part of the dataset containing the values for the i-th variable and $D_1^i = (D_1, \ldots, D_i)$, assuming a topological ordering. Nishikawa-Toomey et al. (2022) use a GFlowNet to model $q_\phi(G)$ in the same way as in Deleu et al. (2022)) and they use a variational approximation for $q_\lambda(\theta|G)$. In order to find the parameters ϕ^* and λ^* that lead to the best approximation of $P(G, \theta|\mathcal{D})$, they derive an ELBO such that $\log p(\mathcal{D}) \geq \text{ELBO}(\phi, \lambda)$:

$$\begin{aligned}
\text{ELBO}(\phi, \lambda) = \mathbb{E}_{G\sim q_\phi}[&\mathbb{E}_{\theta\sim q_\lambda}[\log p(\mathcal{D}|\theta, G)]\\
&- \text{KL}(q_\lambda(\theta|G)||P(\theta|G))]\\
&- \text{KL}(q_\phi(G)||P(G)) && (15)
\end{aligned}$$

which they maximize with respect to ϕ and λ using alternating coordinate ascent steps. In order to update $q_\phi(G)$, they prove that, given a fixed λ, the optimal ϕ^* satisfies

$$\begin{aligned}
\log q_{\phi^*}(G) = \text{const} &+ \mathbb{E}_{\theta\sim q_\lambda}[\log p(\mathcal{D}|\theta, G)]\\
&- \text{KL}(q_\lambda(\theta|G)||P(\theta|G)) + \log p(G) && (16)
\end{aligned}$$

up to a normalizing constant that is independent of G. This means that training the GFlowNet with $\log R(G)$ given by the RHS of Eq. 16 will make it converge to $q_{\phi^*}(G)$. The remainder of the algorithm can be left unchanged from Deleu et al. (2022), which demonstrates the versatility of GFlowNets. In order to find $q_{\lambda^*}(\theta|G)$, one can resort to gradient ascent with respect to λ of Eq. 15 with fixed $\phi = \phi^*$ found at the previous iteration of coordinate ascent. The cycle continues until a convergence criterion is met.

4.3 DynGFN: Bayesian Dynamic Causal Discovery using Generative Flow Networks

Scientific phenomena often involve time-varying behavior and feedback loops. Does current cause voltage or vice-versa? Both, when we unroll the dynamics. So when we have only one time step measurement, it is desirable to extend Bayesian causal discovery to cyclic graphs, which is done for the first time in Atanackovic et al. (2023). Previous works either learn a Bayesian posterior over causal models given a static dataset or return a single point estimate from dynamic data (Atanackovic et al.,

2023). One additional challenge is the joint learning of the structure and nature of the relationships, akin to the parameters of mechanisms in §4.2, in this dynamic context.

In order to formalize the notion of cyclic SCMs, Atanackovic et al. (2023) posits that the data generation process is a time-varying function

$$x(t) = (x_1(t), x_2(t), ..., x_d(t)) : [0, T] \to \mathbb{R}^d$$
$$\text{s.t. } \frac{dx_i(t)}{dt} = f_i(x, \epsilon_{t,i}) \; \forall 1 \leq i \leq d$$

with independent random noises $\epsilon_{t,i}$. In the graphical representation of this stochastic dynamical system, $j \in \mathrm{Pa}(x_i)$ if and only if $\frac{\partial f_i}{\partial x_j} \neq 0$. This is an elegant way to represent instantaneous causal relationships which naturally allows cycles when $\frac{\partial f_i}{\partial x_i} \neq 0$ for some i. The functions f_i are analogous to the causal mechanisms from the usual SCM formalism. We will use the notation f_θ to indicate the aggregation of f_i across i that is parametrized by θ. With dx a shorthand notation for $\frac{dx}{dt}$, the observations are n pairs $\mathcal{D} = \left\{(x^{(k)}, dx^{(k)})\right\}_{k=1}^{n}$[2] taken at different time steps, and the ultimate goal is to learn $f_\theta(x, \epsilon)$.

In a similar spirit as Nishikawa-Toomey et al. (2022), they first use a GFlowNet to sample $G \sim Q(G|\mathcal{D})$. One major problem is the even bigger search space that is comprised of 2^{d^2} structures, since there is no DAG constraint anymore. To solve this issue, they assume a per-node factorization:

$$Q(G|\mathcal{D}) = \prod_{i=1}^{d} Q_i(G[i, .]|\mathcal{D}),$$

which can be derived from assuming a factorization of the prior (similarly to 14). The above is parameterized by an independent policy head for each node. Another assumption that reduces the search space is the sparsity pattern in the Jacobian of f, i.e., that $\frac{df_i}{dx_j} \neq 0$ only for a small set of variables, which is enforced via L_0 regularization of the adjacency matrix of G.

Once the GFlowNet reaches state s_i corresponding to a partially constructed graph, the next step is to sample the parameters θ of the instantaneous dynamics. For this, they use a neural network $\theta = h_\phi(G)$. Unfortunately, this point estimate does not capture any uncertainty, unlike Nishikawa-Toomey et al. (2022)), but the authors suggest that this is in principle possible by putting a prior over ϕ. Given a pair (s_i, θ), they compute $\widehat{dx} := f_\theta(x, s_i)$ and they define the reward that contains the L_0 sparsity penalty as

$$R(s_i) := e^{-(||dx - \widehat{dx}||_2^2 + \lambda_0 ||s_i||_0)}. \tag{17}$$

Then, they update the parameters of the GFlowNet using the same detailed-balance condition as in Eq. 10 for each transition sampled from a tempered version of the training policy. Once the GFlowNet policy decides to stop, they update the parameters ϕ for predicting θ with $\phi \leftarrow \phi - \epsilon \nabla_\phi \log R(s_i)$.

4.4 Joint Bayesian Inference of Graphical Structure and Parameters with a Single Generative Flow Network (JSP-GFlowNet)

The main problem that Nishikawa-Toomey et al. (2022) solves is the ability to jointly learn a posterior distribution $P(G, \theta|\mathcal{D})$ over causal graphs and mechanism parameters. For this, they separate the learning procedure into a GFlowNet part, for the posterior over graphs, and a variational Bayes part for the mechanisms. One reason for this was that at the time, the theory and application of GFlowNets only allowed discrete state and action space, while mechanism parameters are continuous. Recently, a theory of continuous GFlowNets (Lahlou et al., 2023) is developed. This enables us to use a single GFlowNet to sample both discrete graphs and continuous parameters for the above setup of Bayesian causal discovery.

This is done in Deleu et al. (2023). They define complete states as pairs (G, θ). The GFlowNet first constructs G by starting with the empty graph and by adding one edge at the time in the same way as in Deleu et al. (2022). Then, a special "stop" action is selected to indicate that the construction of G, or the first phase, is terminated. This results in an incomplete state $(G, .)$. Once this is done, the

[2] In practice, they use a noisy measurement of $x^{(k)}$.

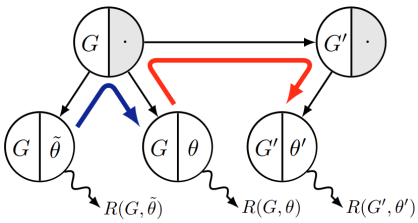

Figure 1: Illustration of undirected paths in the GFlowNet state space. The blue arrow shows and undirected path of length 2, while the red one shows an undirected path of length 3. This figure is borrowed from Deleu et al. (2023).

second phase consists of sampling the parameters θ conditioned on G, following the GFlowNet's forward transition probabilities[3]. We let $P_\phi(G'|G)$ denote the transition probability during the first phase and $P_\phi(\theta|G)$ denote the transition probability during the second phase, both parametrized by the GFlowNet parameters ϕ. By a similar argument as in Deleu et al. (2022), they define the reward

$$R(G, \theta) = P(G, \theta, \mathcal{D}) = P(G)P(\theta|G)P(\mathcal{D}|\theta, G) \tag{18}$$

for any complete state (G, θ). Note that in Eq. 18, both $P(G)$ and $P(\theta|G)$ are priors over graphs and causal mechanism parameters, respectively. The likelihood term $P(\mathcal{D}|\theta, G)$ is easier to compute than the marginal likelihood in Deleu et al. (2022). This makes this approach applicable to any kind of mechanisms parametrization in principle. Note that not all states are complete in this setup, which means that they cannot use the same trick with the detailed-balance loss as in Deleu et al. (2022). Instead, they use a variant of the trajectory balance loss, called the subtrajectory balance loss (Malkin et al., 2022a), that operates on undirected paths of length 3 of the form

$$(G, \theta) \leftarrow (G, .) \rightarrow (G', .) \rightarrow (G', \theta'), \tag{19}$$

as shown in Figure 1.

The resulting loss is

$$\sum_{G \rightarrow G'} \left[\log \frac{R(G', \theta')P_B(G|G')P_\phi(\theta|G)}{R(G, \theta)P_\phi(G'|G)P_\phi(\theta'|G')} \right]^2 \tag{20}$$

which can be computed using an exploratory policy over transitions $G \rightarrow G'$ as in Deleu et al. (2022).

5 Related Work

We briefly review other work on Bayesian causal discovery using variational inference. Using Zheng et al. (2018) as the foundation, Lorch et al. (2021) introduces a latent variable Z that is used to generate graphs from a probabilistic model $P(G|Z)$. It starts with some randomly initialized particles for Z and employs Stein variational gradient descent (SVGD; Liu & Wang, 2016) to model $P(Z|\mathcal{D})$, which is in turn used to model the Bayesian posterior over DAGs $P(G|\mathcal{D})$ or over full SCMs $P(G, \theta|\mathcal{D})$. One crucial weakness of this method is that once trained, it cannot sample new graphs since they must fix the number of particles in advance. On the other hand, GFlowNets amortize all the computational costs during training and can be used to generate an unlimited number of models from the posterior efficiently at test time. Another weakness of Lorch et al. (2021) is that they only enforce the acyclicity constraint in a soft way via a penalty term in the prior $P(Z)$, which makes them prone to output invalid DAGs a small fraction of the time. In contrast, GFlowNets use a mask at each transition to ensure that the DAGs are valid.

Cundy et al. (2021) learns a posterior over the full SCM. They assume a linear-Gaussian model class in order to parametrize a given DAG as a weighted adjacency matrix $W = PLP^T$, where L

[3]For example, we can define $P_\phi(\theta|G, \text{stop}) = \mathcal{N}(\theta|\boldsymbol{\mu}_\phi(G), \Sigma_\phi(G))$, where $\boldsymbol{\mu}_\phi$ and Σ_ϕ are outputs of a neural network that is part of the GFlowNet.

is a strictly lower triangular matrix and P is a permutation matrix. They then learn the posterior $P(P, L, \Sigma|\mathcal{D})$, where Σ is the noise covariance matrix as in §4.1. This particular decomposition is their solution to guarantee acyclicity (since L corresponds to a canonical ordering of the nodes and contains no cycles), something that Lorch et al. (2021) lacks, but entails the limiting parametric assumptions that this work suffers from. In contrast, as seen in §4, GFlowNets solve both of these problems, and more.

A more recent line of work (CSIvA; Ke et al., 2022) casts causal discovery as a supervised learning problem. They use a maximum-likelihood objective to learn a rich multimodal Bayesian posterior over causal structures, just like GFlowNet-based methods, but with the edges probabilities being parametrized by an autoregressive transformer over edges. These posterior distributions are learned on synthetic dataset-causal graph pairs in a way that generalizes to unseen, realistic datasets. A key difference is that the GFlowNet works should require less neural network capacity and training time since they learn to sample from the Bayesian posterior for the *single* actually given dataset rather than for any dataset. Another difference is that, although the CSIvA inference model is very expressive, they do not enforce acyclicity. In contrast, GFlowNets always produce DAGs by design. In concurrent work, Lorch et al. (AVICI; 2022) also learn a posterior distribution that is amortized across different datasets. However, they optimize a variational objective to find the distribution $q(G; \theta)$ that best approximates $p(G|\mathcal{D})$ among a family of distributions over graphs that factorize across edges, i.e.

$$q(G; \theta) = \prod_{i,j} q(g_{i,j}; \theta_{i,j}), \tag{21}$$

where $\theta_{i,j}$ is the parameter of the Bernoulli distribution that is used to sample $g_{i,j}$, which indicates the presence or the absence of an edge between nodes i and j. Although it is impossible to enforce acyclicity under the factorized distribution entailed by Eq. 21, Lorch et al. (2022) uses constrained optimization to enforce that, *in expectation*, the graphs sampled from the learned posterior are acyclic.

6 Conclusion

Causal discovery is a fundamental scientific problem with valuable downstream applications like drug discovery (Jain et al., 2022). We reviewed how GFlowNets (Bengio et al., 2021a,b), a novel deep learning framework, are used to model Bayesian posteriors over causal graphs and mechanisms (Deleu et al., 2022; Nishikawa-Toomey et al., 2022; Atanackovic et al., 2023). With the growing availability of data and computational resources, the Bayesian approach enabled by GFlowNets holds the promise of better robustness under uncertainty and misspecification, a principled incorporation of domain knowledge as prior, and more robust predictions (not confidently wrong) via Bayesian model averaging.

7 Acknowledgements

We would like to thank Dhanya Sridhar and Tristan Deleu for helpful discussions about the project and for their valuable feedback on this paper. This work was partially supported by the NSERC CGS M and by the FRQNT scholarships.

References

Atanackovic, L., Tong, A., Hartford, J., Lee, L. J., Wang, B., and Bengio, Y. Dyngfn: Bayesian dynamic causal discovery using generative flow networks. *arXiv preprint arXiv:2302.04178*, 2023.

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.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Chickering, D. M. Learning bayesian networks is np-complete. *Learning from data: Artificial intelligence and statistics V*, pp. 121–130, 1996.

Cundy, C., Grover, A., and Ermon, S. Bcd nets: Scalable variational approaches for bayesian causal discovery. *Advances in Neural Information Processing Systems*, 34:7095–7110, 2021.

Deleu, T., Góis, A., Emezue, C., Rankawat, M., Lacoste-Julien, S., Bauer, S., and Bengio, Y. Bayesian structure learning with generative flow networks. *arXiv preprint arXiv:2202.13903*, 2022.

Deleu, T., Nishikawa-Toomey, M., Subramanian, J., Malkin, N., Charlin, L., and Bengio, Y. Joint bayesian inference of graphical structure and parameters with a single generative flow network. *arXiv preprint arXiv:2305.19366*, 2023.

Eaton, D. and Murphy, K. Bayesian structure learning using dynamic programming and mcmc. *arXiv preprint arXiv:1206.5247*, 2012.

Eberhardt, F. Introduction to the foundations of causal discovery. *International Journal of Data Science and Analytics*, 3:81–91, 2017.

Geiger, D. and Heckerman, D. Learning gaussian networks. In *Uncertainty Proceedings 1994*, pp. 235–243. Elsevier, 1994.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *International Conference on Machine Learning (ICML)*, 2018.

Heckerman, D. and Geiger, D. Learning bayesian networks: a unification for discrete and gaussian domains. *arXiv preprint arXiv:1302.4957*, 2013.

Hu, E., Malkin, N., Jain, M., Everett, K., Graikos, A., and Bengio, Y. Gflownet-em for learning compositional latent variable models. *arXiv preprint arXiv:2302.06576*, 2023.

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.

Jain, M., Deleu, T., Hartford, J., Liu, C.-H., Hernandez-Garcia, A., and Bengio, Y. Gflownets for ai-driven scientific discovery. *arXiv preprint arXiv:2302.00615*, 2023.

Ke, N. R., Chiappa, S., Wang, J., Bornschein, J., Weber, T., Goyal, A., Botvinic, M., Mozer, M., and Rezende, D. J. Learning to induce causal structure. *arXiv preprint arXiv:2204.04875*, 2022.

Lahlou, S., Deleu, T., Lemos, P., Zhang, D., Volokhova, A., Hernández-García, A., Ezzine, L. N., Bengio, Y., and Malkin, N. A theory of continuous generative flow networks. *arXiv preprint arXiv:2301.12594*, 2023.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose Bayesian inference algorithm. *Neural Information Processing Systems (NIPS)*, 2016.

Lopez, R., Huetter, J.-C., Pritchard, J., and Regev, A. Large-scale differentiable causal discovery of factor graphs. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 19290–19303. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/7a8fa1382ea068f3f402b72081df16be-Paper-Conference.pdf`.

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

Lorch, L., Sussex, S., Rothfuss, J., Krause, A., and Schölkopf, B. Amortized inference for causal structure learning. *arXiv preprint arXiv:2205.12934*, 2022.

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.

Malinsky, D. Introduction to causal discovery, 2022. URL https://simons.berkeley.edu/workshops/causality-boot-camp. Causality Boot Camp.

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

Malkin, N., Lahlou, S., Deleu, T., Ji, X., Hu, E., Everett, K., Zhang, D., and Bengio, Y. GFlowNets and variational inference. *arXiv preprint 2210.00580*, 2022b.

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.

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. *Neural Information Processing Systems (NIPS)*, 2016.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT Press, 2018.

Toth, C., Lorch, L., Knoll, C., Krause, A., Pernkopf, F., Peharz, R., and Von Kügelgen, J. Active bayesian causal inference. *arXiv preprint arXiv:2206.02063*, 2022.

von Kügelgen, J., Rubenstein, P. K., Schölkopf, B., and Weller, A. Optimal experimental design via bayesian optimization: active causal structure learning for gaussian process networks. *arXiv preprint arXiv:1910.03962*, 2019.

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. Dags with no tears: Continuous optimization for structure learning. *Advances in neural information processing systems*, 31, 2018.