Digital Discovery

ROYAL SOCIETY OF CHEMISTRY

PERSPECTIVE

View Article Online
View Journal | View Issue

Check for updates

Cite this: *Digital Discovery*, 2023, **2**, 557

GFlowNets for AI-driven scientific discovery

Moksh Jain,*[ab] Tristan Deleu,[ab] Jason Hartford,[ab] Cheng-Hao Liu,[cb] Alex Hernandez-Garcia[ab] and Yoshua Bengio[abd]

Received 12th January 2023
Accepted 28th March 2023

DOI: 10.1039/d3dd00002h

rsc.li/digitaldiscovery

Tackling the most pressing problems for humanity, such as the climate crisis and the threat of global pandemics, requires accelerating the pace of scientific discovery. While science has traditionally relied on trial and error and even serendipity to a large extent, the last few decades have seen a surge of data-driven scientific discoveries. However, in order to truly leverage large-scale data sets and high-throughput experimental setups, machine learning methods will need to be further improved and better integrated in the scientific discovery pipeline. A key challenge for current machine learning methods in this context is the efficient exploration of very large search spaces, which requires techniques for estimating reducible (epistemic) uncertainty and generating sets of diverse and informative experiments to perform. This motivated a new probabilistic machine learning framework called GFlowNets, which can be applied in the modeling, hypotheses generation and experimental design stages of the experimental science loop. GFlowNets learn to sample from a distribution given indirectly by a reward function corresponding to an unnormalized probability, which enables sampling diverse, high-reward candidates. GFlowNets can also be used to form efficient and amortized Bayesian posterior estimators for causal models conditioned on the already acquired experimental data. Having such posterior models can then provide estimators of epistemic uncertainty and information gain that can drive an experimental design policy. Altogether, here we will argue that GFlowNets can become a valuable tool for AI-driven scientific discovery, especially in scenarios of very large candidate spaces where we have access to cheap but inaccurate measurements or too expensive but accurate measurements. This is a common setting in the context of drug and material discovery, which we use as examples throughout the paper.

1 Introduction

The climate crisis, antibiotic resistance and the prospect of new pandemics are some of the biggest threats to humanity, posing immense risks to global health and food security. One important common aspect to all these threats and others is that significant new scientific discoveries are required to mitigate them. According to the 2022 report by the Intergovernmental Panel on Climate Change (IPCC),[1] limiting global warming will require the adoption of alternative fuels, as well as improvements in the efficiency of energy production and material synthesis. The discovery of new materials, such as electrocatalysts that improve the energy efficiency of chemical reactions, can therefore play a crucial role in such a transition. Correspondingly, growing risks of antimicrobial resistance and pandemics make it essential to accelerate the pipeline for discovery of new drugs. Consequently, the well-being of our societies will strongly depend on the pace of our scientific discoveries.

Historically, scientific discovery has been the outcome of either serendipity—such as penicilin and Teflon[2]—or the rather slow experimental science loop: observations are accumulated from past experiments, which are carefully analyzed by experts who produce new hypotheses and design experiments that will eventually yield new, valuable observations to continue the cycle (see Fig. 1). While this model has well served the progress of science for centuries and will continue to do so in certain domains, the fully manual version of this cycle is too slow for the pressing emergencies of our time. A bottleneck in the cycle occurs when the analysis of data, production of hypotheses and experimental specification are manual. This is further exacerbated when the search space of candidates is dauntingly large, as is the case for drug discovery where there exist 10^{60} feasible small molecules, according to estimations.[3]

The scale at which scientific experiments can be conducted is rapidly increasing, enabled by advances in robotics, biotechnology and computational capabilities, among others.[4] For example, we can now easily and cheaply collect high-dimensional images and videos, electron microscopy data or the gene expression of millions of cells. Furthermore, we can also conduct thousands or even millions of experiments in parallel to screen new candidate molecules, experiment with a sequence of reactions, *etc.* If experimental interventions can

[a]*Université de Montréal, Canada. E-mail: moksh.jain@mila.quebec*
[b]*Mila Quebec AI Institute, Canada*
[c]*McGill University, Canada*
[d]*CIFAR Fellow & IVADO, Canada*

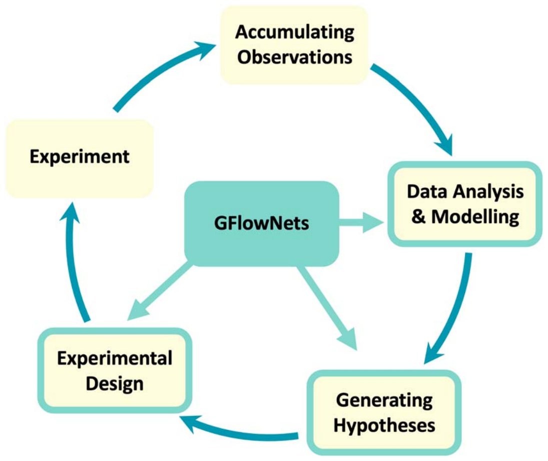


Fig. 1 The iterative experimental loop of scientific discovery: observations and data accumulated from past experiments are analyzed and used to generate new hypotheses, and in turn new experiments that will yield new data to continue to cycle. Highlighted with a blue frame are the steps for which we discuss how GFlowNets can be used.

be combined, we can sample from a combinatorially large space of possible experiments at each step. The avenues opened by such large-scale availability of data and compute have been identified as the fourth paradigm in scientific discovery.[5] Nonetheless, our current tools are not enough to truly utilize all the information and resources at our disposal.[6] In this context, the maturation of tailored machine learning (ML) methodologies offers the possibility to not only analyze and make sense of the data, but also to improve the generation of hypotheses and design of experiments, accelerating the experimental science loop.

ML techniques have been employed in all of the main steps of the experimental science loop illustrated in Fig. 1: (a) analyzing and modeling the data accumulated from experiments, (b) characterizing and generating hypotheses compatible with the data, (c) designing the next experiments, and (d) performing the experiments. Analysis and modeling of data is a naturally appealing scenario for ML methods, which are typically designed for extracting predictive patterns from large data sets. For instance, machine learning methods have been quite successful in modeling the quantum mechanical properties of small molecules.[7] ML approaches have also been studied in the context of designing candidate experiments.[8] A standard example of this is leveraging tools from reinforcement learning (RL) and Bayesian optimization (BO) for searching candidates that optimize (as a reward) some property of the candidate.[9,10] However, as we discuss in detail in Section 2, existing approaches often lack a principled treatment of the challenges introduced by limited data, uncertainty and underspecification of the objectives in scientific domains.

In this paper, we discuss how a novel machine learning (ML) framework, called Generative Flow Networks[11,12]—or GFlowNets for short—can help in addressing the shortcomings of existing ML approaches in scientific domains. GFlowNets are general purpose inference machines, which enable generating samples with a probability proportional to some reward function. As a consequence, GFlowNets have emerged as a potentially transformative tool for scientific discovery, as they can be used to generate hypotheses, design experiments and model the experimental observations, key steps of the experimental science loop (Fig. 1)—note that ML-augmented robotics can also be useful in the experimental step, but we do not discuss this here.

Throughout the paper, we consider the following motivating examples to make the discussion more concrete.

Example 1.1: An important part of fighting growing antimicrobial resistance and emergent infectious diseases is speeding up the discovery of novel small organic molecules (and peptides) that inhibit the action of target bacteria or one to several target proteins. The primary goal is to search the space of molecules (including peptides) for candidates that bind to the target of interest and inhibit or activate its function. The space of molecules is combinatorially large, and the accurate evaluation of the desired activity (*e.g.* binding affinity) *in vitro* or even *in silico* is expensive. Additionally, aside from binding to a target, there are several other pharmacological criteria which a molecule needs to satisfy for use as a therapeutic, such as low toxicity to humans, good Absorption, Distribution, Metabolism, and Excretion ("ADME"), and ease of synthesis.

Example 1.2: Discovery of novel materials for applications in the generation, storage, and use of clean energy that have better efficiency and rely on sustainable raw materials are critical in aiding efforts to reduce rising global temperatures. The goal here is discovering novel inorganic as well as organic materials, and there are often several specific metrics to optimize simultaneously. A representative example is the development of storage materials for lithium/sodium ions in lithium/sodium ion batteries, where the class of metal oxides (*e.g.* Fig. 2) alone already represents a combinatorially large search space. Inside a lithium/sodium ion battery, a good candidate metal oxide for energy storage should show high energy density, together with other metrics such as high capacity retention, low irreversible capacity, and low cost of synthesis.

Example 1.3: Modeling the genetic pathways through which diseases progress within humans plays a critical role in our understanding of human biology. These causal models can help us understand the behavior of various interventions such as therapeutics within the complex environment of the human body. The goal is to learn such causal models with the help of targeted interventions. Even the causal structure of a single cell is a major challenge and raises difficult questions to appropriately scale algorithms and combine learning from data with prior knowledge from biology.

1.1 Organization

In Section 3 we discuss existing ML methodology, highlighting the distinctive features of GFlowNets. Next in Section 4 we introduce ideas around amortized inference, and discuss how GFlowNets emerge from these ideas. In Section 4.2, we highlight scientific discovery problems where GFlowNets have been applied successfully. In Section 4.3, we discuss how GFlowNets can also be used to represent a distribution over causal models

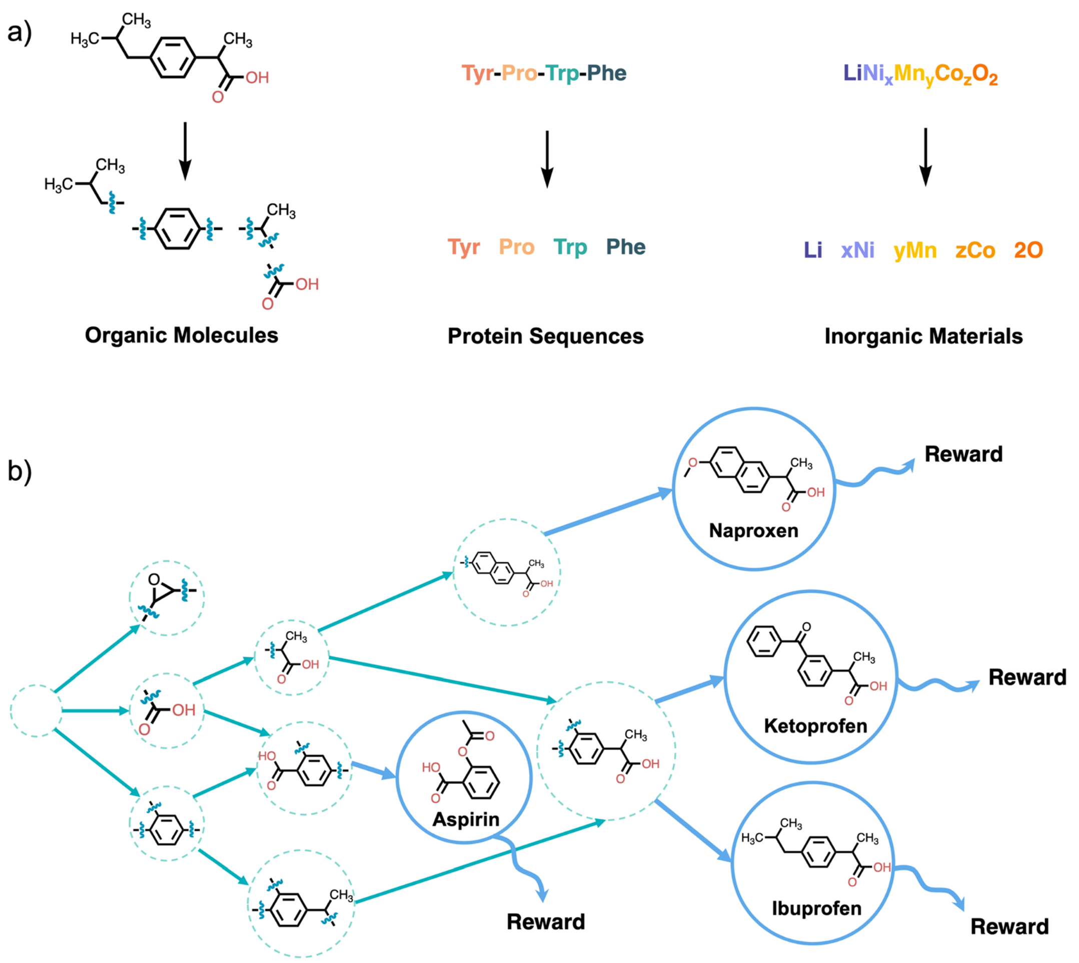


Fig. 2 Example schematics of a GFlowNet. (a) The objects whose distribution is modeled by a GFlowNet should be compositional, like graphs or sequences, built through a sequence of actions; (b) representation of the sequences of actions by which a GFlowNet can construct a molecule by composing smaller fragments, as a directed acyclic graph (DAG) whose nodes represent partially constructed molecules, and a reward is provided when the molecule is completed.

linking multiple random variables of interest and to estimate the posterior distributions of interest along with the marginalized quantities, such as Bayesian posterior densities, that are important to evaluate the information gain from an experiment. To conclude, in Section 5 we chart a potential path towards a unified framework for scientific discovery driven by GFlowNets.

2 Challenges for AI in scientific discovery

In the last few decades, ML has enabled remarkable technological advances ranging from agents that can surpass humans at the game of Go[13] to breakthrough advances in protein folding prediction.[14] These advances have been enabled, in part, by the availability of extremely large datasets and often of a well-specified objective to be optimized. In many scientific discovery applications, however, the limited available data, the uncertainty intrinsic to measurements and the under-specification of objectives pose serious challenges for leveraging ML approaches. In this section, we discuss the relevance of these challenges and argue how GFlowNets can circumvent them.

2.1 Limited data and uncertainty

One critical challenge in leveraging learning-based approaches for scientific discovery is the limited availability of data and the associated uncertainty. Part of the uncertainty is due to unreliable measurements, called aleatoric uncertainty, and part is due to having a limited amount of training data, called epistemic uncertainty, which is the uncertainty associated with theories or models and their parameters,[15] due to the finite size of datasets. Epistemic uncertainty emerges because multiple theories or models or settings of parameters can be compatible with the given data, and Bayesian posteriors on these can capture the epistemic uncertainty. By design, current state-of-the-art ML approaches rely on access to large data sets to extract useful patterns. But owing to experimental limitations, it can be extremely expensive or impossible to obtain large amounts of accurate and precise data at the scale required my modern ML approaches in many applications of interest. In drug discovery, for example, obtaining experimental binding affinities for ligands with a target protein, at the scale required for ML methods, is often very challenging. Furthermore, the differences in experimental techniques and measurement noise can lead to different binding affinities for the same ligand by orders of magnitude. The same is true in materials science: as an example, in the research for new battery materials, a few thousands of data points are already considered high-throughput,[16] and the measurement metrics such as energy density can vary significantly based on minute details of the experimental setup. Thus, it is essential for models to account for the aleatoric and epistemic uncertainty within the context of scientific discovery.

Additionally, scientific phenomena occurring in nature tend to be complex and often a product of complicated processes. Standard machine learning models that learn a function mapping an input to an output from data can fail to generalize to unseen scenarios where the phenomenon occurs. Incorporating the causal structure of the phenomenon can introduce effective inductive biases that can allow models to generalize to novel scenarios.[17] Whereas a given model will account for uncertainty in outcomes, *i.e.*, aleatoric uncertainty, by modeling Bayesian posteriors we can account for all the models that fit well the data, thus also capturing the epistemic uncertainty, which is what we want to reduce through experiments. As we discuss in Section 4.3, GFlowNets can be employed to model the posterior distribution of causal models that fit the data well.

2.2 Underspecification and diversity

ML approaches often assume access to some reward signal or scoring function to evaluate the quality and utility of experimental designs. For instance, to design drug-like molecules, the true objective is to find ligands that specifically inhibit the target protein within the human body. However, this objective can hardly be specified and conveniently quantified as a simple scalar reward. In practice, an estimate of the binding affinity of the molecule with the target protein is used instead as the reward signal to search for candidate molecules. This comes with two caveats: first, the estimation of binding affinity is not immune to systematic and random errors; second, the binding energy alone cannot account for many of the factors that can influence the effect of the ligand within the human body. For instance, a molecule (or a series of similar molecules) that only optimizes this binding energy may be potentially useless or even harmful *in vivo* because it could bind to many other proteins and thus be toxic (more broadly, the molecule could have poor ADME). These aspects make it critical to find diverse hypotheses—in this case, diverse motifs of molecules—to account for the underspecification and uncertainty in the reward signal. Nonetheless, popular approaches to ML-aided scientific discovery, like RL[9] and Bayesian optimization[10] aim to discover a single maximizer of the reward signal, not accounting for underspecification of the reward signal itself. Diversity of candidate solutions is also particularly relevant in the evolution of new generations of technologies. For example, before the use of perovskites (or other new materials) in solar cells, optimization of power conversion efficiency and manufacturing in silicon-based solar cells was yielding diminishing returns, and only the use of radically different materials was able to change this situation. By sampling proportional to the reward, GFlowNets, can mitigate the problem of missing out potentially interesting findings due to underspecification in the target reward, generating a diverse set of high-reward candidate solutions to the discovery problem at hand.

3 Background

In this section, we review the relevant fundamental areas in machine learning and set the stage for Section 4 where we discuss how GFlowNets can address the challenges presented in Section 2.

3.1 Preliminaries

To establish notation, consider an example problem from organic synthesis where chemists want to find new and/or efficient synthetic pathways to obtain a desired molecule (*e.g.* a drug candidate).

• Design: Let $x \in \mathcal{X}$ be the design for an experiment, where $\mathcal{X}$ denotes the design space of all possible experiments. For Example 1.1 and Example 1.2, each experimental design, x, specifies a candidate molecule, antibody or metal oxide material. x can also represent interventions on specific genes for Example 1.3, or experimental parameters for a specific procedure (*e.g.* synthesis conditions). If an experiment is modeled *in silico*, it can also include the fidelity of the approximations used, with the associated computational cost.

• Parameters: We use θ to represent the parameters of our mathematical model of the underlying phenomenon of interest. Like the experimental design x, θ can parameterize a wide range of objects. For instance, θ could represent the parameters of the physical process of supramolecular interactions/protein binding in Example 1.1, or a set of parameters describing the model of energy capacity and mechanisms of capacity loss in a battery for Example 1.2, or a structural causal model describing the causal interaction genetic pathways in Example 1.3, or more general cases such as parameters of a neural network.

• Outcome: We use y to denote the experimental outcome, for example the yield of a chemical reaction. The outcome, y, may also be multi-dimensional, and may include all measurements recorded during the experiment. y can represent the binding affinity to a target, toxicity to humans and synthetic accessibility in Example 1.1. For Example 1.2, y can represent energy capacity and retention loss, as well as materials purity and X-ray diffraction patterns. In Example 1.3, y can even include images and videos of specific interventions on a population of cells. The experimental outcomes are often structurally rich but might lack the abstractions necessary for effective modeling.

• Dataset: Finally, we denote the data collected from previous experiments as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{M}$.

We model the outcome y as a consequence of the design x. The likelihood—denoted $p(y|\theta, x)$—is parameterized by θ, and is a measure of how likely each experimental outcome, y, is to occur, given a particular design, x, and model parameters, θ. This likelihood acts as our abstract model of the underlying phenomenon. If the likelihood is known analytically or can be computed efficiently it is called an explicit model. For example, we might model the outcome of an experiment with a Gaussian distribution, with mean as some function $f_\theta(x)$, such that $p(y|\theta,x) = \mathcal{N}(f_\theta(x);\sigma)$. Alternatively, if the likelihood is intractable, it is called an implicit model. Implicit models are common in the context of scientific discovery because we often model processes *via* simulators that allow us to sample from $p(y|\theta, x)$, without being able to evaluate the likelihood.

We assume that the data set of observed variables $\mathcal{D} = \{x_1, \ldots, x_n\}$, is drawn from the joint distribution $p(x, \theta)$ for some θ. In order to use that data to update our model of likely outcomes, we need an approach to solving the inference problem: estimating the posterior probability distribution $p(\theta|\mathcal{D})$ over the latent variable given the observed data. This problem appears in various contexts across domains. For instance, given observations of the experimental outcomes, say the binding affinities of several ligands to a particular target, we might be interested in estimating the distributions over parameters in the model that describe the process of binding. In principle this distribution can be estimated using Bayes' rule,

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})}. \tag{1}$$

In practice, however, computing the posterior exactly is intractable for high-dimensional θ and x. Thus, approximate inference methods have been studied extensively. We briefly summarize two broad classes of methods: Markov Chain Monte Carlo (MCMC) and Variational Inference (VI).

3.2 Approximate inference

MCMC methods[18] are designed to generate samples from a target distribution, where a correct sampling procedure is not known but the density of the desired distribution is known up to a normalizing constant. They approximate sampling from the desired distribution by constructing a sequence of samples whose asymptotic distribution (as the sequence becomes longer) matches the desired target distribution. At each step of the sequence a new sample is generated by performing a small random perturbation from the previous sample. When trying to sample from a Bayesian posterior $p(\theta|\mathcal{D})$, we can compute the unnormalized form of the posterior as the product of the prior and the likelihood, $p(\theta)p(\mathcal{D}|\theta)$. Unfortunately, in high-dimensional problems and problems where the modes of the distribution occupy a tiny relative volume and can be far from each other, MCMC methods can take exponentially more time to properly sample from all the modes (or even just move from one mode to another). Methods to improve the performance of MCMC for sampling from high-dimensional distributions[19–21] are limited to certain classes of distributions and do not apply to sampling complex objects such as graphs and sequences which are important in a number of applications in scientific discovery.

Variational Inference methods[22] approach the problem of sampling from the posterior using instead an optimization-based approach, finding a member of family of distributions that is closest to the posterior distribution that we seek. In particular, they search for a distribution $q(\theta)$—by optimizing the values of q or parameters that define it—so as to minimize the reverse Kullback–Liebler (KL) divergence $E_{\theta\sim q}[\log q(\theta) - \log p(\theta,\mathcal{D})]$. Because this measure of "closeness" is a reverse KL, VI methods tend to drop most modes of the true posterior or even focus on just one of them.[23,24]

GFlowNets address this issue of mode dropping that plagues both MCMC and typical VI methods. They are similar to amortized variational methods (*i.e.*, they learn a parameterization for q) but use a different training objective which favors a greater diversity of samples by allowing the use of exploration in the space of samples[25] as we discuss in Section 4.1.

3.3 Experimental design

Experiments are the primary interface for interaction between our abstract models and the complexities of the real world. A key element of scientific methodology has been the careful design of experiments that allow the acquisition of knowledge corresponding to a reliable understanding of the underlying phenomena. However, experiments are expensive—either computationally, financially or in time. Therefore, we need methods to design experiments that maximize the amount of information our models learn from each experiment. This task of automated experimental design has been extensively studied in statistics and machine learning.

The field of experimental design studies the problem of designing "useful" experiments effectively. The usefulness of an experiment is defined by a utility function (or reward) $U(\cdot;\mathcal{D}) : \mathcal{X} \rightarrow \mathbb{R}$, which may change as a function of the data, $\mathcal{D}$, that we have observed from previous experiments. Given this utility function, we are typically interested in selecting the most useful designs, x^*,

$$x^* = \underset{x\in\mathcal{X}}{\operatorname{argmax}}\, U(x;\mathcal{D}). \tag{2}$$

The process of experimental science is often iterative, as illustrated in Fig. 1. We design an experiment, perform the experiment and observe the experimental outcome, update our model based on the observations and then design the next experiment guided by the updated model. This is referred to as sequential experimental design. The sequential experimental design setting is thus formalized at any iteration k as follows, in terms of the estimated utility of the experiment x considered and the past data $\mathcal{D}_{k-1}$:

$$x_k = \underset{x\in\mathcal{X}}{\operatorname{argmax}}\, U(x;\mathcal{D}_{k-1}) \tag{3}$$

where $\mathcal{D}_{k-1} = \{(x_i, y_i)\}_{i=1}^{k-1}$ consists of the designs and outcomes of the experiments performed till iteration k.

Designing utility functions that accurately reflect the value of an experiment while being efficient to compute has been a problem of interest in various communities. Classical work on experimental design relied on the Fisher information matrix to quantify the information about parameters θ contained in the experimental outcome y.[26,27] This measure of information can be efficiently computed in linear models, where the outcome depends linearly on the design.[8] When this relationship is nonlinear, a variety of methods exist to select among a version space of nonlinear functions that are consistent with what we have observed; see Settles[28] for a survey. In scientific discovery, we are not agnostic to the set of functions that could explain our observations: we typically have significant prior knowledge from the literature and previous experiments that what we can use to weigh the relative likelihood of potential experimental outcomes. Bayesian experimental design, introduced below, provides a principled approach to incorporating these priors into our choice of future experiments.

3.4 Bayesian experimental design

Bayesian Experimental Design (BED) or Bayesian Optimal Experimental Design (BOED)[8,29] approaches experimental design by modeling a rational agent that aims to maximize their expected utility of new experiments with respect to prior beliefs. The utility function provides a real-valued score for each potential outcome, y, of any potential experimental design, x. At each round of experimentation, the agent selects an experimental design,† x, that gives the highest expected utility weighted by how likely each outcome is to occur under the agent's prior beliefs (parameterized by θ). By specifying the agent's prior belief, we can encode scientific knowledge of known relationships and uncertainties in the observed outcomes, thereby making the procedure more efficient at exploring unknown parts of the experimental design space.

A common choice for the agent's utility function is the mutual information [MI; ref. 30] between the experimental parameters θ and the outcome y observed upon performing an experiment y, given the dataset $\mathcal{D}$ of previous experiments.

$$U(x;\mathcal{D}) = I(y;\theta|x,\mathcal{D}) \tag{4}$$

$$U(x;\mathcal{D}) = \mathbb{E}_{p(y|\theta,x)p(\theta|\mathcal{D})}\left[\log\frac{p(\theta|y,x,\mathcal{D})}{p(\theta|\mathcal{D})}\right] \tag{5}$$

$$U(x;\mathcal{D}) = \mathbb{E}_{p(y|\theta,x)p(\theta|\mathcal{D})}\left[\log\frac{p(y|\theta,x)}{p(y|x,\mathcal{D})}\right]. \tag{6}$$

MI can be interpreted as how much information can we expect to gather—or equivalently how much we reduce our uncertainty—about some random variable of interest (say the model parameters θ) thanks to the experimental outcome. Eqn (5) and (6) give two equivalent definitions of MI, but both are intractable in general so we will need to rely on approximations.

† Sequences of experiments are also possible, but even more computationally involved.

Assuming access to the likelihood function, $p(y|\theta, x)$, we first need to estimate the various posterior distributions, depending on which of the above formulations of the MI we choose. To sample one of the sums, we need to estimate or at least sample from either the unknown and generally intractable posterior over parameters $p(\theta|\mathcal{D})$, or the marginal likelihood $p(y|x)$ (where θ has been summed out). Estimating high-dimensional posteriors can be challenging and marginalizing out θ in $p(y|\theta, x)$ can also be intractable. Additionally, the MI itself involves a high dimensional integral which can be intractable. Consequently, developing efficient estimators to approximate the MI has been one of the central challenges in BED. Estimators of MI have been developed for both implicit and explicit models. The estimators typically leverage tools from approximate inference, including MCMC and VI discussed in Section 3.1, to approximate the posterior over parameters or the marginal likelihood,[31–36] and the likelihood in the case of implicit models.[37]

3.4.1 Towards amortization. Conventional approaches discussed above consist of two distinct steps: estimating or approximating the posterior over parameters or marginal likelihood to estimate the MI and then maximizing this estimator of MI to find the optimal experiment. Despite being applied in various contexts in scientific discovery,[37,38] each of these steps alone can be computationally expensive.[39] introduced a unified stochastic gradient method to combine estimation of MI and the selection of the optimal experiment. They propose jointly optimizing the parameters ϕ of the MI estimator and the experiment design x using gradient based methods.

Ref. 40 formalizes the conventional BED approach in terms of a design policy π, which directly maps a history of experimental data $h_t = [(x_1, y_1), \ldots(x_t, y_t)]$ to the next experiment design x_{t+1}. Once this policy is trained, the next experiment can be selected directly using the policy, instead of the usual MI estimation and optimization. In essence, the training of the policy amortizes the cost of estimating and optimizing the MI.[41] extends this to implicit models.

These are examples of amortized inference: instead of expensive Monte Carlo sampling to estimate or maximizes some expected value at run-time, we pre-train a function that directly produces an approximation of the desired quantities. We elaborate on learned amortized inference in Section 4, with a focus on GFlowNets and a discussion of its advantages over MCMC methods.

Current BED methods have been limited to continuous design spaces. While recent work has considered extensions to incorporate discrete designs,[42] they are limited to small problem domains. BED methods in general are hard to scale to larger problem settings.

3.5 Bayesian optimization

A typical problem encountered in various domains of scientific interest is that of optimizing the value of some expensive to compute black-box function f. For instance, consider the task of designing novel molecules to inhibit the activity of a particular target protein. We are interested in searching for a molecule $x^{\star}$

which minimizes the binding energy of the molecule with the target protein, f:

$$x^{\star} = \underset{x \in \mathcal{X}}{\operatorname{argmin}}\, f(x). \tag{7}$$

Further, we only observe noisy measurements y on evaluating f, as described by $p(y|f(x))$. In the context of the binding energy, each experimental evaluation can be expensive and take weeks to perform in the lab and the observed results are noisy. Consequently the goal here is to discover $x^{\star}$ with the fewest possible evaluations of f. This problem is studied broadly within the framework of global optimization.[43] Solving the global optimization problem for any general function f is NP-hard in discrete spaces, and intractable without special structural constraints (*e.g.* convexity) on f in the continuous case. Methods for finding approximate solutions to this problem have received significant attention in the literature due to the broad applicability. Among the wide variety of techniques studied, Bayesian Optimization (BO)[44–46] is a popular and widely used approach. Bayesian optimization has been applied extensively to a wide variety of scientific problems.[47–50] Broadly, Bayesian optimization consists of an iterative process to search for the global optimum in a sample-efficient manner, relying on tools from Bayesian Inference. Algorithm 1 provides a general overview of the Bayesian optimization approach.

$$\underset{x \in \mathcal{X}}{\operatorname{argmax}}\, I\left(y; x^{\star} | x, \mathcal{D}\right). \tag{8}$$

This objective resembles the information gain from Section 3.3, where the parameter of interest is $\theta = x^{\star} = \operatorname{argmax}_{x \in \mathcal{X}} f(x)$. Indeed, BO can be viewed as an instantiation of experimental design with an implicit model f, where we are interested in a particular random variable, the location of the maximum value of f, rather than all the parameters.

3.5.1 Acquisition functions. The acquisition function plays a critical role in Bayesian optimization and over the years various acquisition functions have been proposed. Expected improvement[53] was one of the earliest acquisition functions. Upper-Confidence Bound [UCB; ref. 54] and Thompson Sampling [TS; ref. 55 and 56] were inspired by the bandit learning literature. More recently, there have been significant developments in entropy search methods which adopt the information theoretic perspective introduced in eqn (8).[57] and[58] introduced Entropy Search (ES) and Predictive Entropy Search (PES) as acquisition functions based on eqn (8).[59,60] instead considered the mutual information between the outcome and the max value of f rather than the arg max, resulting in the Max Value Entropy Search (MES) acquisition function:

$$\underset{x \in \mathcal{X}}{\operatorname{argmax}}\, I\left(y; f\left(x^{\star}\right) | x, \mathcal{D}\right). \tag{9}$$

Algorithm 1: Bayesian Optimization

```
Input: Oracle $f \sim p(f)$, Initial dataset, $\mathcal{D}_0 = \{(x_i, y_i)\}_{i=1}^{M}$, Acquisition function $\alpha$;
Result: $x_T \approx \arg\max_{x \in \mathcal{X}} f(x)$
for $i = 1 \ldots T$ do
    Infer surrogate model $p(f \mid \mathcal{D})$ given dataset $\mathcal{D}$;
    Acquire $x_i = \arg\max_{x \in \mathcal{X}} \alpha(x, p(f \mid \mathcal{D}))$;
    Observe $y_i \sim p(y \mid f(x_i))$;
    Update dataset $\mathcal{D} = \mathcal{D} \cup \{(x_i, y_i)\}$
end
```

The two key ingredients of a Bayesian optimization algorithm are the surrogate model $p(f|\mathcal{D})$ which approximates f (and its epistemic uncertainty) and the acquisition function $\alpha(x, p(f|\mathcal{D}))$ which quantifies the utility of acquiring a point. Gaussian Processes [GPs; ref. 51] are an appealing choice for the surrogate model owing to simple analytical form for the posterior. Consequently, GPs are the default choice in modern BO methods.[52]

From an information theoretic perspective, in each round we are interested in acquiring candidates that maximize the mutual information between the observed value and the global optimum of the function:

Ref. 61 further proposed considering a lower-bound on the mutual information resulting in a general purpose information-theoretic acquisition function GIBBON, which is also applicable to various extensions we discuss below.

3.5.2 Extensions. Inspired by various practical applications, several extensions to the standard Bayesian optimization setting have been studied. A common scenario is where f can be evaluated on multiple different candidates in parallel. In practice, we can often evaluate multiple candidates with nearly the same cost as a single candidate. For example, phage display can produce libraries of millions of antibodies in one batch. Batch Bayesian optimization[62,63] is an extension of BO where in each round we acquire a batch of candidates instead of a single candidate. Additionally, we might have access to oracles with different costs and fidelities to evaluate f; for example, to obtain the simulated binding affinity of a molecule to a protein, oracles can include free energy perturbation and molecular docking, where the former is substantially more accurate but the computational cost is orders of magnitude higher. This setting is studied in multi-fidelity Bayesian optimization.[64–66] Another important aspect in practical applications is multiple objectives. For example, we are interested in multiple properties such as the drug-likeness, toxicity to humans, and synthesizability in addition to binding energy in the drug discovery setting. Multi-objective Bayesian optimization[58,67,68] methods study this problem setup. Recent work has also incorporated physical inductive biases as priors for efficient Bayesian optimization.[69,70] Finally, while traditional BO methods mainly consider continuous x, recent work has enabled BO on discrete spaces.[71,72] Despite recent progress, BO methods have typically been limited to small problems due to challenges in scaling surrogate models to larger domains. Additionally, as mentioned in Section 2, as BO is concerned with maximizing or minimizing a function, it can miss out on diversity which is critical for many scientific applications.

3.6 Causal discovery

An important goal of the scientific methodology is understanding the causes and effects of certain phenomena based on prior observations and experiments. Causal discovery studies the problem of learning the causal structure from data.

A Bayesian Network[73] is a representation of the joint distribution over d random variables $\{Y_1, \ldots, Y_d\}$, whose conditional independencies are encoded in a compact graphical way. These random variables correspond to the nodes of a directed acyclic graph (DAG) G that determines the factorization of the joint distribution as

$$P(Y_1, \ldots, Y_d; \theta) = \prod_{i=1}^{d} P(Y_i | \mathrm{Pa}_G(Y_i); \theta_i),$$

where $\mathrm{Pa}_G(Y_i)$ is the set of parent nodes of Y_i in the graph G, and θ_i are the parameters of the conditional distribution associated with the random variable Y_i.

Although in a Bayesian Network the edges connecting the nodes in G only encode associations between random variables, a causal graphical model enhances this framework with notions of causality. In a causal graphical model, again represented by a DAG, G, over the random variables, any directed edge $Y_i \rightarrow Y_j$ represents a direct causal influence of Y_i on Y_j. This allows the model to not only represent the joint distribution of the system (*i.e.*, passively observing the system), but also the effects of actively experimenting on it. A causal model specifies the distribution that would be obtained under any intervention. For example, a "DO-intervention" sets the value of a variable, ignoring its usual causes. However, because the same causal mechanisms (the conditionals $P(Y_i|\mathrm{Pa}_G(Y_i);\ \theta_i)$) are shared across all interventions, if the causal mechanisms (*i.e.* θ) and the graph G have been inferred correctly, a causal model can generalize to distributions never seen during training (*i.e.*, out-of-distribution), corresponding to new interventions.

3.6.1 Causal structure learning. The structure G of a causal graphical model is often assumed to be known, where for example the causal relationships are determined using expert knowledge. This allows us to perform a number of tasks using these models, such as inference (either probabilistic, or causal), or learning the parameters θ of the causal model from observations of the system.

However in the context of scientific discovery, the objective is precisely to discover causal relationships that may have eluded experts thus far. For example, in the development of a disease, we want to find what factors (*e.g.* social factors, proteomes, pathogens) are involved. In this situation, we would like to learn the structure of the causal graphical model (or at least part of it) using data, stored in a dataset $\mathcal{D}$. This data could either come from passive observations of the system (called observational data, *e.g.* the statistics of protein expressions in patients), or from active experiments (called interventional data, *e.g.* genome-wide gene perturbation). This problem is known as causal structure learning, or causal discovery.[74–78]

3.6.2 Bayesian causal discovery. Similar to how there may be multiple theories explaining the same phenomenon, there may also be multiple models that could explain our observations equally well, even in the limit where we have a very large amount of data. Concretely, this means that many standard structure learning methods would typically choose an arbitrary model, which could lead to undesirable (and potentially harmful) outcomes. For example, if we had a system with only two (correlated) random variables A and B, there would be no way in general to distinguish between the two causal models $A \rightarrow B$ and $B \rightarrow A$ using observational data only, even though both models have significantly different causal conclusions. Moreover, in practice, the amount of data available in $\mathcal{D}$ to identify the causal model may be scarce, and this introduces another source of variability: since causal discovery methods only return a single candidate, some theory may be favored only due to the limited evidence. Ideally, we would like to quantify our epistemic uncertainty to avoid model misspecification. This can be done using the Bayesian posterior over the causal structures G, given a dataset $\mathcal{D}$, similar to the description in Section. 4.1.6. Using Bayes' rule, the posterior is given by

$$P(G|\mathcal{D}) = \frac{P(\mathcal{D}|G)P(G)}{P(\mathcal{D})}. \tag{10}$$

In the expression above, $P(G)$ represents our prior belief, and may encode some *a priori* knowledge about the structure G. For example, we may encourage causal graphs to be sparse, *i.e.* to limit the number of parents for any node in the graph. While this prior may be designed based on expert knowledge, this usually encodes only soft beliefs about the causal model, and does not represent a single graph G unlike when the structure is assumed to be known. The term $P(\mathcal{D}|G)$ is called the marginal likelihood, and is defined by integrating over all possible values of the causal mechanisms

$$P(\mathcal{D}|G) = \int_{\Theta} P(\mathcal{D}|\theta, G)P(\theta|G)\mathrm{d}\theta, \tag{11}$$

where $P(\mathcal{D}|\theta, G)$ is the likelihood of the data under a specific choice of causal structure and mechanisms, and $P(\theta|G)$ is a prior distribution over causal mechanisms. The marginal likelihood represents how well the data $\mathcal{D}$ fits a certain hypothesis, given by the causal model G, regardless of the choice of the causal mechanisms themselves.

As is typically the case in Bayesian statistics, the difficulty in evaluating eqn (10) arises from the marginal evidence $P(\mathcal{D})$, which is almost always intractable. To circumvent this issue, approximations of the Bayesian posterior are often necessary, for example based on MCMC,[79–83] bootstrapping,[84,85] or more recently variational inference.[86–90]

3.6.3 Active learning of causal structures. With an approximation of the Bayesian posterior over causal graphs, we can also leverage the tools from Bayesian experimental design in Section 3.3 in order to design interventions on the system that would refine our beliefs about its causal structure.[91] This creates a feedback loop between the estimation of our uncertainty about the causal graph based on data, the decisions about which experiments to perform, and the acquisition of new experimental data (see Fig. 1). Active causal discovery can be used to learn the causal structure of either the whole system,[92–95] or part of it.[85] Recently, Toth *et al.*[96] proposed a novel framework called Active Bayesian Causal Inference (ABCI) to infer not only the causal graph, but also jointly learning the posterior over causal queries of interest.

4 Generative Flow Networks

We begin by introducing the broader ideas around using neural networks to efficiently learn a mapping from sampled observations to proposed high-dimensional probability distributions and intractable sums, as a general substitute for the popular MCMC-based inference. These ideas lead into GFlowNets, which provide a general framework for amortized inference with neural networks.

4.1 Learning to perform amortized inference

Let us first look at how neural nets can be used to amortize the inference problem introduced in Section 3.1, *i.e.*, by being trained to approximately perform the sampling or summing task that is otherwise intractable. Specifically, we consider how an intractable expectation or sum can be transformed into a tractable training task to approximate the desired sum. This is the fundamental principle underlying GFlowNets.

4.1.1 Simple mean squared error criterion to amortize an intractable expectation. Consider a set of intractable expectations that we would like to approximate, for a pair of random variables x and y that can both take an exponential number of values or live in a high-dimensional space:

$$S(x) = \sum_y p(y|x)R(x,y) \tag{12}$$

which is then intractable because of the exponential number of terms in the sum.

If we know how to sample from $p(y|x)$, we could, however, train a neural net $\hat{S}$ with input x, stochastic target output $R(x, y)$ and Mean Squared Error (MSE) loss

$$L(x,y) = \left(\hat{S}(x) - R(x,y)\right)^2 \tag{13}$$

where $y \sim p(y|x)$, to train the estimator $\hat{S}$ with parameters θ. When we sample training examples (x, y), the stochastic gradients $\frac{\partial L(x,y)}{\partial \theta}$ would make $\hat{S}$ converge to S if it has enough capacity and is trained long enough.[97]

For any new x, we would then have an amortized estimator $\hat{S}(x)$ which in one pass through the network would give us an approximation of the intractable sum $S(x)$. We can consider this an efficient alternative to doing a Monte Carlo approximation

$$\hat{S}_{\mathrm{MC}}(x) = \mathrm{mean}_{y\sim p(y|x)} R(x,y), \tag{14}$$

which would require a potentially large number of samples and computations of $R(x, y)$ for each x at run-time, especially if $p(y|x)$ $R(x, y)$ is a rich multimodal function (for which averaging just a few samples of y does not give us a good estimator of the expectation).

Besides the advantage of faster run-time, a crucial potential advantage of the amortized version is that it could benefit from generalizable structure in the product $p(y|x)R(x, y)$: if observing a training set of $(x, y, R(x, y))$ triplets can allow us to generalize to new (x, y) pairs, then we may not need to train $\hat{S}$ with an exponential number of examples before it captures the generalizable structure and provides good answers (*i.e.*, approximates $E_{Y|x}$[-$R(x, Y)$] well) on new x's. This ability to generalize from structure in the data is actually what explains the remarkable success of ML (and in particular of deep learning) in the vast set of modern AI applications.

When we do not have a $p(y|x)$ that we can sample from easily, we can, in principle, use MCMC methods that form chains of samples of y's whose distribution converge to the desired $p(y|x)$, and where the next sample is generally obtained from the previous one by a small stochastic change that favors increases in $p(y|x)$. Unfortunately, when the modes of the summand, $p(y|x)R(x, y)$, occupy a small volume in the search space (*i.e.* throwing darts does not find them) and these modes are well-separated (by low-probability regions), especially in high dimension, it tends to take exponential time to mix from one mode to another. However, such an MCMC approach leaves money on the table: the attempts $(x, y, R(x, y))$ contain information that one could use to train an ML model. To the extent that the space is sufficiently structured, such a model could guess where the yet unseen modes might be given the location of the already observed modes, as illustrated in Fig. 3.

4.1.2 GFlowNet criterion to obtain a sampler and estimate intractable sums. Let us consider the situation where we do not have a handy $p(y|x)$ and our objective is just to approximate a set of intractable sums (for any x)

$$S(x) = \sum_y R(x,y) \tag{15}$$

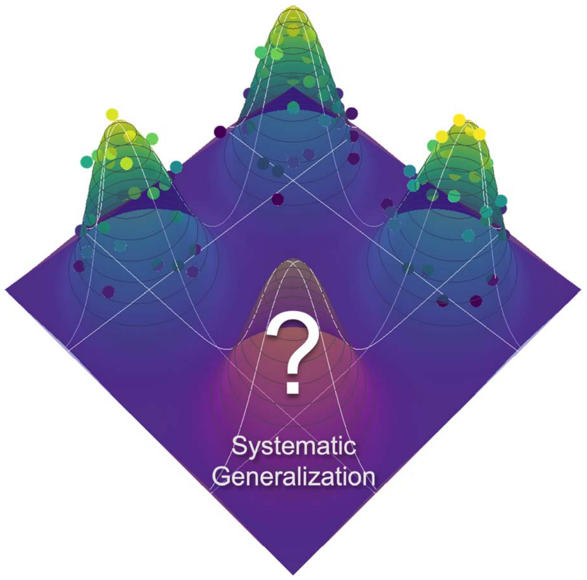


Fig. 3 An illustration of the feasibility of systematic generalization enabled by ML methods: if we have already discovered the three modes shown of a reward function, a learner that can generalize may guess the presence of a 4th mode, because the first three modes seem to align on a grid. The existence of such generalization structure is why amortized ML samplers can potentially do much better than MCMC samplers.

where we have the constraint that $R(x, y) \geq 0$ and $S(x) > 0$. This may be useful to estimate a normalization constant for energy-based models or Bayesian posteriors (where y corresponds to learnable parameters and x to observed data). Hence we may also be interested in the sampling policy

$$\pi(y|x) = \frac{R(x,y)}{S(x)} \propto R(x,y). \tag{16}$$

Now, GFlowNet losses are derived from a set of constraints that we would like to be true:

$$\forall (x, y){:}\ \pi(y|x)S(x) = R(x, y) \tag{17}$$

We can define estimators $\hat{\pi}$ and $\hat{S}$ and train them with a loss such as

$$L(x,y) = (\hat{\pi}(y|x)\hat{S}(x) - R(x,y))^2 \tag{18}$$

or with an interpretation of R as unnormalized probabilities that we want well calibrated in the log-domain,

$$L(x,y) = (\log(\hat{\pi}(y|x)\hat{S}(x)) - \log R(x,y))^2 \tag{19}$$

where (x, y) are sampled from a training distribution $\tilde{p}(x, y)$ that has full support. It can then be shown[11,12] that with $\hat{\pi}$ and $\hat{S}$ with enough capacity and trained for long enough they both converge to their desired value:

$$\hat{S}(x) \to S(x)$$
$$\hat{\pi}(y|x) \to \pi(y|x) \propto R(x,y).$$

This is a crucial property of GFlowNets: they are trained to sample objects y (given x) with probability proportional to a given reward function $R(x, y)$.

4.1.3 Marginalizing over compositional random variables. To make the notion of intractable sum more concrete, it is good to think of y (and potentially x as well) as a compositional object, like a subset of variable-value pairs or a graph. For example, we can construct compositional objects sequentially through a series of steps where a new piece of the compositional object is inserted at each step, such as the molecular fragments composed to form a larger molecule in Fig. 2b. The sampling policy π then sequentially and stochastically chooses a constructive action at each step, and after each step we get a partially constructed object s which we call a GFlowNet state. A sequence of such states and actions forms a GFlowNet trajectory τ. In the basic GFlowNet framework, the actions are stochastic, but the next state is deterministically obtained from the previous state and the action, because they are not happening in an external environment but are part of the internal computation of a sampler. In addition, the GFlowNet mathematical results, as they currently stand, assume that each step is constructive, *i.e.*, we cannot return to the same partially constructed object s twice. This means that the set of all possible trajectories forms a directed acyclic graph (DAG), illustrated in Fig. 2. Because the transitions are deterministic, we can specify a trajectory τ with a sequence of states (or, equivalently, an initial state and a sequence of actions). A special "exit" action is also defined to declare the construction of the object y is completed. The policy $\pi(y|x)$ is now specified by a forward transition distribution $P_F(s|s')$ which specifies how to generate each constructive step given the previous state, and we are interested in parameterizing and learning this P_F.

For instance, consider the molecule graph example illustrated in Fig. 2. We can construct a molecule graph sequentially using nodes representing atoms as building blocks. Starting from a special empty state, which we denote as s_0 (this can be an empty graph, null set or empty sequence or chosen based on the value of a conditioning variable x, maybe specifying some desired characteristics of the molecule), the object y can be constructed through a sequence of steps, each consisting of adding a single block $a \in \mathcal{A}$. We assume that the actions are limited to be constructive, and deletion of blocks is not allowed. At each step, we have a partially-constructed object $s \in \mathcal{S}$, where $\mathcal{S}$ denoted the space of all possible partially-constructed objects and $\mathcal{Y} \subset \mathcal{S}$. Another assumption we make throughout is that these states are Markovian, that is, they incorporate all the information from their history. This results in a directed acyclic graph (DAG) $\mathcal{G}$ which is defined by a tuple $(\mathcal{S}, \mathcal{E})$, where the set of nodes corresponds to $\mathcal{S}$, and an edge $s \to s' \in \mathcal{E}$ indicates that object s' can be constructed by adding a block $a \in \mathcal{A}$ to s, $s \xrightarrow{a} s'$. We can define a trajectory as a sequence of steps describing the construction of an object $\tau = (s_0 \xrightarrow{a_1} s_1 \ldots \xrightarrow{a_n} y)$.‡ Let $R(x, y)$ denote the utility (reward) for a given object y in context x. For instance,

‡ Note that there can be multiple trajectories resulting in the same object at the end.

it can be the binding energy for the ligand molecule y with a given target protein x.

Formally, the training objective is to learn a stochastic policy π which sequentially generates an object y with a probability proportional to its reward, *i.e.*, $\pi(y|x) \propto R(x, y)$.

4.1.4 Multiple parent states. Besides the sequential nature of the generative process for y, an interesting complication is that there may be many ways (in fact exponentially many trajectories) to construct y from some starting point and context x. This means that a partially constructed object, *i.e.*, a state s, may have multiple parents s' for which an action a exists that leads to s. Otherwise (when each state only has one parent), the DAG is a tree, which makes the computation much simpler. But when it is not a tree, it turns out to be convenient to consider and parameterize a backward transition probability function $P_B(s'|s)$ which is consistent with that DAG and the associated forward transition probabilities P_F. The constraint in eqn (17) can be reformulated in several ways, in particular what is called the detailed balance constraint:

$$\forall (s, s'): P_F(s|s')F(s') = P_B(s'|s)F(s) \qquad (20)$$

where $F(s)$ is called the flow at state s and plays a role similar to $S(x)$ above, *i.e.*, it is an intractable sum, and there is a starting state $s_0 = x$ from which a trajectory is initiated, as well as a constraint that the flow into a terminal state $s = y$ equals $R(x, y)$. Similarly to the simpler case above, this can be turned into a training loss that we want to minimize, but now over all (x, τ) pairs or over all (s, s') pairs. As a consequence of satisfying the detailed balance constraint at all (s, s') pairs, the initial flow becomes equal to the normalizing constant:[11]

$$F(s_0) = S(x) = \sum_y R(x, y). \qquad (21)$$

4.1.5 Learning objectives. In practice, we would like to approximate $P_F(\cdot|\cdot;\theta)$, $P_B(\cdot|\cdot;\theta)$, and $F(\cdot;\theta)$ with learnable parameters θ, and we want to choose those parameters to satisfy as well as possible the detailed balance constraint $\forall s, s' \in \mathcal{S}$. The detailed balance constraint can be converted to the following loss to learn the parameters θ:

$$\mathcal{L}_{DB}(s, s'; \theta) = \left(\log \frac{P_F(s'|s;\theta)F(s';\theta)}{P_B(s|s';\theta)F(s;\theta)}\right)^2. \qquad (22)$$

Several alternative learning objectives for GFlowNets have been proposed, especially for longer trajectories to sample the object y.[98,99] Trajectory balance [ref. 98, TB] is a prominent learning objective for training GFlowNets. Contrary to the detailed balance objective, which considers constraints on pairs of states, trajectory balance jointly applies the detailed balance constraint over entire trajectories. A learnable parameter Z is introduced which at convergence is equal to the desired sum. The trajectory balance loss over a trajectory τ is defined as follows:

$$\mathcal{L}_{TB}(\tau; \theta) = \left(\log \frac{Z_\theta \prod_{s\to s' \in \tau} P_F(s'|s;\theta)}{R(x, y)\prod_{s\to s' \in \tau} P_B(s|s';\theta)}\right)^2. \qquad (23)$$

Algorithm 2 illustrates the typical approach for training GFlowNets, by sampling trajectories from a sampling policy $\hat{P}_F$ which is typically a tempered P_F or a mixture of P_F with a uniform policy to enable exploration, and optimizing the loss induced by the learning objective with respect to θ, with stochastic gradient descent.§ As a result of this procedure, we learn a P_F with the objective that the marginal likelihood of a trajectory terminating at a terminal state x, denoted as $\pi(x)$ become proportional to the reward $R(x)$. The learnable objects P_F, P_B, F are typically parameterized by neural networks. These neural networks must have appropriate inductive biases depending upon the type of objects we are constructing. These neural networks must also have enough capacity to model the underlying distribution. In Section 4.2 and Section 4.3 we discuss specific cases of leveraging GFlowNets for problems of molecule generation and causal modeling respectively.

Algorithm 2: General recipe for training GFlowNets

```
Input: Reward function $R(x, y)$, Learning objective or training loss $\mathcal{L}$ ;
Initialize: $F(\cdot;\theta), P_F(\cdot \mid \cdot;\theta), P_B(\cdot \mid \cdot;\theta)$ for DB and $Z_\theta, P_F(\cdot \mid \cdot;\theta), P_B(\cdot \mid \cdot;\theta)$ for TB;
for $i = 1 \ldots N$ do
    Sample trajectory $\tau$ from the sampling policy $\hat{P}_F(\cdot \mid \cdot;\theta)$ ;
    Compute the training loss $\mathcal{L}$;
    Update parameters $\theta$ with stochastic gradient descent;
end
Result: Policy $\pi(x) \propto R(x)$
```

§ Code implementing various GFlowNet learning objectives on simple synthetic domains: saleml/gfn.

4.1.6 Implications for Bayesian ML. Let us consider the special case where $y = \theta$ is a latent parameter, *i.e.*, in a Bayesian setting, and $x = \mathcal{D}$ is the available data. Then we can define the reward function

$$R(\mathcal{D}, \theta) = P(\theta)P(\mathcal{D}|\theta) \tag{24}$$

from the parameter prior $P(\theta)$ (how plausible are these parameters *a priori*) and the data likelihood $P(\mathcal{D}|\theta)$ (how well does this choice of parameters fit the data). Training a GFlowNet provides us with an approximate sampler for the posterior over parameters (the policy $\pi(\theta|\mathcal{D})$) given data as well as an estimator of the normalizing constant of the Bayesian posterior, through the learned initial flow $S(\mathcal{D})$. Hence, we have used amortization to turn a tractable function (prior times likelihood, as a function of θ) into estimators of these generally intractable quantities. We get a fast sampler for the posterior with no need for a Markov chain going through a large number of candidate samples. With that sampler, we can generate many independent samples $\theta|\mathcal{D}$ (possibly in parallel) that are likely to visit the larger modes of the posterior (where the reward is larger).

To make computations more ML-friendly (especially for large datasets) while training the GFlowNet, we can note that the GFlowNet squared loss objectives naturally lend themselves to the case where the reward or log-reward is stochastic and is an unbiased estimator of the true reward. For example, we can typically decompose the overall dataset log-likelihood $\log P(\mathcal{D}|\theta)$ into a sum of per-example or per-minibatch terms, and we can introduce a multiplicative correction to account for the prior $P(\theta)$:

$$\log \hat{R}(Z, \theta) = \log P(\theta) + |\mathcal{D}| \log P(Z|\theta)$$
$$E_Z\left[\log \hat{R}(Z, \theta)\right] = \log P(\theta) + \sum_{Z \in \mathcal{D}} \log P(Z|\theta) = \log R(\mathcal{D}, \theta)$$

where the expectation over Z is just a sum over the Z's in $\mathcal{D}$. This makes it possible to train the GFlowNet posterior estimator using stochastic gradient descent on single examples or minibatches, which is the state-of-the-art to train deep nets.

4.1.7 Why GFlowNets? Let us look at how GFlowNets differ from other related conceptual frameworks:

• Markov Chain Monte Carlo: GFlowNets do not construct Markov chains with semantics like those in MCMC methods. GFlowNets and MCMC approaches differ fundamentally: with MCMC approaches, the chain is irreducible (every state is reachable from every other state), and the stationary distribution of this chain is of interest. In order to generate samples from this stationary distribution, we need to run long chains (ideally infinitely long ones). In GFlowNets, however, the chains only need every state to be accessible from the initial state, and what we have is a bounded sequence of stochastic transitions. The expensive stochastic "search" (to reach low energy configurations) normally performed by MCMC is replaced by the training phase of the GFlowNet, using the principle of amortized inference, so that during inference a sample can be generated in a single short trajectory by the policy. To exemplify, if we were to run an MCMC to generate samples of desirable molecules, we would probably have a molecule at every state, and we would have actions that can stochastically transform one molecule into a nearby one,[100] whereas a typical way to sample a desirable molecule with GFlowNets is constructive, where we start from an empty object (which is not really a molecule) and we sequentially and stochastically add molecular fragments: the state corresponds to these intermediate objects, which may or may not correspond to a well-formed molecule.[11] That being said, the overall objective is the same: turn a given energy function (or unnormalized distribution) into a generative procedure for obtaining samples from that distribution. Both MCMC and GFlowNets can also be used to marginalize, *i.e.*, compute intractable sums, although again in different ways. MCMC turn the exact sum into a Monte Carlo approximation of it while GFlowNets perform amortized inference, *i.e.*, they train a neural net whose output will, after training, approximate the sum. This becomes useful when we have more than one marginalization to do, say given a context, and thus the neural net can take that context as input and rapidly produce an estimator of the intractable sum as output.

• Reinforcement learning: GFlowNets learn policies to sample trajectories that land in a terminal state with probability proportional to the reward of the terminal state rather than trajectories that maximize the expected reward, as in standard deep reinforcement learning. As shown by Bengio *et al.*,[11] Jain *et al.*,[101] this results in a diversity of samples which is important when the reward function is an imperfect proxy for the property that we actually care about: it avoids putting all our eggs in the wrong basket.

• Deep generative models: traditional generative models in deep learning such as variational auto-encoders or VAEs[102,103] or GANs[104] require positive samples to model the distribution of interest, whereas GFlowNets are trained from a reward function.

• Variational inference: variational inference trains an approximate sampler (and the corresponding density) so as to reduce the forward KL-divergence (the evidence lower bound or ELBO) with a given distribution function (playing the same role as the reward). This requires on-policy training (sampling from the learned sampler), which has a tendency for focusing on a single mode rather than find a diversity of modes. Instead (see Malkin *et al.*[25] for details), GFlowNet objectives enable off-policy training without requiring the high-variance importance sampling correction necessary with the ELBO.

To summarize, GFlowNets shine in problems with the following properties:

• It is possible to define or learn a non-negative reward function which will specify from what distribution the GFlowNet should sample.

• The reward function of interest is highly multimodal. This emphasizes the advantage of GFlowNets in terms of diversity of samples. If the reward function was unimodal, existing RL or variational inference methods (which tend to focus on a single mode) could be used instead.

• It is advantageous to sample sequentially, *e.g.*, there is compositional structure that can be exploited by sequential generation.

4.1.7.1 Current limitations. Until recently, GFlowNets have been limited to sampling from distributions over discrete objects (*e.g.* graphs). Recent work by Lahlou *et al.*[105] presents a theoretical framework for extending GFlowNets to sample from distributions over continuous spaces. Leveraging this framework for sampling from distributions over high-dimensional continuous or mixed (discrete and continuous) spaces remains an open problem. Another potential limitation, shared with other reinforcement learning methods, is that effective credit assignment over very long trajectories (for generating large objects, such as proteins) is more difficult. Pan *et al.*[106] take initial steps to tackle this problem, proposing a way to assign partial rewards earlier in the generated trajectory which results in more effective credit assignment. Another open question is that of the best policy for sampling training trajectories for a GFlowNet. Existing theoretical results from Bengio *et al.*[12] assume that the policy sampling trajectories should have full support over the space of trajectories, but designing this policy (beyond the heuristics discussed in Section 4.1.5) for sample-efficient learning is an open problem.

4.2 Diverse candidate generation

A fundamental problem in chemistry is the synthesis of novel chemical structures (*e.g.* molecules) that satisfy some criteria. As alluded to in Section 4.1, generation of molecules to optimize for a particular chemical property is an appealing use case for GFlowNets, because GFlowNets will tend to generate a diverse set of molecules optimizing that property. An important problem in the context of drug discovery, which we introduced in Example 1.1 is to discover molecules that bind to a particular target, potentially inhibiting the target in the process. From the computational design perspective, molecular docking simulations can give scoring functions to approximately evaluate proposed molecules. More recently, graph neural networks which approximate the binding energy[107] are used to approximate docking as they are much faster. As discussed in Section 2 as these scoring functions serve as approximations to the underlying process, it is important to generate diverse candidates for downstream applications, to avoid putting all our eggs in the same basket.

Ref. 11 leverage GFlowNets for the problem of diverse molecule generation.¶ Soluble epoxide hydrolase (sEH) in the 4JNC configuration is studied as a target in the paper. It is a useful target as it plays a role in certain respiratory and heart diseases.[108,109] Autodock Vina[110] was used for docking the generated molecules to evaluate the binding energy. Docking each molecule with Autodock Vina can be quite slow and takes several minutes to run, making it prohibitively expensive to train a policy directly using it as a reward. Instead, the authors rely on a graph neural network, trained using a data set of docking scores for 300 000 molecules, as the reward for training the policy. The molecules are generated using fragments, as illustrated in Fig. 2. At each step, the policy picks a fragment from a library to add to the partially constructed molecule, and choose where to place that fragment. The library of fragments is derived from the Zinc database.[111]

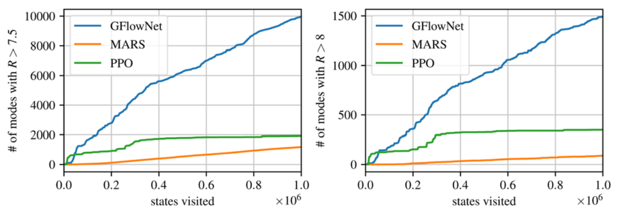
¶ Code for molecule generation recursionpharma/gflownet.

The molecule design problem possesses all the key properties discussed in Section 4.1.7 for GFlowNets to be effective – (a) there is compositional structure in generation as molecules are built using subgraphs with unique chemical properties (b) the reward function is an approximation of what we really care about, as the docking score and its approximation by a neural network (which has epistemic uncertainty associated with it due to finite training) are approximations of the underlying phenomenon of a molecule binding to a target and inhibiting it, and (c) the reward is multi-modal since there can be multiple motifs of molecules that bind well to a given target.

Ref. 11 showed that GFlowNets result in substantial improvements over existing methods on this molecule generation task. In particular, as shown in Fig. 4a, GFlowNets discover significantly more modes of the reward function (*i.e.* many different molecules that have high predicted docking score) relative to other reinforcement learning (PPO) and MCMC (MARS) approaches. Sampling proportional to the reward results in high reward and diverse samples. Though it is important to note that while using GFlowNets results in significant improvements in the diversity of generated samples, they do not always lead to the highest scoring candidates, because there is a natural trade-off between diversity and reward.[112] introduce metrics to study the ability of GFlowNets to explore novel regions in molecular space.

Further,[11] also consider an active learning setup, starting with a data set of 2000 molecules. In each round, a surrogate model is trained on the data set. This surrogate model is used as the reward for the GFlowNet. Next, molecules are generated with the GFlowNet policy, evaluated with docking, and added to the data set for the next round. Using GFlowNet to acquire the batches of molecules results in significant improvements in the reward over the initial data set, shown in Fig. 4b, demonstrating the potential of GFlowNets to accelerate large-scale virtual screenings.

From a practical perspective, we are often interested in multiple objectives rather than a single one. For instance, in the context of drug discovery, an ideal drug candidate should specifically inhibit the target but also be synthesizable in large quantities, soluble, and harmless to humans; alternatively, in material science, to have an efficient solar cell means the optimization of current, voltage, and fill factor. Typically, there are very few candidates which simultaneously satisfy all the objectives, which might even be conflicting with each other (*e.g.* in solar cells, photocurrent can increase with a photoactive material with lower bandgap, but the voltage decreases). Instead, there exists a set of candidates with the optimal trade-offs between the objectives where further optimizing an objective is impossible without compromising another, *i.e.* the Pareto front. Moreover, as with the single objective case, diversity is still critical in the multi-objective case. Multi-Objective GFlowNets [ref. 113, MOGFNs;] extend GFlowNets to tackle multi-objective optimization problems. Building upon scalarization approaches in multi-objective optimization,[114] MOGFNs decompose the multi-objective optimization problem into a family of sub-problems which can solved simultaneously. This family of sub-problems is modeled simultaneously with


(a) Diversity in generated molecules

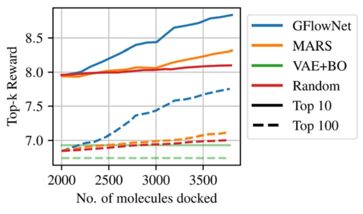

(b) Active Learning

Fig. 4 (a) Molecules generated with GFlowNets cover significantly more modes of the reward distribution, resulting in diverse high reward molecules. (b) Acquiring molecules generated with GFlowNets results in significant improvements over the starting pool of molecules. Figures taken from [11] with permission.

a reward conditional GFlowNet.[12] MOGFNs demonstrate state-of-the-art performance on a variety of small molecule generation and protein design tasks. MOGFNs consistently generate diverse pareto-optimal candidates. For instance, MOGFNs are able to generate molecules that bind to the sEH target, while achieving a high synthesizability and QED score.

The active learning setting is further explored by.[101]‖ Incorporating ideas from Bayesian optimization discussed in Section 3.4, GFlowNet-AL[101] incorporates information about the epistemic uncertainty of the surrogate model in the reward for the GFlowNet with an acquisition function. This epistemic uncertainty helps in guiding the GFlowNet to optimize the promising less explored regions in the state space. As such, instead of maximizing the acquisition function in Algorithm 1,[101] propose sampling proportional to the acquisition function. Equipped with information about the epistemic uncertainty in the reward and other improvements, GFlowNet-AL outperforms various existing methods on a variety of biological sequence design tasks, including generation of peptide sequences with antimicrobial properties. The state space $\mathcal{S}$ consists of partially constructed sequences with each action being the addition of a token from a vocabulary (*e.g.* a residue from a set of amino acids) to the end of the current partial sequence. Candidate sequences generated by GFlowNets are significantly more diverse and have high rewards. The diversity of generated candidates demonstrates the potential of GFlowNets to accelerate the process of discovering novel antibiotics to tackle the growing and highly concerning phenomenon of antimicrobial resistance.[115]

These initial empirical successes demonstrate the potential of GFlowNets to make a significant impact in improving experimental design for a wide variety of scientific problems.

4.3 Modeling posteriors over causal models

4.3.1 Bayesian causal discovery with GFlowNets. As we have seen in Section 4.1.6, GFlowNets offer a general solution to approximate Bayesian posteriors, like the one in eqn (10). GFlowNets are all the more adapted to the problem of Bayesian causal discovery that causal structures G, represented as a DAG, are compositional objects. Deleu *et al.*[116] ** used this observation to introduce a GFlowNet whose states are DAGs, and where some graph G is created sequentially by adding one edge at a time, starting from the completely disconnected graph over d nodes; the structure of this GFlowNet is shown in Fig. 5 (left). Similar to eqn (24), for a fixed dataset $\mathcal{D}$, the reward function of the GFlowNet is defined as $R(G) = P(\mathcal{D}|G)P(G)$. By constraining the set of valid actions at every state, the edges are added in such a way that they will never introduce a cycle, which guarantees that graphs remain acyclic at every stage of the construction. Therefore, all the states of the GFlowNet are valid causal structures. Deleu *et al.*[116] leveraged this property and showed that such a GFlowNet may be trained using a modification of the detailed balance loss [ref. 12, see also eqn (20)], specifically adapted to the case where all the states are terminating.

To evaluate the reward function $R(G)$ though, one needs to evaluate the marginal likelihood $P(\mathcal{D}|G)$ in eqn (11), which is in general intractable.[117] Deleu *et al.*[116] experimented only with models such as multinomial-Dirichlet (for discrete data) and linear-Gaussian (for continuous data), for which the marginal likelihood may be computed efficiently in closed form. Alternatively though, instead of approximating the (marginal) Bayesian posterior $P(G|\mathcal{D})$ over structures only, we could approximate the posterior $P(\theta, G|\mathcal{D})$ over both the causal structures G and the causal mechanisms θ,[118] to avoid the intractable integration in eqn (11).

4.3.1.1 Beyond DAGs. Work on causal discovery focuses primarily on the causal graphical model framework introduced in Section 3.5 that assumes a DAG structure, but acyclicity is indeed an assumption that may not hold in certain domains. For instance, in gene regulatory networks, there are feedback loops between multiple genes interacting with one another.[119] Nevertheless, when we consider the temporally unfolded cyclic graph, *i.e.*, the dynamics, we are back to a DAG. Some recent work has studied the problem of learning the structure of non-acyclic causal models.[120–122] The GFlowNet approach discussed above naturally extends to cases where the causal graph might be cyclic. The masking mechanism introduced by Deleu *et al.*[116] prevents cycles from being introduced at every step where an edge is being added, ensuring the generated graphs are DAGs. If we remove this additional constraint and allow the GFlowNet to introduce cycles in the generated graph, then it would

‖ Code for active learning with biological sequences: mj10/BioSeq-GFN-AL and alexhernandezgarcia/gflownet.

** Code for modeling posterior over causal models: tristandeleu/jax-dag-gflownet.

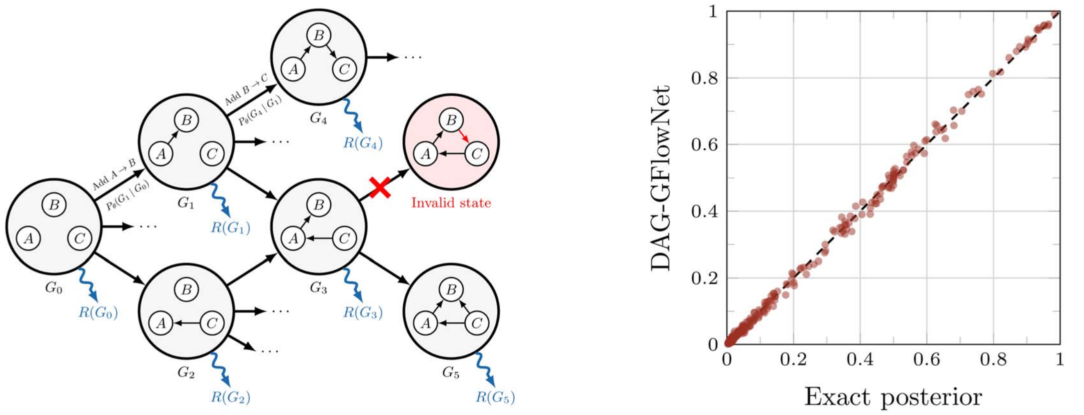


Fig. 5 GFlowNet for Bayesian causal discovery. (Left) The structure of the GFlowNet introduced in ref. 116, where directed acyclic graphs (DAGs) are constructed one edge at a time. Each DAG is associated with a reward $R(G)$. (Right) Comparison between the edge marginals computed using the exact posterior distribution, and approximated using the GFlowNet; the GFlowNet is capable of accurately approximating the exact posterior $P(G|\mathcal{D})$. Used with permission from Tristan Deleu.[116]

approximate the posterior distribution over cyclic causal models which fit the given observations.

4.3.2 Bayesian posteriors for scientific discovery

4.3.2.1 Posterior predictive. Once the structure of the causal graphical model is known, we can use the model to perform inference, *i.e.* answering (possibly causal) questions about the system of interest, in the context of different interventions (which we can interpret as setting some variables, *i.e.*, designing an experiment). If we have information about the epistemic uncertainty, through the Bayesian posterior $P(G|\mathcal{D})$, we can even go one step further and average out the predictions made with all possible causal models, weighted using the posterior distribution. Concretely, given a new observation y in the context of an intervention x this corresponds to evaluating

$$P(y|x, \mathcal{D}) = \sum_G P(y|x, G, \mathcal{D})P(G|\mathcal{D}). \tag{25}$$

This is called the posterior predictive, or Bayesian model averaging.[123,124] The advantage of eqn (25) over making predictions using a single causal model is that multiple concurrent theories may now participate in those predictions. In this way, we avoid using only theory, which may be incorrect, and we obtain a more conservative answer, thus avoiding catastrophic outcomes due to a single theory (say a particular causal graph G) being confidently wrong.

4.3.2.2 Amortized posterior predictive. Instead of performing a Monte Carlo average to estimate $P(y|x, \mathcal{D})$ for each candidate x as per eqn (25) (which may be fairly expensive if we want to train a policy that is trained by considering a large number of possible x's), we can use a neural network $g_\phi(x, y)$ to amortize that calculation. This can be done by training g over (x, y, G) triplets with squared loss

$$\mathcal{L}(\phi) = \mathbb{E}_{P(G|\mathcal{D})}\left[\mathbb{E}_{P(y|x,G,\mathcal{D})}\left[\left(g_\phi(x,y) - P(y|x,G,\mathcal{D})\right)^2\right]\right]. \tag{26}$$

Other amortization approaches are possible. For example, using the GFlowNet framework, a policy $Q(y_i|x_i, y_1^{i-1}, x_1^{i-1})$ can be trained to first sample one outcome y_i at a time, given the input experiment specification x_i and the previous experimental results y_1^{i-1} of the previous experiments x_1^{i-1}. The GFlowNet constraint to satisfy is that

$$P(\mathcal{D}, G) = Q(\mathcal{D}, G) \tag{27}$$

$$P(G)P(\mathcal{D}|G) = Q(\mathcal{D})Q(G|\mathcal{D}) \tag{28}$$

$$P(G)\prod_{i=1}^{|\mathcal{D}|} P(y_i|x_i, G) = Q(G|\mathcal{D})\prod_{i=1}^{|\mathcal{D}|} Q\left(y_i|x_i, {y_1}^{i-1}, {x_1}^{i-1}\right) \tag{29}$$

where the trained posterior predictive takes explicitly a partial dataset (x_1^{i-1}, y_1^{i-1}) as input, similarly to neural processes[125] and $Q(G|\mathcal{D})$ is the GFlowNet causal graph sampler as described above, except that we allow interventions (different choices of x) in the data.

4.3.2.3 Interpretability. Bayesian posteriors over causal models also provide a natural tool for interpretability, since they encode the belief that a causal structure fits the observed data. By inspecting which causal structures contain a certain edge, we can obtain a belief that certain causal relationships between two random variables exist.[126] This is called the edge marginal distribution:

$$P(Y_i \to Y_j \in G \mid \mathcal{D}) = \sum_G \mathbb{1}(Y_i \to Y_j \in G)P(G \mid \mathcal{D}) \tag{30}$$

Fig. 5 (right) shows a comparison of the edge marginals computed with the posterior approximation returned by DAG-GFlowNet[116] against the exact edge marginals, highlighting the capacity of GFlowNets to accurately approximate the posterior over graphs $P(G|\mathcal{D})$. Note that this kind of comparison is typically limited to small problems where the true posterior $P(G|\mathcal{D})$ may be computed efficiently in closed-form, and in general one may be concerned with the calibration of these estimated marginals.[90]

5 Towards a unified framework for scientific discovery with GFlowNets

In this paper, we have introduced GFlowNets as a tool for modeling and for experimental design in the context of the scientific discovery loop (Fig. 1). We have summarized how they have been used and could be further used on both fronts. In this concluding section, we outline research directions based on this early work and aimed at providing scientists with a powerful ML-based framework applicable when it is possible to iteratively generate informative experimental data.

5.1 Exploiting amortized causal Bayesian modeling for defining the utility of an experiment

An appealing theoretical framework for defining the objective of an experiment is that of information gain introduced in Section 3.2: "how much information about a random variable of interest can we expect to gain through the experiment?" A good policy for experimental design should propose experiments with a high value of this information gain as a reward. Note that this framework is broadly applicable to a wide range of interactive learning domains, such as reinforcement learning and active learning, encapsulating the fundamental problem of exploration. In general, the decision to perform an experiment may not simply be based on the number of bits of information gained but also on the risks and costs involved.[127] We can however incorporate the notion of a cost or budget by considering the information gain per unit of cost incurred as the reward.

Information gain can be measured in principle by the mutual information between the outcome of an experiment (a random variable since the experiment has not taken place yet) and the variable of interest (about which we seek to gain information), given the experimental specification and any other knowledge (including data) we may already have. In the simplest and purely unsupervised knowledge-seeking scenario, the variable of interest may be the causal model explaining the outcomes of experiments. In a more targeted scenario, for example in drug discovery, it would be the set of molecules that have certain desirable characteristics (*e.g.*, affinity with a target protein is above a threshold and toxicity is below a threshold and synthesis cost is below a threshold). Let Y be the experimental outcome, x be the experiment specification, and V the variable of interest about which we seek to gain information. The information-theoretic utility for our experimental design would then be defined as

$$I(Y;V|x,\mathcal{D}) = \sum_{y,v} P(y,v|x,\mathcal{D}) \log \frac{P(y,v|x,\mathcal{D})}{P(y|x,\mathcal{D})P(v|x,\mathcal{D})} \tag{31}$$

$$I(Y;V|x,\mathcal{D}) = \sum_{y,v} P(y,v|x,\mathcal{D}) \log \frac{P(y|v,x,\mathcal{D})}{P(y|x,\mathcal{D})} \tag{32}$$

where $\mathcal{D}$ is our dataset of prior experimental results $\{(x, y)\}$ and any other constraint we want to exploit to condition the probabilities. With V typically being a much higher-dimensional object than Y, eqn (32) tends to be more practical numerically. To evaluate the expression in eqn (32) we can leverage the ideas of amortization and GFlowNets to estimate (a) the numerator and denominator probabilities $P(y|v,x,\mathcal{D}), P(y|x,\mathcal{D})$ (b) a sampler for the joint $P(y,v|x,\mathcal{D})$, and (c) an estimator $\hat{I}$ of the MI itself, as a function of x. In the case where the variable of interest are the parameters of some underlying process $v = \theta$ (as introduced in Section 3.2), we can follow the amortization approach outlined in Section 4.3.2 to estimate the posterior predictive $P(y|x,\mathcal{D})$ in the denominator. On the other hand, the likelihood in the numerator $P(y|\theta,x,\mathcal{D}) = P(y|\theta,x)$ is available in the case of explicit models, and can be approximated in the case of implicit models as discussed in Section 3.3. Next to learn a sampler for the joint $P(y,\theta|x,\mathcal{D})$ we first approximate samples from the posterior over parameters $Q(\theta|\mathcal{D})$ following Section 4.1.6. By combining the samples from the posterior $Q(\theta|\mathcal{D})$ and the likelihood $P(y|x,\theta)$ we can learn a sampler $Q(y,\theta|x,\mathcal{D}) = Q(\theta|\mathcal{D})P(y|x,\theta)$ to approximate the joint $P(y,\theta|x,\mathcal{D})$.

As for the estimator of MI itself, one possibility is to train a neural network $\hat{I}(x)$ to amortize the expected value over (θ, y) given $(x, \mathcal{D})$ using the samples from the joint $Q(y,\theta|x,\mathcal{D})$ and the estimators for the probabilities in the log-prob ratio from with a squared loss

$$\left(\hat{I}(x) - \log \frac{P(y|x,\theta)}{Q(y|x,\mathcal{D})}\right)^2 \tag{33}$$

where x is sampled from a dataset or a generative model of inputs, $\theta \sim Q(\theta|\mathcal{D})$ and $y \sim P(y|x,\theta)$. With enough capacity and training time, Q converges to P and $\hat{I}(x)$ converges to $I(Y;\theta|x,\mathcal{D})$. What is particularly interesting if x is in a high-dimensional space is that it generally won't be necessary to see more than one value of y and θ for each value of x in order to train $\hat{I}$, as usual in supervised learning. This can work if it is possible for the learning procedure for $\hat{I}$ to generalize from the (θ, x, y) triplets used to train it.

5.2 Additional open challenges

5.2.1 Modeling and causality. One open challenge on the modeling side is to leverage GFlowNets to model Bayesian posteriors beyond causal models.[128] take an initial step in this direction, using GFlowNets to model the posterior over dropout masks in a neural network. Moreover, in the context of causal models, many challenges remain to extend the work done by.[116] This includes (a) accommodating larger causal graphs efficiently (b) making it possible to handle unobserved causal variables by also learning how the raw inputs (*e.g.*, images) may be related to the causal variables (c) learning how experimental choices relate to interventions on the causal variable when this is not known perfectly *a priori*.

5.2.2 Experimental design. Our discussion has focused primarily on the case where we are interested in information gain about the parameter θ to drive knowledge acquisition in the experimental design loop. As we discussed in the previous section, it is possible to incorporate any random variable V (eqn

(31)) that we can learn to model and sample. An interesting case is one where the random variable *V* is an extremum, *e.g.*, the top molecular candidates, for some task, as in eqn (9). Furthermore, in many practical experimental settings, we have access to measurements of varying fidelities as the outcome of our experiments (*e.g.*, computer simulations with varying accuracy-computational cost trade-offs). Such a multi-fidelity setting is discussed above (Section. 3.4) but needs to be incorporated within the GFlowNet framework. Another important practical scenario also introduced in the same section is the existence of multiple objectives, with early work to incorporate that in the GFlowNet framework by Jain *et al.*[113]

Finally, as introduced in Section 3.5 these experimental design tools could be integrated within the causal discovery framework, by focusing the knowledge acquisition on the causal graph itself, an object of great value to scientists from an interpretability point of view. This could be achieved by using the graph itself as the target variable *V* (or a part of it), to drive the experimental design to accelerate the discovery of the causal structure.

Data availability

No new data are provided in this article.

Conflicts of interest

Yoshua Bengio is an advisor to Recursion and Dreamfold. Jason Hartford's postdoctoral position was partly funded by Recursion.

Acknowledgements

The authors thank DreamFold and Maksym Korablyov for important help, suggestions and feedback, as well as acknowledge funding from CIFAR, Recursion, IVADO, NSERC, and the Quebec government.

References

1 C. Adler, P. Wester, I. Bhatt, C. Huggel, G. E. Insarov, M. D. Morecroft, V. Muccione, and A. Prakash, *Cross-Chapter Paper 5: Mountains*, Cambridge University Press, Cambridge, UK and New York, USA, 2022, pp. 2273–2318, ISBN 9781009325844, DOI: 10.1017/9781009325844.022.2273.

2 T. A. Ban, The role of serendipity in drug discovery, *Dialogues Clin. Neurosci.*, 2022, **8**(3), 335–344.

3 R. S. Bohacek, C. McMartin and W. C. Guida, The art and practice of structure-based drug design: A molecular modeling perspective, *Med. Res. Rev.*, 1996, **16**(1), 3–50, DOI: 10.1002/(SICI)1098-1128(199601)16:1<3::AID-MED1>3.0.CO;2-6, https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291098-1128%28199601%2916%3A1%3C3%3A%3AAID-MED1%3E3.0.CO%3B2-6.

4 B. P. MacLeod, F. G. L. Parlane, T. D Morrissey, F. Häse, L. M. Roch, K. E. Dettelbach, R. Moreira, L. P. E. Yunker, M. B. Rooney, J. R. Deeth, *et al.*, Self-driving laboratory for accelerated discovery of thin-film materials, *Sci. Adv.*, 2020, **6**(20), eaaz8867.

5 J. G. H. Anthony, T. Stewart, K. M. Tolle, *et al.*, *The fourth paradigm: data-intensive scientific discovery*, Microsoft research Redmond, WA, 2009, vol. 1.

6 A. Agrawal and A. Choudhary, Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science, *Apl Materials*, 2016, **4**(5), 053208.

7 H. Stärk, D. Beaini, G. Corso, P. Tossou, C. Dallago, S. Günnemann, and L. Pietro, 3d infomax improves gnns for molecular property prediction, in *International Conference on Machine Learning*, PMLR, 2022, pp. 20479–20502.

8 E. G. Ryan, C. C. Drovandi, J. M. McGree and A. N. Pettitt, A review of modern computational algorithms for bayesian optimal design, *Int. Stat. Rev.*, 2016, **84**(1), 128–154.

9 C. Angermueller, D. Dohan, D. Belanger, R. Deshpande, K. Murphy, and L. Colwell, Model-based reinforcement learning for biological sequence design, in *International conference on learning representations*, 2019.

10 S. Kim, P. Y. Lu, C. Loh, J. Smith, J. Snoek, and M. Soljacic, *Deep learning for bayesian optimization of scientific problems with high-dimensional structure*, Transactions of Machine Learning Research, 2022.

11 E. Bengio, M. Jain, M. Korablyov, D. Precup, and Y. Bengio, Flow network based generative models for non-iterative diverse candidate generation, in *Advances in Neural Information Processing Systems*, ed. A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, 2021, https://openreview.net/forum?id=Arn2E4IRjEB.

12 Y. Bengio, T. Deleu, E. J. Hu, S. Lahlou, M. Tiwari, and E. Bengio, *Gflownet foundations*, 2021.

13 D. Silver, A. Huang, C. J. Maddison, A. Guez, S. Laurent, G. V. D. Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, Mastering the game of go with deep neural networks and tree search, *Nature*, 2016, **529**(7587), 484–489.

14 J. Jumper, R. Evans, P. Alexander, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, Ž. Augustin, P. Anna, *et al.*, Highly accurate protein structure prediction with alphafold, *Nature*, 2021, **596**(7873), 583–589.

15 A. Der Kiureghian and O. Ditlevsen, Aleatory or epistemic? does it matter?, *Structural Safety*, 2009, **31**(2), 105–112, DOI: 10.1016/j.strusafe.2008.06.020, ISSN 0167-4730.

16 S. J. Honrao, X. Yang, B. Radhakrishnan, S. Kuwata, H. Komatsu, A. Ohma, M. Sierhuis and J. W. Lawson, Discovery of novel li sse and anode coatings using interpretable machine learning and high-throughput multi-property screening, *Sci. Rep.*, 2021, **11**(1), 1–14.

17 B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal and Y. Bengio, Toward causal representation learning, *Proc. IEEE*, 2021, **109**(5), 612–634.

18 C. Andrieu, N. De Freitas, A. Doucet and M. I. Jordan, An introduction to mcmc for machine learning, *Mach. Learn.*, 2003, **50**(1), 5–43.

19 G. M. Torrie and J. P. Valleau, Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling, *J. Comput. Phys.*, 1977, **23**(2), 187–199.

20 D. J. Earl and M. W. Deem, Parallel tempering: Theory, applications, and new perspectives, *Phys. Chem. Chem. Phys.*, 2005, **7**(23), 3910–3916.

21 A. Beskos, F. J. Pinski, J. Marıa Sanz-Serna and A. M. Stuart, Hybrid monte carlo on hilbert spaces, *Stoch. Process. Their Appl.*, 2011, **121**(10), 2201–2230.

22 D. M. Blei, A. Kucukelbir and J. D. McAuliffe, Variational inference: A review for statisticians, *J. Am. Stat. Assoc.*, 2017, **112**(518), 859–877.

23 M. Tom, *et al.*, *Divergence measures and message passing*, Technical report, Microsoft Research, 2005.

24 R. E. Turner and M. Sahani, *Two problems with variational expectation maximisation for time series models*, Cambridge University Press, 2011, pp. 104–124, DOI: 10.1017/CBO9780511984679.006.

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

26 V. V. Fedorov, *Theory of optimal experiments*, Elsevier, 1972.

27 A. C. Atkinson and A. N. Donev, *Optimum experimental designs*, Clarendon Press, 1992, vol. 5.

28 S. Burr, Active learning, *Synth. Lect. Artif. Intell. Mach. Learn.*, 2012, **6**(1), 1–114.

29 K. Chaloner and I. Verdinelli, *Bayesian experimental design: A review*, Statistical Science, 1995, pp. 273–304.

30 D. V. Lindley, On a measure of the information provided by an experiment, *Ann. Math. Stat.*, 1956, **27**(4), 986–1005.

31 D. R. C. Jay I Myung and M. A. Pitt, A tutorial on adaptive design optimization, *J. Math. Psychol.*, 2013, **57**(3–4), 53–67.

32 T. Rainforth, R. Cornish, H. Yang, A. Warrington, and F. Wood, On nesting monte carlo estimators, in *International Conference on Machine Learning*, PMLR, 2018, pp. 4267–4276.

33 A. Foster, J. Martin, E. Bingham, H. Paul, Y. W. Teh, T. Rainforth and N. Goodman, Variational bayesian optimal experimental design, *Adv. Neural Inf. Process. Syst.*, 2019, **32**, https://papers.nips.cc/paper_files/paper/2019/file/d55cbf210f175f4a37916eafe6c04f0d-Bibtex.bib.

34 S. Kleinegesse and M. U. Gutmann, Efficient bayesian experimental design for implicit models, in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019, pp. 476–485.

35 S. Kleinegesse and M. U. Gutmann, Bayesian experimental design for implicit models by mutual information neural estimation, in *International Conference on Machine Learning*, PMLR, 2020, pp. 5316–5326.

36 F. Heinrich, P. A. Kienzle, D. P. Hoogerheide and M. Lösche, Information gain from isotopic contrast variation in neutron reflectometry on protein–membrane complex structures, *J. Appl. Crystallogr.*, 2020, **53**(3), 800–810.

37 O. Antony and J. McGree, Bayesian design of experiments for intractable likelihood models using coupled auxiliary models and multivariate emulation, *Bayesian Anal.*, 2020, **15**(1), 103–131.

38 C. C. Drovandi, J. M. McGree and A. N. Pettitt, Sequential monte carlo for bayesian sequentially designed experiments for discrete data, *Comput. Stat. Data Anal.*, 2013, **57**(1), 320–335.

39 A. Foster, J. Martin, M. O'Meara, Y. W. Teh, and T. Rainforth. A unified stochastic gradient approach to designing bayesian-optimal experiments, in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020, pp. 2959–2969.

40 A. Foster, D. R. Ivanova, I. Malik, and T. Rainforth. Deep adaptive design: Amortizing sequential bayesian experimental design, in *International Conference on Machine Learning*, PMLR, 2021, pp. 3384–3395.

41 D. R. Ivanova, A. Foster, S. Kleinegesse, M. U. Gutmann and T. Rainforth, Implicit deep adaptive design: policy-based experimental design without likelihoods, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 25785–25798.

42 T. Blau, E. V Bonilla, I. Chades, and A. Dezfouli. Optimizing sequential experimental design with deep reinforcement learning, in *International Conference on Machine Learning*, PMLR, 2022, pp. 2107–2128.

43 R. Horst, P. M. Pardalos, and N. V. Thoai, *Introduction to global optimization*, Springer Science & Business Media, 2000.

44 J. Mockus, V. Tiesis, and A. Zilinskas, The application of bayesian methods for seeking the extremum, *Towards global optimization*, vol. 2, 2, 1978, pp. 117–129.

45 D. R. Jones, M. Schonlau and W. J. Welch, Efficient global optimization of expensive black-box functions, *J. Glob. Optim.*, 1998, **13**(4), 455–492.

46 R. Garnett, *Bayesian Optimization*, Cambridge University Press, 2022.

47 J. Gonzalez, L. Joseph, D. C. James, and N. D. Lawrence, Bayesian optimization for synthetic gene design, *arXiv*, 2015, preprint, arXiv:1505.01627, DOI: 10.48550/arXiv.1505.01627.

48 R.-R. Griffiths and J. M. Hernández-Lobato, Constrained bayesian optimization for automatic chemical design using variational autoencoders, *Chem. Sci.*, 2020, **11**(2), 577–586.

49 H. B. Moss and R.-R. Griffiths, Gaussian process molecule property prediction with flowmo, *arXiv*, 2020, preprint, arXiv:2010.01118, DOI: 10.48550/arXiv.2010.01118.

50 B. J. Shields, J. Stevens, J. Li, M. Parasram, F. Damani, J. I. Martinez Alvarado, M. J. Jacob, R. P. Adams and A. G. Doyle, Bayesian reaction optimization as a tool for chemical synthesis, *Nature*, 2021, **590**(7844), 89–96.

51 C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning series*, MIT Press, 2005, ISBN 9780262182539, https://books.google.ca/books?id=GhoSngEACAAJ.

52 M. Balandat, B. Karrer, D. Jiang, S. Daulton, B. Letham, A. G. Wilson and E. Bakshy, Botorch: a framework for efficient monte-carlo bayesian optimization, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 21524–21538.

53 J. B. Mockus and L. J. Mockus, Bayesian approach to global optimization and application to multiobjective and constrained problems, *J. Optim. Theory Appl.*, 1991, **70**(1), 157–172.

54 N. Srinivas, A. Krause, K. Sham, and M. Seeger, Gaussian process optimization in the bandit setting: no regret and experimental design, in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 2010, pp. 1015–1022.

55 W. R. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, *Biometrika*, 1933, **25**(3–4), 285–294.

56 S. Vakili, H. Moss, A. Artemev, D. Vincent and V. Picheny, Scalable thompson sampling using sparse Gaussian process models, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 5631–5643.

57 P. Hennig and C. J. Schuler, Entropy search for information-efficient global optimization, *J. Mach. Learn. Res.*, 2012, **13**(6), 1809–1837.

58 J. M. Hernández-Lobato, M. W. Hoffman and Z. Ghahramani, Predictive entropy search for efficient global optimization of black-box functions, *Adv. Neural Inf. Process. Syst.*, 2014, **27**, **https://papers.nips.cc/paper_files/paper/2014/file/069d3bb002acd8d7dd095917f9efe4cb-Bibtex.bib**.

59 M. W. Hoffman and Z. Ghahramani, Output-space predictive entropy search for flexible global optimization, in *NIPS workshop on Bayesian Optimization*, 2015, pp. 1–5.

60 Z. Wang and S. Jegelka, Max-value entropy search for efficient bayesian optimization, in *International Conference on Machine Learning*, PMLR, 2017, pp. 3627–3635.

61 H. B. Moss, D. S. Leslie, J. Gonzalez and R. Paul, Gibbon: General-purpose information-based bayesian optimisation, *J. Mach. Learn. Res.*, 2021, **22**(235), 1–49.

62 J. González, Z. Dai, P. Hennig, and N. Lawrence, Batch bayesian optimization *via* local penalization, in *Artificial intelligence and statistics*, PMLR, 2016, pp. 648–657.

63 K. Kandasamy, A. Krishnamurthy, J. Schneider, and B. Póczos, Parallelised bayesian optimisation *via* thompson sampling, in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2018, pp. 133–142.

64 V. Picheny, D. Ginsbourger, and Y. Richet. *Noisy expected improvement and on-line computation time allocation for the optimization of simulators with tunable fidelity*, 2010.

65 K. Kandasamy, G. Dasarathy, J. Schneider, and B. Póczos, Multi-fidelity bayesian optimisation with continuous approximations, in *International Conference on Machine Learning*, PMLR, 2017, pp. 1799–1808.

66 S. Takeno, H. Fukuoka, Y. Tsukada, T. Koyama, M. Shiga, I. Takeuchi, and M. Karasuyama, Multi-fidelity bayesian optimization with max-value entropy search and its parallelization, in *International Conference on Machine Learning*, PMLR, 2020, pp. 9334–9345.

67 S. Belakaria, A. Deshwal and J. Rao Doppa, Max-value entropy search for multi-objective bayesian optimization, *Adv. Neural Inf. Process. Syst.*, 2019, **32**, **https://proceedings.neurips.cc/paper_files/paper/2019/file/82edc5c9e21035674d481640448049f3-Bibtex.bib**.

68 S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, Multi-objective bayesian optimization over high-dimensional search spaces, in *Uncertainty in Artificial Intelligence*, PMLR, 2022, pp. 507–517.

69 M. A. Ziatdinov, Y. Liu, A. N. Morozovska, E. A. Eliseev, X. Zhang, I. Takeuchi and S. V. Kalinin, Hypothesis learning in automated experiment: application to combinatorial materials libraries, *Adv. Mater.*, 2022, **34**(20), 2201345.

70 M. Austin, M. Frontzek, A. T. Savici, M. Doucet, E. E. Rodriguez, K. Meuse, J. Opsahl-Ong, D. Samarov, I. Takeuchi, W. Ratcliff, *et al.*, On-the-fly autonomous control of neutron diffraction *via* physics-informed bayesian active learning, *Appl. Phys. Rev.*, 2022, **9**(2), 021408.

71 H. Moss, D. Leslie, D. Beck, J. Gonzalez and R. Paul, Boss: Bayesian optimization over string spaces, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 15476–15486.

72 S. Kevin, Y. Rubanova, D. Dohan, and K. Murphy, Amortized bayesian optimization over discrete spaces, in *Conference on Uncertainty in Artificial Intelligence*, PMLR, 2020, pp. 769–778.

73 J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, 1988.

74 P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, Prediction, and Search*, MIT press, 2000.

75 D. Maxwell Chickering, Optimal Structure Identification With Greedy Search, *J. Mach. Learn. Res.*, 2002, 507–554.

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

77 N. R. Ke, O. Bilaniuk, A. Goyal, S. Bauer, H. Larochelle, B. Schölkopf, M. C. Mozer, C. Pal, and Y. Bengio, Learning neural causal models from unknown interventions, *arXiv*, 2019, arXiv:1910.01075, preprint, DOI: **10.48550/arXiv.1910.01075**.

78 P. Brouillard, S. Lachapelle, A. Lacoste, S. Lacoste-Julien and D. Alexandre, Differentiable Causal Discovery from Interventional Data, *Adv. Neural Inf. Process. Syst.*, 2020, 21865–21877.

79 D. Madigan, J. York and D. Allard, Bayesian Graphical Models for Discrete Data, *Int. Stat. Rev.*, 1995, **63**(2), 215–232.

80 N. Friedman and D. Koller, Being Bayesian About Network Structure. A Bayesian Approach to Structure Discovery in Bayesian Networks, *Mach. Learn.*, 2003, **50**, 95–125.

81 P. Giudici and R. Castelo, Improving Markov chain Monte Carlo model search for data mining, *Mach. Learn.*, 2003, **50**, 127–158.

82 T. Niinimäki, P. Parviainen and M. Koivisto, Structure Discovery in Bayesian Networks by Sampling Partial Orders, *J. Mach. Learn. Res.*, 2016, 1–47.

83 J. Viinikka, A. Hyttinen, J. Pensar and M. Koivisto, Towards Scalable Bayesian Learning of Causal DAGs, *Adv. Neural Inf. Process. Syst.*, 2020, 6584–6594.

84 N. Friedman, M. Goldszmidt, and W. Abraham, Data Analysis with Bayesian Networks: A Bootstrap Approach, *Proceedings of the Fifteenth conference on Uncertainty in Artificial Intelligence*, 1999.

85 R. Agrawal, C. Squires, K. Yang, K. Shanmugam, and C. Uhler, Abcd-strategy: Budgeted experimental design for targeted causal structure discovery, in *The 22nd International Conference on Artificial Intelligence and Statistics*, PMLR, 2019.

86 Y. Annadani, J. Rothfuss, A. Lacoste, N. Scherrer, A. Goyal, Y. Bengio, and S. Bauer, Variational Causal Networks: Approximate Bayesian Inference over Causal Structures, *arXiv*, 2021, preprint, DOI: 10.48550/arXiv.2106.07635.

87 C. Cundy, A. Grover and S. Ermon, BCD Nets: Scalable Variational Approaches for Bayesian Causal Discovery, *Adv. Neural Inf. Process. Syst.*, 2021, 7095–7110.

88 B. Wang, M. R. Wicker, and M. Kwiatkowska, Tractable Uncertainty for Structure Learning, *International Conference on Machine Learning*, 2022.

89 L. Lorch, J. Rothfuss, B. Schölkopf and A. Krause, DiBS: Differentiable Bayesian Structure Learning, *Adv. Neural Inf. Process. Syst.*, 2021, 24111–24123.

90 L. Lorch, S. Scott, J. Rothfuss, A. Krause and B. Schölkopf, Amortized Inference for Causal Structure Learning, *Adv. Neural Inf. Process. Syst.*, 2022, 13104–13118.

91 W. BUNTINE, Theory refinement on bayesian networks, in *Proc. 7th Conf. Uncertainty in Artificial Intelligence*, 1991, vol. 1991, pp. 52–60.

92 S. Tong and D. Koller. Active learning for structure in Bayesian networks. *International Joint Conference on Artificial Intelligence*, 2001.

93 K. Murphy. *Active Learning of Causal Bayes Net Structure*, 2001.

94 N. Scherrer, O. Bilaniuk, Y. Annadani, A. Goyal, P. Schwab, B. Schölkopf, M. C. Mozer, Y. Bengio, S. Bauer, and N. R. Ke, Learning Neural Causal Models with Active Interventions, *arXiv*, 2021, arXiv:2109.02429, preprint, DOI: 10.48550/2109.02429.

95 P. Tigas, Y. Annadani, A. Jesson, B. Schölkopf, Y. Gal, and S. Bauer, *Interventions, Where and How? Experimental Design for Causal Models at Scale*, Neural Information Processing Systems, 2022.

96 C. Toth, L. Lorch, C. Knoll, A. Krause, F. Pernkopf, R. Peharz, and J. von Kügelgen, *Active Bayesian Causal Inference*, Neural Information Processing Systems, 2022.

97 C. M. Bishop, *et al.*, *Neural networks for pattern recognition*, Oxford university press, 1995.

98 N. Malkin, M. Jain, E. Bengio, C. Sun and Y. Bengio, Trajectory balance: Improved credit assignment in gflownets, *Advances in Neural Information Processing Systems*, 2022, pp. 5955–5967.

99 K. Madan, J. Rector-Brooks, M. Korablyov, E. Bengio, M. Jain, A. Nica, T. Bosc, Y. Bengio, and N. Malkin, Learning gflownets from partial episodes for improved convergence and stability, *arXiv*, 2022, arXiv:2209.12782, preprint, DOI: 10.48550/arXiv.2209.12782.

100 Y. Xie, C. Shi, H. Zhou, Y. Yang, W. Zhang, Y. Yu, and L. Li, {MARS}: Markov molecular sampling for multi-objective drug discovery, in *International Conference on Learning Representations*, 2021, https://openreview.net/forum?id=kHSu4ebxFXY.

101 M. Jain, E. Bengio, A. Hernandez-Garcia, J. Rector-Brooks, B. F. P. Dossou, C. A. Ekbote, J. Fu, T. Zhang, M. Kilgour, D. Zhang, *et al.*, Biological sequence design with gflownets, in *International Conference on Machine Learning*, PMLR, 2022, pp. 9786–9801.

102 P. Diederik, Kingma and Max Welling. Auto-encoding variational Bayes, *International Conference on Learning Representations*, ICLR, 2014.

103 D. Jimenez Rezende, S. Mohamed, and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, *International Conference on Machine Learning (ICML)*, 2014.

104 I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C Courville, and Y. Bengio, in *Generative adversarial nets*, NIPS, 2014.

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

106 L. Pan, N. Malkin, D. Zhang, and Y. Bengio, *Better training of gflownets with local credit and incomplete trajectories*, 2023, https://arxiv.org/abs/2302.01687.

107 S. Zhang, Y. Liu, and L. Xie, Molecular mechanics-driven graph neural network with multiplex graph for molecular structures, *arXiv*, 2020, arXiv:2011.07457, preprint, DOI: 10.48550/arXiv.2011.07457.

108 N. Chiamvimonvat, C.-M. Ho, H.-J. Tsai and B. D. Hammock, The soluble epoxide hydrolase as a pharmaceutical target for hypertension, *J. Cardiovasc. Pharmacol.*, 2007, **50**(3), 225–237.

109 D. I. John and B. D. Hammock, Soluble epoxide hydrolase as a therapeutic target for cardiovascular diseases, *Nat. Rev. Drug Discovery*, 2009, **8**(10), 794–805.

110 O. Trott and A. J. Olson, Autodock vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading, *J. Comput. Chem.*, 2010, **31**(2), 455–461.

111 J. J. Irwin and B. K. Shoichet, Zinc- a free database of commercially available compounds for virtual screening, *J. Chem. Inf. Model.*, 2005, **45**(1), 177–182.

112 A. C. Nica, M. Jain, E. Bengio, C.-H. Liu, M. Korablyov, M. M. Bronstein, and Y. Bengio, Evaluating generalization in GFlownets for molecule design, in *ICLR2022 Machine Learning for Drug Discovery*, 2022, https://openreview.net/forum?id=JFSaHKNZ35b.

113 M. Jain, S. Chandra Raparthy, A. Hernandez-Garcia, J. Rector-Brooks, Y. Bengio, S. Miret, and E. Bengio,

Multi-objective gflownets, *arXiv*, 2022, arXiv:2210.12765, preprint, DOI: **10.48550/arXiv.2210.12765**.

114 M. Ehrgott, *Multicriteria optimization*, Springer Science & Business Media, 2005, vol. 491.

115 O'Neill UK government study, *Antimicrobial resistance: Tackling a crisis for the health and wealth of nations*, 2014.

116 T. Deleu, A. Góis, C. C. Emezue, M. Rankawat, S. Lacoste-Julien, S. Bauer, and Y. Bengio, Bayesian structure learning with generative flow networks, in *The 38th Conference on Uncertainty in Artificial Intelligence*, 2022.

117 D. Heckerman, D. Geiger, and D. M. Chickering, *Learning bayesian networks: The combination of knowledge and statistical data*, Machine learning, 1995.

118 M. Nishikawa-Toomey, T. Deleu, J. Subramanian, Y. Bengio and L. Charlin, Bayesian learning of causal structure and mechanisms with gflownets and variational bayes, *arXiv*, 2022, arXiv:2211.02763, preprint, DOI: **10.48550/arXiv.2211.02763**.

119 J. W. Freimer, O. Shaked, S. Naqvi, N. Sinnott-Armstrong, A. Kathiria, C. M. Garrido, A. F. Chen, T. Jessica, W. J. G. Cortez, J. K. Pritchard, *et al.*, Systematic discovery and perturbation of regulatory genes in human t cells reveals the architecture of immune networks, *Nat. Genet.*, 2022, **54**(8), 1133–1144.

120 A. Hyttinen, F. Eberhardt and P. O. Hoyer, Learning linear cyclic causal models with latent variables, *J. Mach. Learn. Res.*, 2012, **13**(1), 3387–3439.

121 L. Lorch, S. Scott, J. Rothfuss, A. Krause and B. Schölkopf, Amortized inference for causal structure learning, *arXiv*, 2022, arXiv:2205.12934, preprint, DOI: **10.48550/arXiv.2205.12934**.

122 M. G. Sethuraman, R. Lopez, R. Mohan, F. Fekri, T. Biancalani and J.-C. Hütter, Nodags-flow: Nonlinear cyclic causal structure learning, *arXiv*, 2023, arXiv:2301.01849, preprint, DOI: **10.48550/arXiv.2301.01849**.

123 D. Madigan, A. E. Raftery, C. Volinsky, and J. Hoeting, Bayesian Model Averaging, in *Proceedings of the AAAI Workshop on Integrating Multiple Learned Models*, 1996.

124 A. H. Jennifer, D. Madigan, A. E. Raftery, and C. T. Volinsky, *Bayesian model averaging: a tutorial with comments by M. Clyde, David Draper and EI George, and a rejoinder by the authors*, Statistical science, 1999.

125 M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, S. Murray, Y. W. Teh, D. Rezende, and S. M. A. Eslami, Conditional neural processes, in *International Conference on Machine Learning*, PMLR, 2018, pp. 1704–1713.

126 D. Eaton and K. Murphy, *Bayesian structure learning using dynamic programming and MCMC*, Uncertainty in Artificial Intelligence, 2007.

127 S. Zheng, D. Hayden, J. Pacheco and J. W. Fisher III, Sequential bayesian experimental design with variable cost structure, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 4127–4137.

128 D. Liu, M. Jain, B. Dossou, Q. Shen, S. Lahlou, A. Goyal, N. Malkin, C. Emezue, D. Zhang, N. Hassen, *et al.*, Gflowout: Dropout with generative flow networks, arXiv, 2022, preprint, DOI: **10.48550/arXiv.2210.12928**.