# Benchmarking Bayesian Causal Discovery Methods for Downstream Treatment Effect Estimation

**Chris Chinenye Emezue** [1 2 3] **Alexandre Drouin** [3 4] **Tristan Deleu** [3 5] **Stefan Bauer** [2 6] **Yoshua Bengio** [3 5 7 8]

## Abstract

The practical utility of causality in decision-making is widely recognized, with causal discovery and inference being inherently intertwined. Nevertheless, a notable gap exists in the evaluation of causal discovery methods, where insufficient emphasis is placed on downstream inference. To address this gap, we evaluate six established baseline causal discovery methods and a newly proposed method based on GFlowNets, on the downstream task of treatment effect estimation. Through the implementation of a robust evaluation procedure, we offer valuable insights into the efficacy of these causal discovery methods for treatment effect estimation, considering both synthetic and real-world scenarios, as well as low-data scenarios. Furthermore, the results of our study demonstrate that GFlowNets possess the capability to effectively capture a wide range of useful and diverse ATE modes.

## 1. Introduction

Causal inference has a wide variety of real-world applications in domains such as healthcare (Tu et al., 2019; Huang et al., 2019; Bica et al., 2020; Geffner et al., 2022) , marketing (Zhang & Chan, 2006; Battocchi et al., 2021; Sanchez et al., 2022), political science, and online advertising (Waisman et al., 2019; Maiya, 2021). Treatment effect estimation, the process of estimating the effect or impact of a treatment on an outcome in the presence of other covariates as potential confounders (and mediators), is a fundamental problem in causal inference that has received widespread interest for decades (Chu et al., 2023).

[1]Work done as a visiting research student at Mila [2]Technical University of Munich, Munich, Germany [3]Mila - Quebec AI Institute, Montreal, Canada [4]ServiceNow Research, Montreal, Canada [5]Université de Montréal, Montreal, Canada [6]Helmholtz AI [7]CIFAR AI Chair [8]CIFAR Senior Fellow. Correspondence to: Chris Chinenye Emezue <chris.emezue@gmail.com>.

*Accepted to ICML 2023 Workshop on Structured Probabilistic Inference Generative Modeling*

The existing powerful methods for treatment effect estimation from data require a complete (or partial) a priori knowledge of the causal graph (Pearl, 2009a; Pearl & Mackenzie, 2018). When the graph is unknown, this requires solving a problem of *causal structure learning*, also known as *causal discovery*. In structure learning, one learns a graph (typically characterized by a directed acyclic graph or DAG for short) that best describes the dependence structure in a given data set (Drton & Maathuis, 2017; Heinze-Deml et al., 2017; Glymour et al., 2019). In this approach, structure learning is required to learn a causal graph, which can then be applied to *infer* the influence of treatments on the outcomes of interest (Pearl, 2009a; Naser, 2022). It should be noted that the actual causal graph can only be inferred up to its Markov Equivalence class (MEC), and the available data does not offer any means of further differentiation (Maathuis et al., 2008; Pearl, 2009b). Learning a single graph has been shown to lead to poor predictions in a downstream causal inference task (Madigan et al., 1995a; Cundy et al., 2021; Tigas et al., 2022).

Instead of learning a single causal graph, the problem of structure learning can be tackled from a Bayesian perspective where we learn a posterior over the causal graphs. This has the unique advantage of accounting for epistemic uncertainty over the causal graphs in the MEC, thereby leading to a more enriching predictive performance in a downstream causal inference task. However, learning such a posterior over the causal graphs is plagued by challenges. One major issue is the combinatorially large sample space of causal graphs. The second major challenge is related to MCMC mode-mixing (Jasra et al., 2005; Bengio et al., 2012): the mode-mixing problem occurs when the chances of going from one mode to a neighboring one may become exponentially small and require exponentially long chains, if the modes are separated by a long sequence of low-probability configurations. Therefore by using MCMC, there is an important set of distributions for which finite chains are unlikely to provide enough diversity of the modes of the distribution (Bengio et al., 2021b). While there are a number of existing causal discovery methods (both Bayesian and non-Bayesian), our benchmark study centers on DAG-GFlowNet (Deleu et al., 2022), which is a unique method that leverages a novel class of probabilistic models called *Generative*

*Flow Networks* (Bengio et al., 2021a;b) to approximate the posterior distribution over causal graphs.

Although causal inference is an inherent downstream application of causal discovery, most causal discovery evaluation methods are not aligned with causal inference because these two fields are typically studied independently (Geffner et al., 2022). For example, many causal discovery evaluation methods use the structural hamming distance (SHD) which compares the learned causal DAG (or the samples from the posterior distribution of DAGs in Bayesian structure learning) to the true DAG of the data generating process. Measuring the proximity of the learned DAGs, however, does not reveal much about their actual performance in treatment effect estimation given a treatment and outcome variable of interest, which is a predominantly downstream evaluation.

In this work, we set out to benchmark causal discovery methods for the downstream task of treatment effect estimation, specifically the average treatment effect. As an extension to the DAG-GFlowNet, we offer insights on the application of GFlowNets to average treatment effect estimation, by comparing it with six other baseline methods for causal discovery.

## 2. Background

We provide a detailed background, in Section B of the Appendix, on some of the key concepts used in this paper: Bayesian network, interventional distribution, Bayesian causal discovery, average treatment effect and our structure learning baselines.

The structure learning baselines employed in our study follow those utilized by Deleu et al. (2022). In addition to DAG-GFlowNet (Deleu et al., 2022), we leveraged six baseline causal discovery algorithms: PC (Spirtes et al., 2001), GES (Chickering, 2003), MC3 (Madigan et al., 1995b), BCDNets (Cundy et al., 2021), Gadget (Viinikka et al., 2020), and DiBS (Lorch et al., 2021). Due to space restrictions, we move our explanation of the causal discovery methods to Section B.3 in the Appendix.

## 3. Experimental Setup

Figure 1 provides an illustrative overview of our experimental pipeline. The initial step involves Bayesian causal discovery, where, as discussed in Section B.1, the objective is to learn a posterior distribution of the directed acyclic graphs (DAGs) that provide the most plausible explanations for the training dataset. The subsequent stage involves the estimation of the average treatment effect (ATE). Here, the ATE for each DAG in the posterior is estimated for every non-matching variable pair. In addition, the DAGs within the Markov equivalence class (MEC) of the true graph are enumerated and used to calculate the ATE estimates for each of them. The evaluation process, in stage 3, then involves a comparison of the average treatment effect (ATE) distributions between the true graph Markov equivalence class (MEC) and the learned posterior distribution of DAGs.

For our experiments on synthetic data, we worked with 6 baselines in total and 25 seeds for each baseline. Each seed corresponds to a causal discovery experiment with a randomly sampled truth graph and observational data.

### 3.1. Causal discovery experiments

Following Deleu et al. (2022), we performed causal discovery experiments on synthetic and real-world scenarios. For PC and GES we implement bootstrapping to achieve DAG posterior samples.

**Analysis on synthetic data:** Following Deleu et al. (2022), we performed an experimental analysis on synthetic graphs and simulated data. We sample synthetic data from linear Gaussian Bayesian networks with randomly generated structures. We experimented with Bayesian networks of two sizes: $d = 20$ and $d = 100$. A small sample size of 20 was specifically chosen to evaluate the capabilities of the causal discovery algorithms in a low-data regime. The ground-truth graphs are sampled according to an Erdos-Rényi model.

**Analysis on flow cytometry data:** DAG-GFlowNet was evaluated against the baselines on real-world flow cytometry data (Sachs et al., 2005) to learn protein signaling pathways. The data consists of continuous measurements of $d = 11$ phosphoproteins in individual T-cells. They used the first $n = 853$ observations and the DAG, inferred by Sachs et al. (2005) and containing 11 nodes and 17 edges, as the dataset and ground-truth graph respectively for their causal discovery experiments. We continued with this direction in our experimental analysis and our goal was to show the downstream performance of DAG-GFlowNet on average treatment effect of the phosphoproteins in the protein signaling pathways.

### 3.2. ATE experiments

For our ATE experiments, we utilized all non-matching variable pair combinations: the rationale behind this was to thoroughly explore the possible treatment effects across various combinations. Therefore given $d$ random variables $\{X_1, ..., X_d\}$, we performed ATE evaluations on $d^2 - d$ variable pairs. To achieve this in practice, we leveraged the DoWhy package (Sharma et al., 2019; Sharma & Kiciman, 2020), which facilitated the implementation of the do-calculus algorithm. To ensure consistency and clarity in our results, we set the treatment values at 1.0 and 0.0 for

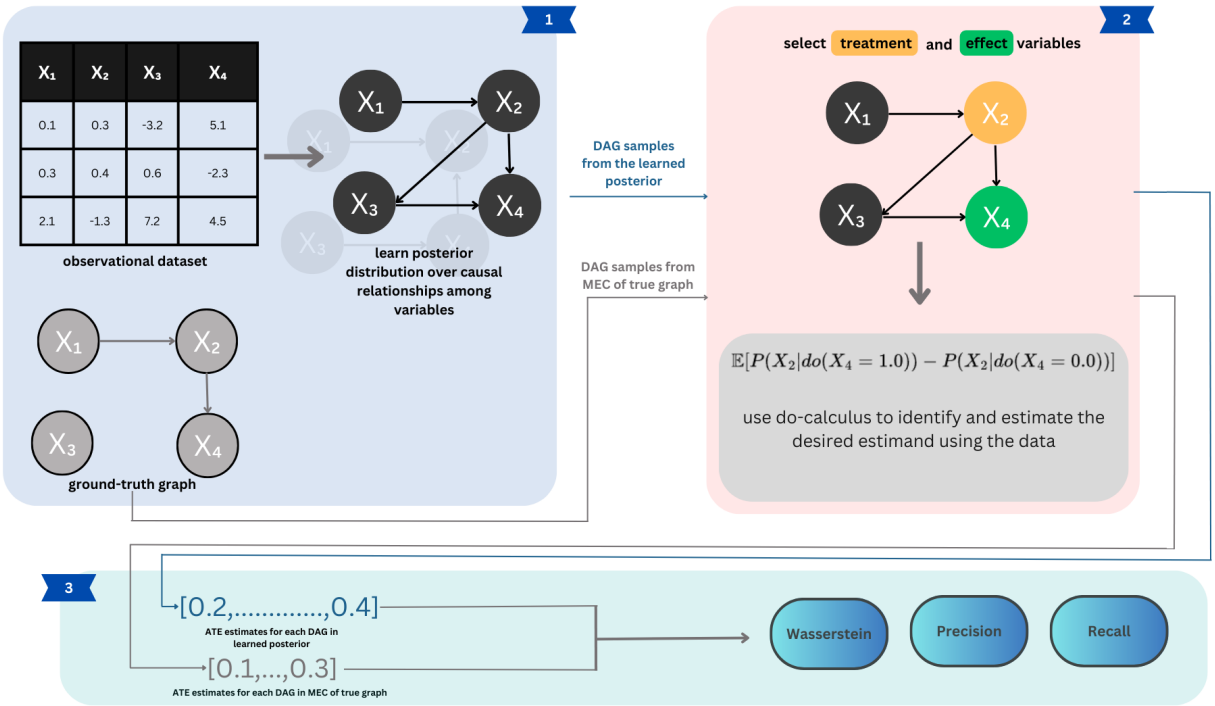


*Figure 1.* Illustration of our experimental pipeline using four random variables. We start with (Bayesian) causal discovery which involves learning a posterior over DAGs that best explain the training dataset. The next step is average treatment effect estimation (ATE). For each pair of non-matching variable pairs, the second stage involves estimating its ATE for each DAG in the posterior. Furthermore, from the true graph, we enumerate the DAGs in its Markov equivalence class (MEC) and find the ATE estimates for all of them. This allows us to perform our evaluation which involves comparing the ATE distributions between the true graph MEC and learned posterior distribution of DAGs. For clarity, this illustration has been given using only one baseline and one seed. For our experiments, we worked with 6 baselines and 25 seeds for each baseline.

all our experiments. The choice of values 1.0 and 0.0 does not relate to the existence or absence of a treatment, as is commonly used in most causal inference literature.

Performing such a robust experiment involved a huge computation load. For example, for our baselines, each with 25 random seeds, each consisting of 1000 sampled DAGs, we had to do $d * (d-1) * 1000 * 25 * 6$ ATE estimations. For the synthetic graph with 20 nodes, this leads to 57M estimations. In order to optimize the computational efficiency of our experiments, we implemented parallelism techniques. The GNU parallel computing tool (Tange, 2011) enabled us to distribute the computational workload across multiple processors or cores, thereby significantly reducing the overall computation time.

### 3.3. Evaluation Framework

Our evaluation methodology goes beyond single-point ATE estimation, which is employed in standard causal inference benchmarking, by performing ATE evaluations based on posterior samples. This approach aims to provide a more comprehensive assessment of the quality of the learned posterior average treatment effect (ATE). Specifically, our evaluation pipeline involves the following metrics:

**Wasserstein distance (WD):** To obtain a quantitative measure of the similarity between the true ATE sample-based distrbution and that of the learned ATE, we calculate and report their Wasserstein distance (Ramdas et al., 2017) using their samples[1].

**Precision and Recall:** We compute the precision and recall of the modes present in the learned ATE distribution and compare them to the modes in the true ATE distribution. In order to calculate the precision and recall, we first identify the unique modes for each of the true, $A_T$ and learned $A_\prime$ ATE samples. Then based on these set of modes, we calculate the true positive (modes from $A_T$ that are found in $A_\prime$) , false negative (modes from $A_T$ that are missed in $A_\prime$), and false positive (modes from $A_\prime$ that are not in $A_T$). Note that the lists $A_T$ and $A_\prime$ have been regrouped prior to running the evaluation (see Section 3.4).

[1]We utilize the Python implementation available here.

### 3.4. Additional settings

**Enumerating the MEC of the true graph:** In order to achieve our evaluation using our strategy (see Section 3.3), it is necessary to not work with just one true graph. For a given ground-truth graph, we enumerate all the DAGs in its Markov equivalence class (MEC).

**Regrouping ATE values:** The estimation of average treatment effects (ATE) through regression analysis is susceptible to generating estimates that may exhibit slight variations within numerical precision (e.g., 1.000000001 and 1). As our precision and recall metrics essentially perform 'hard matches" on floating point values, it becomes crucial to consider the influence of numerical precision. In order to accomplish this objective, we group ATE values that are numerically close. More details are in Appendix D.

## 4. Results & Discussion

The results presented in Table 1 illustrate the Wasserstein distance (WD), precision, and recall metrics of all baseline methods in terms of their learned ATE samples. Upon examining the Wasserstein distance, PC achieves the lowest Wasserstein distance, while GES attains the highest.

*Table 1.* ATE Evaluation on synthetic data (20 variables, 20 samples). We chose a small sample size of 20 in order to evaluate the capabilities of the causal discovery algorithms in a low-data regime. We report the Wasserstein distance (WD), precision and recall of all baselines. Average and variance is taken over all the $20 * 19$ variable combination pairs and over the 25 seeds for each baseline.

| | WD ↓ | PRECISION ↑ | RECALL ↑ |
|---|---|---|---|
| BCDNETS | 0.273 ±0.797 | **0.80 ±0.39** | 0.75 ±0.41 |
| BOOTSTRAP-GES | 0.328 ±0.615 | 0.01 ±0.01 | **0.98 ±0.11** |
| BOOTSTRAP-PC | **0.256 ±0.787** | 0.52 ±0.33 | 0.79 ±0.39 |
| DIBS | 0.263 ±0.770 | 0.49 ±0.41 | 0.80 ±0.37 |
| GADGET | 0.304 ±0.709 | 0.01 ±0.01 | 0.94 ±0.21 |
| MC3 | 0.297 ±0.650 | 0.01 ±0.04 | 0.94 ±0.20 |
| DAG-GFLOWNETS | 0.325 ±0.610 | 0.01 ±0.01 | 0.97 ±0.15 |

When focusing on precision, we observe that apart from BCDNets, all the methods seem to be performing very poorly. However all the methods attain relatively high recall scores, with the highest achieved by GES and closely followed by DAG-GFlowNet. This high recall indicates the ability of the methods to capture diverse modes within their ATE distribution.

The WD, precision, and recall for the synthetic data experiments with 100 samples are presented in Table 2. Given an increased number of observational samples compared to the previous table, it is anticipated that the task of causal discovery will be simpler. This is evidenced in the lower WD scores compared to Table 1. In a manner similar to the scenario involving 20 samples, it is observed that the methods, with the exception of BCDNets, exhibit a considerably low precision score, while concurrently displaying high recall values.

*Table 2.* ATE Evaluation on synthetic data (20 variables, 100 samples). We report the Wasserstein distance (WD), precision and recall of all baselines. Average and standard deviation are taken over all the $20 * 19$ variable combination pairs and over the 25 seeds for each baseline.

| | WD ↓ | PRECISION ↑ | RECALL ↑ |
|---|---|---|---|
| BCDNETS | **0.100 ±0.293** | **0.79 ±0.38** | 0.79 ±0.37 |
| BOOTSTRAP-GES | 0.165 ±0.411 | 0.01 ±0.03 | **0.99 ±0.09** |
| BOOTSTRAP-PC | 0.178 ±0.537 | 0.50 ±0.35 | 0.82 ±0.36 |
| DIBS | 0.187 ±0.489 | 0.05 ±0.12 | 0.97 ±0.13 |
| GADGET | 0.244 ±0.528 | 0.02 ±0.07 | 0.91 ±0.26 |
| MC3 | 0.177 ±0.450 | 0.06 ±0.20 | 0.93 ±0.23 |
| DAG-GFLOWNETS | 0.201 ±0.484 | 0.03 ±0.13 | 0.95 ±0.20 |

*Table 3.* ATE Evaluation on Sachs dataset (Sachs et al., 2005). We report the Wasserstein distance (WD), precision and recall of all baselines. Average and standard deviation are taken over all the $11 * 10$ variable combination pairs.

| METHOD | WD ↓ | PRECISION ↑ | RECALL ↑ |
|---|---|---|---|
| BCDNETS | 0.040 ±0.109 | **0.97 ±0.15** | 0.65 ±0.36 |
| BOOTSTRAP-GES | **0.037 ±0.099** | 0.13 ±0.13 | 0.96 ±0.09 |
| BOOTSTRAP-PC | 0.038 ±0.103 | 0.83 ±0.24 | 0.82 ±0.27 |
| DIBS | 0.037 ±0.101 | 0.17 ±0.14 | 0.94 ±0.13 |
| GADGET | 0.038 ±0.103 | 0.42 ±0.26 | 0.92 ±0.14 |
| MC3 | 0.041 ±0.111 | 0.46 ±0.33 | 0.86 ±0.24 |
| DAG-GFLOWNETS | 0.039 ±0.107 | 0.14 ±0.19 | **0.98 ±0.08** |

Table 3 presents the evaluation results of the analysis on flow cytometry using the Sachs dataset. Overall, all methods demonstrate comparable performance in terms of the Wasserstein distance: the range of the WD is 0.004, unlike in Table 1 which is 0.072 or Table 2 which is 0.144. When considering precision, BCDNets and PC outperform DAG-GFlowNet, which exhibits lower performance. Notably, DAG-GFlowNet achieves the highest recall, indicating its ability to learn samples from diverse modes within the true ATE distribution.

### 4.1. Filtering Low-Probability Modes

In all our evaluations (Tables 1, 2, 3), we witness a trend of DAG-GFlowNet and other methods exhibiting very low precision scores.

In Figure 3 we observe that DAG-GFlowNet (and other baselines like GES, DiBs) tends to learn new modes, but those modes have a very low probability in the estimated distribution. In our current evaluation framework however,

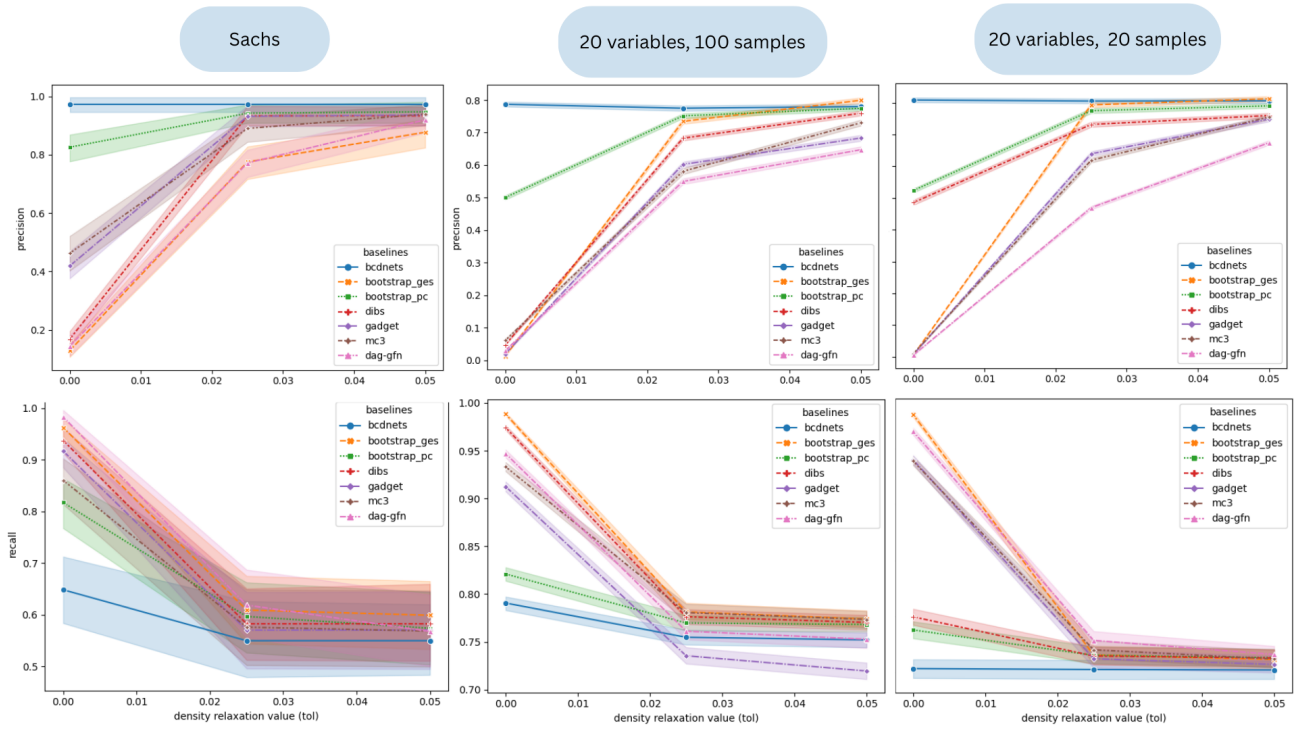


*Figure 2.* Lineplot of the average precision (top) and recall (bottom) for each of the methods. The lineplots are visualized for each dataset used in our experiment, denoted by the columns.

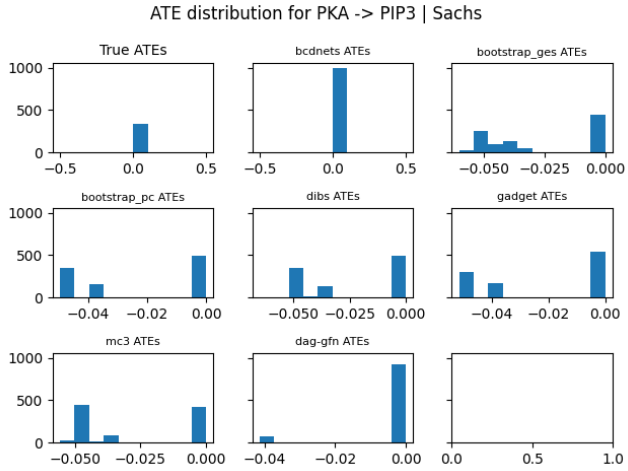


*Figure 3.* Comparison of ATE distribution of the methods using flow cytometry data. We focus on the treatment variable $PKA$ and the outcome variable $PIP3$ from the Sachs dataset. We see that while DAG-GFlowNet correctly captures all modes of the true distribution (just like BCDNets), it additionally learns values at 0.04 albeit with a very low probability mass.

we include all values in the list that have non-zero densities, which leads to unfair penalization of methods that exhibit multimodal diversity. Consequently, these methods receive disproportionately low precision values. However, when we apply a filtering approach that removes the low-probability modes before calculating the metrics, a more insightful narrative emerges for these methods, as shown in Figure 2. In particular, we notice a significant increase in precision for all the methods that initially exhibited very low precision values (in Tables 1, 2, and 3), when we apply a density relaxation tolerance of 0.05 (i.e for any list of ATEs, we only consider ATE values that have a mass of at least 0.05). This trend is consistent across all the experimental settings (100 samples, 20 samples, Sachs dataset).

## 5. Conclusion

In conclusion, the practical importance of causality in decision-making is widely acknowledged, and the interplay between causal discovery and inference is evident. In order to bridge the gap in the evaluation of causal discovery methods, where limited attention is given to downstream inference tasks, we conducted a comprehensive evaluation that assessed six established baseline causal discovery methods alongside a novel approach utilizing GFlowNets. By implementing a robust evaluation procedure, we provided valuable insights into their effectiveness for downstream treatment effect estimation.

## References


Ahmed, O., Trauble, F., Goyal, A., Neitz, A., Wuthrich, M., Bengio, Y., Schölkopf, B., and Bauer, S. Causalworld: A robotic manipulation benchmark for causal structure and transfer learning. *International Conference On Learning Representations*, 2020.

Angrist, J. D., Imbens, G. W., and Rubin, D. Identification of causal effects using instrumental variables. Working Paper 136, National Bureau of Economic Research, June 1993. URL http://www.nber.org/papers/t0136.

Battocchi, K., Dillon, E., Hei, M., Lewis, G., Oprescu, M., and Syrgkanis, V. Estimating the long-term effects of novel treatments. *Advances in Neural Information Processing Systems*, 34:2925–2935, 2021.

Beinlich, I. A., Suermondt, H. J., Chavez, R. M., and Cooper, G. F. *The ALARM Monitoring System: A Case Study with two Probabilistic Inference Techniques for Belief Networks*, pp. 247–256. Springer Berlin Heidelberg, 1989. doi: 10.1007/978-3-642-93437-7_28. URL http://link.springer.com/content/pdf/10.1007/978-3-642-93437-7_28.pdf.

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

Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. Better mixing via deep representations. *International Conference On Machine Learning*, 2012.

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

Bica, I., Alaa, A., Jordon, J., and Schaar, M. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. *International Conference On Learning Representations*, 2020.

Carvalho, C. M., Polson, N. G., and Scott, J. G. Handling sparsity via the horseshoe. In van Dyk, D. and Welling, M. (eds.), *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pp. 73–80, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA, 16–18 Apr 2009. PMLR. URL https://proceedings.mlr.press/v5/carvalho09a.html.

Chevalley, M., Roohani, Y., Mehrjou, A., Leskovec, J., and Schwab, P. CausalBench: A Large-scale Benchmark for Network Inference from Single-cell Perturbation Data. *arXiv preprint arXiv:2210.17283*, 2022.

Chickering, D. M. Optimal structure identification with greedy search. *J. Mach. Learn. Res.*, 3(null):507–554, mar 2003. ISSN 1532-4435. doi: 10.1162/153244303321897717. URL https://doi.org/10.1162/153244303321897717.

Chu, Z., Huang, J.-B., Li, R., Chu, W., and Li, S. Causal effect estimation: Recent advances, challenges, and opportunities. *ARXIV.ORG*, 2023. doi: 10.48550/arXiv.2302.00848.

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

Deleu, T., G'ois, A., Emezue, C. C., Rankawat, M., Lacoste-Julien, S., Bauer, S., and Bengio, Y. Bayesian structure learning with generative flow networks. *Conference On Uncertainty In Artificial Intelligence*, 2022.

Drton, M. and Maathuis, M. H. Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, 4(1):365–393, 2017. doi: 10.1146/annurev-statistics-060116-053803. URL https://doi.org/10.1146/annurev-statistics-060116-053803.

Geffner, T., Antorán, J., Foster, A., Gong, W., Ma, C., Kıcıman, E., Sharma, A., Lamb, A., Kukla, M., Pawlowski, N., Allamanis, M., and Zhang, C. Deep end-to-end causal inference. *ARXIV.ORG*, 2022.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

Hahn, P. R., Dorie, V., and Murray, J. S. Atlantic causal inference conference (acic) data analysis challenge 2017. *arXiv preprint arXiv: 1905.09515*, 2019.

Hasan, U., Hossain, E., and Gani, M. O. A survey on causal discovery methods for temporal and non-temporal data. *arXiv preprint arXiv: Arxiv-2303.15027*, 2023.

Heinze-Deml, C., Maathuis, M. H., and Meinshausen, N. Causal structure learning. *arXiv preprint arXiv: Arxiv-1706.09141*, 2017.

Huang, B., Zhang, K., Sanchez-Romero, R., Ramsey, J., Glymour, M., and Glymour, C. Diagnosis of autism spectrum disorder by causal influence strength learned from resting-state fmri data. *Neural Engineering Techniques For Autism Spectrum Disorder*, 2019. doi: 10.1016/b978-0-12-822822-7.00012-0.

Jain, M., Bengio, E., Hernández-García, A., Rector-Brooks, J., Dossou, B. F. P., Ekbote, C., Fu, J., Zhang, T., Kilgour, M., Zhang, D., Simine, L., Das, P., and Bengio,

Y. Biological sequence design with gflownets. *International Conference On Machine Learning*, 2022. doi: 10.48550/arXiv.2203.04115.

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

Jasra, A., Holmes, C. C., and Stephens, D. A. Markov chain monte carlo methods and the label switching problem in bayesian mixture modeling. *Statistical Science*, 20(1): 50–67, 2005. ISSN 08834237. URL http://www.jstor.org/stable/20061160.

Kalainathan, D. and Goudet, O. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv: 1903.02278*, 2019a.

Kalainathan, D. and Goudet, O. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv: 1903.02278*, 2019b.

Karavani, E., Shimoni, Y., and Yanover, C. Ibm causal inference benchmarking framework, January 2018. URL https://doi.org/10.5281/zenodo.1163587.

Keith Battocchi, Eleanor Dillon, M. H. G. L. P. O. M. O. V. S. EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/py-why/EconML, 2019. Version 0.x.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009. ISBN 0262013193.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

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

Maathuis, M. H., Kalisch, M., and Bühlmann, P. Estimating high-dimensional intervention effects from observational data. *arXiv preprint arXiv: 0810.4214*, 2008. URL https://arxiv.org/abs/0810.4214v3.

MacDorman, M. F. and Atkinson, J. O. Infant mortality statistics from the linked birth/infant death data set–1995 period data. *Mon Vital Stat Rep*, 46(6 Suppl 2):1–22, February 1998.

Madigan, D., Gavrin, J., and Raftery, A. E. Eliciting prior information to enhance the predictive performance of bayesian graphical models. *Communications in Statistics - Theory and Methods*, 24(9):2271–2292, 1995a. doi: 10.1080/03610929508831616. URL https://doi.org/10.1080/03610929508831616.

Madigan, D., York, J., and Allard, D. Bayesian graphical models for discrete data. *International Statistical Review / Revue Internationale de Statistique*, 63(2): 215–232, 1995b. ISSN 03067734, 17515823. URL http://www.jstor.org/stable/1403615.

Maiya, A. S. Causalnlp: A practical toolkit for causal inference with text. *ARXIV.ORG*, 2021.

Menegozzo, G., Dall'Alba, D., and Fiorini, P. Cipcad-bench: Continuous industrial process datasets for benchmarking causal discovery methods. *Ieee 18th International Conference On Automation Science And Engineering (case)*, 2022. doi: 10.1109/CASE49997.2022.9926420.

Muller, S., Hollmann, N., Arango, S. P., Grabocka, J., and Hutter, F. Transformers can do bayesian inference. *International Conference On Learning Representations*, 2021.

Naser, M. Causality, causal discovery, and causal inference in structural engineering. *ARXIV.ORG*, 2022. doi: 10.48550/arXiv.2204.01543.

Neal, B., Huang, C.-W., and Raghupathi, S. Realcause: Realistic causal inference benchmarking. *arXiv preprint arXiv: 2011.15007*, 2020.

Pearl, J. Causal diagrams for empirical research. *Biometrika*, 82:669–688, 1995.

Pearl, J. Causal inference in statistics: An overview. *Statistics Surveys*, 3(none):96 – 146, 2009a. doi: 10.1214/09-SS057. URL https://doi.org/10.1214/09-SS057.

Pearl, J. *Causality*. Cambridge University Press, 2 edition, 2009b. doi: 10.1017/CBO9780511803161.

Pearl, J. and Mackenzie, D. The book of why: The new science of cause and effect. *Science*, 361(6405):855–855, 2018. doi: 10.1126/science.aau9731. URL https://www.science.org/doi/abs/10.1126/science.aau9731.

Ramdas, A., García Trillos, N., and Cuturi, M. On wasserstein two-sample testing and related families of nonparametric tests. *Entropy*, 19(2):47, 2017.

Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., and Sejdinovic, D. Detecting causal associations in large nonlinear time series datasets. *arXiv preprint arXiv: 1702.07007*, 2017.

Runge, J., Bathiany, S., Bollt, E., Camps-Valls, G., Coumou, D., Deyle, E., Glymour, C., Kretschmer, M., Mahecha, M. D., Muñoz-Marí, J., van Nes, E. H., Peters, J., Quax, R., Reichstein, M., Scheffer, M., Schölkopf, B., Spirtes, P., Sugihara, G., Sun, J., Zhang, K., and Zscheischler, J. Inferring causation from time series in earth system sciences. *Nature Communications*, 2019. doi: 10.1038/s41467-019-10105-3. URL https://doi.org/10.1038/s41467-019-10105-3.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721): 523–529, 2005. doi: 10.1126/science.1105809. URL https://www.science.org/doi/abs/10.1126/science.1105809.

Sanchez, P., Voisey, J. P., Xia, T., Watson, H. I., O'Neil, A. Q., and Tsaftaris, S. A. Causal machine learning for healthcare and precision medicine. *Royal Society Open Science*, 9(8):220638, 2022.

Scutari, M. Bayesian network constraint-based structure learning algorithms: Parallel and optimised implementations in the bnlearn r package. *arXiv preprint arXiv: 1406.7648*, 2014.

Sharma, A. and Kiciman, E. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.

Sharma, A., Kiciman, E., et al. DoWhy: A Python package for causal inference. https://github.com/microsoft/dowhy, 2019.

Shimoni, Y., Yanover, C., Karavani, E., and Goldschmnidt, Y. Benchmarking framework for performance-evaluation of causal inference analysis. *arXiv preprint arXiv: 1802.05046*, 2018.

Spirtes, P., Glymour, C., and Scheines, R. *Causation, Prediction, and Search*. The MIT Press, 01 2001. ISBN 9780262284158. doi: 10.7551/mitpress/1754.001.0001. URL https://doi.org/10.7551/mitpress/1754.001.0001.

Tange, O. Gnu parallel - the command-line power tool. *;login: The USENIX Magazine*, 36(1):42–47, Feb 2011. doi: http://dx.doi.org/10.5281/zenodo.16303. URL http://www.gnu.org/s/parallel.

Tigas, P., Annadani, Y., Jesson, A., Scholkopf, B., Gal, Y., and Bauer, S. Interventions, where and how? experimental design for causal models at scale. *Neural Information Processing Systems*, 2022. doi: 10.48550/arXiv.2203.02016.

Tu, R., Zhang, C., Ackermann, P., Mohan, K., Kjellström, H., and Zhang, K. Causal discovery in the presence of missing data. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pp. 1762–1770. PMLR, 16–18 Apr 2019. URL https://proceedings.mlr.press/v89/tu19a.html.

Viinikka, J., Hyttinen, A., Pensar, J., and Koivisto, M. Towards scalable bayesian learning of causal dags. *Advances in Neural Information Processing Systems*, 33: 6584–6594, 2020.

Vowels, M. J., Camgoz, N. C., and Bowden, R. D'ya like dags? a survey on structure learning and causal discovery. *ACM Comput. Surv.*, 55(4), nov 2022. ISSN 0360-0300. doi: 10.1145/3527154. URL https://doi.org/10.1145/3527154.

Waisman, C., Nair, H. S., and Carrion, C. Online causal inference for advertising in real-time bidding auctions. *arXiv preprint arXiv: Arxiv-1908.08600*, 2019.

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

Zhang, K. and Chan, L.-W. Extensions of ica for causality discovery in the hong kong stock market. In King, I., Wang, J., Chan, L.-W., and Wang, D. (eds.), *Neural Information Processing*, pp. 400–409, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-46485-3.

# A. Related Work

**Benchmarking methods** Benchmarks have played a crucial role in advancing entire research fields, for instance computer vision with the introduction of ImageNet (Krizhevsky et al., 2012). When it comes to causal discovery, benchmarks usually come in the form of research surveys (Vowels et al., 2022), benchmark datasets (Beinlich et al., 1989; Chevalley et al., 2022; Menegozzo et al., 2022), learning environments (Runge et al., 2019; 2017; Ahmed et al., 2020), and software packages or platforms (Kalainathan & Goudet, 2019a; Scutari, 2014; Kalainathan & Goudet, 2019b). However these methods only evaluate the closeness of the causal DAG, or the samples from the posterior distribution of DAGs in Bayesian structure learning, from various causal discovery methods to the ground-truth DAG. Measuring the proximity of the learned DAGs, however, does not reveal much about their actual performance in treatment effect estimation given a treatment and outcome variable of interest, which is a predominantly downstream evaluation.

In causal inference, datasets (MacDorman & Atkinson, 1998; Hahn et al., 2019), frameworks (Shimoni et al., 2018; Karavani et al., 2018; Neal et al., 2020), and software packages (Sharma et al., 2019; Keith Battocchi, 2019) provide valuable tools for predicting the causal effects of treatments on outcomes. Causal inference plays a crucial role in decision-making and finds numerous practical applications in various domains such as healthcare, advertising, and decision-making processes. This implies that causal inference has a more downstream impact. In causal inference, the graph represents the structure of the joint distribution of variables, which is then leveraged to identify the causal estimand.

Therefore, the evaluation of causal discovery methods on downstream causal inference tasks provides more practical insights into the effectiveness and practicality of causal methods within real-world scenarios. Typically, the fields of causal discovery and inference are approached separately, resulting in limited intertwined evaluation methods. This is the aspect that distinguishes our work. Similar approaches can be found in studies that jointly integrate causal discovery and inference in an end-to-end manner, such as the notable example of DECI (Geffner et al., 2022). However, our work differs in two key aspects: firstly, we employ the novel GFlowNets for causal inference, increasing our span and secondly, we specifically focus on linear noise structural equation models, whereas DECI addresses the problem of end-to-end causal inference in non-linear additive noise structural equation models (SEM).

# B. Background

We offer a detailed background, in this section, on some of the key concepts used in this paper.

**Bayesian network:** A (causal) Bayesian network (Koller & Friedman, 2009; Pearl, 2009b) is a probabilistic model over $d$ random variables $\{X_1, ..., X_d\}$, whose joint probability distribution factorizes according to a DAG G (whose edges express causal dependencies) as:

$$P(X_1, ..., X_d) = \prod_{k=1}^{d} P(X_k | Pa_G(X_k)), \tag{1}$$

where $Pa_G(X)$ is the set of parents of the node $X$, i.e the nodes with an edge onto $X$ in G, interpreted as the direct causes of $X$.

**Interventional distribution:** Given a random variable $X_k$, a (hard) intervention on $X_k$, denoted by $do(X_k = a)$, is obtained by replacing the conditional probability distribution (CPD) $P(X_k|Pa_G(X_k))$ with a Dirac distribution $\delta_{X_k=a}$ which forces $X_k$ to take on the value of $a$. Note that intervening on a variable, in a graphical sense, results in a mutilated graph where all incoming edges to the node corresponding to that variable are removed (Pearl, 2009a).

### B.1. (Bayesian) Causal discovery

Given a dataset $D \stackrel{\text{def}}{=} \left\{\mathbf{x}^{(i)}\right\}_{i=1}^{n}$ of $n$ observations, such that $\mathbf{x}^{(j)} \sim P(X_1, ..., X_d)$, the goal of structure learning is to learn the DAG $G$ corresponding to the causal Bayesian network that best models $D$. It is important to note that $D$ could be observational samples or interventional data samples (got from performing hard or soft interventions). In a Bayesian structure learning setting, the task is to approximate the posterior distribution $P(G|D)$ over Bayesian networks that model these observations. A distribution over the DAGs allows quantifying the epistemic uncertainty and the degree of confidence in any given Bayesian network model, which is especially useful when the amount of data to learn from is small (Lorch et al., 2021; Muller et al., 2021).

### B.2. Average treatment effect (ATE) estimation

The average treatment effect (ATE) is a quantity that allows us to estimate the *impact* of a treatment variable on an outcome variable. Given $X_T$ and $X_Y$, our treatment and effect variables of interest respectively, the ATE on targets $X_Y$ for treatment $X_T = a$ given a reference $X_T = b$ is given by (Pearl, 2009a):

$$ATE(a, b) = \mathbb{E}[X_Y|do(X_T = b)] - \mathbb{E}[X_Y|do(X_T = a)]$$

In practice, this causal inference is broken down into two steps: identification and estimation. Identification deals with converting the causal estimand $P(X_Y|do(X_T = b)$ into a statistical estimand that can be estimated using the dataset $D$. Some identification methods include the back-door criterion, front-door criterion (Pearl, 2009a), instrumental variables (Angrist et al., 1993) and mediation. Causal estimation then computes the identified statistical estimand from the data set using a range of statistical methods. The do-calculus algorithm (Pearl, 1995) provides a powerful, systematic, programmable framework for the identification and estimation of the causal estimand.

### B.3. Causal discovery baseline algorithms

In Table 4 we briefly describe the structure learning algorithms we use in this work. The structure learning baselines employed in our study follow those utilized by Deleu et al. (2022). For PC and GES we implement bootstrapping to achieve DAG posterior samples.

*Table 4.* Details of baseline algorithms. We document the underlying algorithm behind each baseline, alongside an indication of whether there is a guarantee that the learned graph is a DAG.

| Baseline | Underlying algorithm | DAG Support |
|---|---|---|
| PC | constraint-based | ✔ |
| GES | score-based | ✔ |
| MC3 | MCMC | ✔ |
| Gadget | MCMC | ✔ |
| DiBS | variational inference | ✗ |
| BCDNets | variational inference | ✔ |
| DAG-GFlowNet | GFlowNets | ✔ |

**DAG-GFlowNet:** DAG-GFlowNet (Deleu et al., 2022) employs GFlowNets (Bengio et al., 2021a;b) as a substitute for MCMC in order to estimate the posterior distribution of Bayesian network structures, based on a set of observed data. An overview of GFlowNets is presented in Section C of the Appendix. The process of creating a sample DAG from an approximate distribution is considered a sequential decision task. This involves constructing the graph incrementally, one edge at a time, by utilizing transition probabilities that have been learned by a GFlowNet. We refer the reader to Deleu et al. (2022) for a comprehensive study of DAG-GFlowNet.

**DiBS:** The DiBS framework (Lorch et al., 2021) is an approach to Bayesian structure learning that is fully differentiable. It operates within the continuous space of a latent probabilistic graph representation. In contrast to prior research, the DiBS method does not rely on a specific format for the local conditional distributions. Additionally, it enables the simultaneous estimation of the graph structure and the parameters of the conditional distributions.

**MC3:** In the MC3 algorithm (also known as structured MCMC) (Madigan et al., 1995b), the authors present a hierarchical Bayesian approach to structure learning that leverages a prior over the classes of variables using nonparametric block-structured priors over Bayes net graph structures. This approach relies heavily on the assumption that variables come in one or more classes and that the prior probability of an edge existing between two variables is a function only of their classes (Madigan et al., 1995b).

**GES:** The Greedy Equivalence Search (GES) algorithm (Chickering, 2003) is a score-based method for causal discovery that has been in use for a considerable amount of time. It operates by performing a greedy search across the set of equivalence

classes of DAGs. The representation of each search state is accomplished through a completed partially directed acyclic graph (CPDAG), which includes operators for the insertion and deletion of edges. These operators enable the addition or removal of a single edge, respectively (Hasan et al., 2023).

**PC:** The Peter-Clark (PC) algorithm (Spirtes et al., 2001) is a prominent constraint-based method for causal discovery. It leverages conditional independence (CI) tests to infer the underlying causal structure. The algorithm yields a completed partially directed acyclic graph (CPDAG) that represents the relationships between variables. It follows a three-step process: 1) identifying the skeleton of the graph, 2) determining v-structures or colliders $(X \longrightarrow Y \longleftarrow Z)$ based on d-separation, and 3) propagating edge orientations. Initially, the algorithm creates a fully connected undirected graph using all variables in the dataset. It then eliminates edges that are unconditionally or conditionally independent (skeleton detection), identifies and orients v-structures using the d-separation set, and finally orients the remaining edges while ensuring the absence of new v-structures and cycles. The PC algorithm relies on the assumptions of acyclicity, causal faithfulness, and causal sufficiency.

**BCDNets:** BCDNets (Cundy et al., 2021) is another variational inference framework like DiBS. In their work they focus on estimating a distribution over DAGs characterizing a linear-Gaussian SEM and propose techniques to scale to high dimensions, such as using deep neural networks to model a variational family of factorized posterior distributions over the SEM parameters (including the edge weights and noise variance), and a horseshoe prior (Carvalho et al., 2009) on the edge weights, which promotes sparsity.

**Gadget:** Gadget (Viinikka et al., 2020) is based on MCMC: sampling DAGs by simulating a Markov chain whose stationary distribution is the posterior distribution. However, to enhance the mixing of the chain, and reduce the space and time requirements, they build a Markov chain on the smaller space of ordered partitions of the node set, each state being associated with multiple DAGs.

## C. Generative Flow Networks (GFlowNets)

The Generative Flow Networks (Bengio et al., 2021a;b), also known as GFlowNets, are a type of inference models that have a broad range of applications. GFlowNets are capable of generating samples with a probability that is proportional to a given reward function. The GFlowNets have been extensively studied and discussed in research papers such as Bengio et al. (2021a) and Bengio et al. (2021b). The models facilitate the process of selecting a varied pool of potential candidates, while adhering to a training objective that ensures a nearly proportional sampling based on a specified reward function. GFlowNets are characterized unique training objectives like the *flow-matching condition* (Bengio et al., 2021a), *the detailed balance condition* (Bengio et al., 2021b), etc, through which a policy is learned. Through the training objectives, this policy is designed to ensure that the probability $P_T(s)$ of sampling an object $s$ is roughly proportional to the value $R(s)$ of a specified reward function applied to that object. The GFlowNets technique is designed to reduce the computational burden of MCMC methods by performing the necessary work in a single generative pass that has been trained for this purpose.

GFlowNets are well-suited for modeling and sampling from distributions over sets and graphs, as well as estimating free energies and marginal distributions (Jain et al., 2022; Zhang et al., 2022). They excel in problem scenarios with specific characteristics (Jain et al., 2023): (1) the ability to define or learn a non-negative or non-marginalized reward function that determines the distribution to sample from, (2) the presence of a highly multi modal reward function, showcasing GFlowNets' strength in generating diverse samples, and (3) the benefit of sequential sampling, where compositional structure can be leveraged for sequential generation.

Since its inception, GFlowNets have exhibited promising results in diverse domains such as discrete probabilistic modeling (Zhang et al., 2022), molecular design (Bengio et al., 2021a; Jain et al., 2022), and causal discovery (Deleu et al., 2022). The aim of our research is to provide significant findings on the feasibility of employing GFlowNets for causal inference.

## D. Regrouping ATE values

The estimation of average treatment effects (ATE) through regression analysis is susceptible to generating estimates that may exhibit slight variations within numerical precision (e.g., 1.000000001 and 1). As our precision and recall metrics essentially perform 'hard matches" on floating point values, it becomes crucial to consider the influence of numerical precision. In order to accomplish this objective, we group ATE values that are numerically close. We use the following equation to test

whether two floating point values, $a$ and $b$, are equivalent:

$$|a - b| <= (atol + rtol * |b|),$$

where $rtol$ is the relative tolerance parameter and $atol$ is the absolute tolerance parameter. Practically, we use the 'isclose' function from the Numpy package[2] which uses the equation above and returns a boolean indicating whether $a$ and $b$ are equal within the given tolerance. We used the default values from Numpy, $rtol = 1e - 05$, $atol = 1e - 08$. We apply regrouping to the list of ATEs for precision and recall evaluation, but not for Wasserstein distance.

[2] https://numpy.org/doc/stable/reference/generated/numpy.isclose.html