BayesDAG: Gradient-Based Posterior Sampling for Causal Discovery

Yashas Annadani [*1 2] **Nick Pawlowski** [3] **Joel Jennings** [3] **Stefan Bauer** [1 4 5] **Cheng Zhang** [3] **Wenbo Gong** [*3]

Abstract

Bayesian causal discovery aims to infer the posterior distribution over causal models from observed data, quantifying epistemic uncertainty and benefiting downstream tasks. However, computational challenges arise due to joint inference over combinatorial space of Directed Acyclic Graphs (DAGs) and nonlinear functions. In this work, we introduce a scalable Bayesian causal discovery framework based on stochastic gradient Markov Chain Monte Carlo (SG-MCMC) that directly samples DAGs from the posterior without any DAG regularization, simultaneously draws function parameter samples and is applicable to both linear and nonlinear causal models. To enable our approach, we derive a novel equivalence to the permutation-based DAG learning, which opens up possibilities of using any relaxed gradient estimator defined over permutations. To our knowledge, this is the first framework applying gradient-based MCMC sampling for causal discovery. Empirical evaluations on synthetic and real-world datasets demonstrate our approach's effectiveness compared to state-of-the-art baselines.

1. Introduction

The quest for discovering causal relationships in data-generating processes lies at the heart of empirical sciences and decision-making (Pe'er et al., 2001; Sachs et al., 2005; Van Koten & Gray, 2006). Structural Causal Models (SCMs) (Pearl, 2009) and their associated Directed Acyclic Graphs (DAGs) provide a mathematical framework for modeling such relationships. Knowledge of the underlying SCM permits predictions of unseen interventions and causal reasoning, thus making causal discovery – learning an unknown SCM and its associated DAG from observed data – a subject of extensive research (Peters et al., 2017).

In contrast to traditional methods that infer a single graph or its Markov equivalence class (MEC) (Chickering, 2002; Spirtes et al., 2000), Bayesian causal discovery (Friedman & Koller, 2003; Heckerman et al., 2006; Tong & Koller, 2001) aims to infer a posterior distribution over SCMs and their DAGs from observed data. This approach encapsulates the epistemic uncertainty, degree of confidence in every causal hypothesis, which is particularly valuable for real-world applications when data is scarce. It is also beneficial for downstream tasks such as experimental design (Annadani et al., 2023; Murphy, 2001; Tigas et al., 2022).

The central challenge in Bayesian causal discovery lies in inferring the posterior distribution over the union of the exponentially growing (discrete) DAGs space and (continuous) causal model parameters. Prior works have used Markov Chain Monte Carlo (MCMC) to directly sample DAGs or bootstrap traditional discovery methods (Chickering, 2002; Murphy, 2001; Tong & Koller, 2001), but these methods are typically limited to linear models which admit closed-form marginalization over continuous parameters. Recent approaches have begun to utilize gradient information for more efficient inference. These approaches are based on either the DAG regularizer (Zheng et al., 2018) (for e.g. DIBS (Lorch et al., 2021)) with continuous relaxation of adjacency matrices, or permutation based methods which directly infer permutation matrices over nodes through Variational Inference (VI) (Cundy et al., 2021). DAG regularizer based methods are usually computationally expensive due to the DAG constraint and cannot guarantee DAG generation, while existing permutation based methods are usually restricted to only linear causal models.

In this work, we propose `BayesDAG`, which overcomes the above limitations. Our contributions are:

1. We prove that an augmented space of edge beliefs and node potentials $(\boldsymbol{W}, \boldsymbol{p})$, similar to NoCurl (Yu et al., 2021), permits equivalent Bayesian inference in DAG space without the need for any regularizer.

2. We derive an equivalence relation from this augmented space to permutation-based DAG learning which provides a general framework for gradient-based posterior inference.

[*]Equal contribution. This work was done during YA's internship at Microsoft Research. [1]Helmholtz AI [2]KTH Stockholm [3]Microsoft Research [4]TU Munich [5]CIFAR Azrieli Global Scholar. Correspondence to: Wenbo Gong <wenbogong@microsoft.com>.

Accepted to ICML workshop on Structured Probabilistic Inference & Generative Modeling, Honolulu, Hawaii, USA. 2023.

3. Based on this general framework, we propose a scalable Bayesian causal discovery approach with SG-MCMC for DAGs and causal model parameters. Our approach is model-agnostic for linear and non-linear cases and also offers improved inference quality with sampling approach (Ma et al., 2015).

2. Background

Structural Causal Model The causal relationships among d variables $\boldsymbol{X} \in \mathbb{R}^d$ can represented by a Structural Causal Model (SCM) which consists of a set of structural equations (Peters et al., 2017) where each variable X_i is a function of its direct causes $\boldsymbol{X}_{\mathbf{Pa}^i}$ and an exogenous noise variable ϵ_i (for e.g Gaussian):

$$X_i := f_i(\boldsymbol{X}_{\mathbf{Pa}^i}) + \epsilon_i \quad \text{with } \epsilon_i \sim \mathcal{N}(0, \sigma_i) \tag{1}$$

These equations induce a causal graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, typically assumed to be a DAG (Pearl, 2009). A directed edge between a node pair $v_i, v_j \in \boldsymbol{V}$ (i.e., $v_i \to v_j$) represents that X_i causes X_j. We use the binary adjacency matrix $\boldsymbol{G} \in \{0,1\}^{d\times d}$ to represent the causal graph. Given a dataset $\boldsymbol{D} = \{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(N)}\}$, DAG $\boldsymbol{G}$ and SCM functional parameters $\boldsymbol{\Theta}$, they induce a unique joint distribution $p(\boldsymbol{D}, \boldsymbol{\Theta}, \boldsymbol{G}) = p(\boldsymbol{D}|\boldsymbol{G}, \boldsymbol{\Theta})p(\boldsymbol{G}, \boldsymbol{\Theta})$ with the prior $p(\boldsymbol{G}, \boldsymbol{\Theta})$ and likelihood $p(\boldsymbol{D}|\boldsymbol{G}, \boldsymbol{\Theta})$ (Friedman & Koller, 2003). Bayesian causal discovery aims to infer the posterior $p(\boldsymbol{G}, \boldsymbol{\Theta}|\boldsymbol{D}) = p(\boldsymbol{D}, \boldsymbol{\Theta}, \boldsymbol{G})/p(\boldsymbol{D})$. However, this posterior is intractable due to super-exponential growth of the possible DAGs $\boldsymbol{G}$ (Robinson, 1973) and continuously valued model parameters $\boldsymbol{\Theta}$ of nonlinear functions. VI (Zhang et al., 2018) or SG-MCMC (Gong, 2022; Ma et al., 2015) are two types of methods developed to tackle general Bayesian inference problems, but adaptations are required for Bayesian causal discovery.

NoCurl Characterization Inferring causal graphs is challenging due to the DAG constraint. Recently, (Yu et al., 2021) introduced NoCurl, a novel characterization of the **weighted DAG** space. They define a potential $p_i \in \mathbb{R}$ for each node i, grouped as potential vector $\boldsymbol{p} \in \mathbb{R}^d$. They also introduce a gradient operator on $\boldsymbol{p}$, i.e. $(\operatorname{grad} \boldsymbol{p})(i,j) = p_i - p_j$. Based on the above operation, a mapping that directly maps from an augmented space $(\boldsymbol{W}, \boldsymbol{p})$ to the DAG space, $\gamma(\cdot,\cdot) : \mathbb{R}^{d\times d} \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$, was proposed:

$$\gamma(\boldsymbol{W}, \boldsymbol{p}) = \boldsymbol{W} \odot \operatorname{ReLU}(\operatorname{grad} \boldsymbol{p}) \tag{2}$$

where $\operatorname{ReLU}(\cdot)$ is the ReLU activation function and $\boldsymbol{W}$ is a skew-symmetric **continuously weighted** matrix. This formulation is complete (Theorem 2.1 in (Yu et al., 2021)), as any continuously weighted DAG can be represented by a $(\boldsymbol{W}, \boldsymbol{p})$ pair and vice versa. NoCurl translates the learning of a single weighted DAG to a corresponding $(\boldsymbol{W}, \boldsymbol{p})$ pair. However, direct gradient-based optimization is challenging due to a highly non-convex loss landscape, which leads to the reported failure in (Yu et al., 2021). Although NoCurl appears suitable for our purpose, the failure in directly learning suggests non-trivial optimizations. We hypothesize that this arises from the continuously weighted matrix $\boldsymbol{W}$. In the following, we introduce our proposed parametrization inspired by NoCurl to characterize the **binary** DAG adjacency matrix.

3. Sampling the DAGs

3.1. Bayesian Inference in W, p Space

The NoCurl formulation (Equation (2)) focuses on learning *a single weighted* DAG, which is not directly useful for our purpose. We need to address two key questions: (1) considering only binary adjacency matrices without weights; (2) ensuring Bayesian inference in $(\boldsymbol{W}, \boldsymbol{p})$ is valid.

Consider the mapping $\tau : \{0,1\}^{d\times d} \times \mathbb{R}^d \to \{0,1\}^{d\times d}$:

$$\tau(\boldsymbol{W}, \boldsymbol{p}) = \boldsymbol{W} \odot \operatorname{Step}(\operatorname{grad} \boldsymbol{p}) \tag{3}$$

where we abuse the term $\boldsymbol{W}$ for binary matrices, and replace $\operatorname{ReLU}(\cdot)$ with $\operatorname{Step}(\cdot)$. $\boldsymbol{W}$ acts as mask to disable the edge existence. Thus, due to the Step, τ can only output a binary adjacency matrix. We show that performing Bayesian inference in this modified augmented $(\boldsymbol{W}, \boldsymbol{p})$ space is valid, i.e., using the posterior $p(\boldsymbol{W}, \boldsymbol{p}|\boldsymbol{D})$ to replace $p(\boldsymbol{G}|\boldsymbol{D})$. This differs from NoCurl, which focuses on a single graph rather than the validity for Bayesian inference, requiring a new theory for soundness. We provide the main results and relegate all proofs and details to the appendix.

Theorem 3.1 (Equivalence of inference in $(\boldsymbol{W}, \boldsymbol{p})$ and binary DAG space)**.** *Assume graph $\boldsymbol{G}$ is a binary adjacency matrix representing a DAG and node potential $\mathbf{p}$ does not contain the same values, i.e. $p_i \neq p_j \ \forall i, j$. Then, with the induced joint observational distribution $p(\boldsymbol{D}, \boldsymbol{G})$, dataset $\boldsymbol{D}$, and a corresponding prior $p(\boldsymbol{G})$, we have*

$$p(\boldsymbol{G}|\boldsymbol{D}) = \int p_\tau(\boldsymbol{p}, \boldsymbol{W}|\boldsymbol{D})\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}\, d\boldsymbol{p} \tag{4}$$

if $p(\boldsymbol{G}) = \int p_\tau(\boldsymbol{p}, \boldsymbol{W})\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}\, d\boldsymbol{p}$, where $p_\tau(\boldsymbol{W}, \boldsymbol{p})$ is the prior, $\mathbb{1}(\cdot)$ is the indicator function, and $p_\tau(\boldsymbol{p}, \boldsymbol{W}|D)$ is the posterior distribution over $\boldsymbol{p}, \boldsymbol{W}$.

This theorem guarantees that instead of performing inference directly in the constrained space (i.e. DAG space), we can apply Bayesian inference in a less complex $(\boldsymbol{W}, \boldsymbol{p})$ space where $\boldsymbol{W} \in \{0,1\}^{d\times d}$ and $\boldsymbol{p} \in \mathbb{R}^d$.

For inference of $\boldsymbol{p}$, we adopt a sampling-based approach, which is asymptotically accurate (Ma et al., 2015) and provides better inference quality compared to VI (Gong et al.,

2019; Springenberg et al., 2016; Trippe & Turner, 2018). In particular, we consider SG-MCMC (refer to Section 4), which avoids the expensive Metropolis-Hastings acceptance step and scales to large datasets. We emphasize that any other suitable sampling algorithms can be directly plugged in, thanks to the generality of the framework.

3.2. Equivalent Formulation

The mapping τ does not provide meaningful gradient information for $\boldsymbol{p}$ due to the piecewise constant $\mathrm{Step}(\cdot)$ function. We address this issue by deriving an equivalence to a permutation learning problem that enables approximating the gradient of $\boldsymbol{p}$.

Intuition The node potential $\boldsymbol{p}$ implicitly defines a topological ordering through the mapping $\mathrm{Step}(\mathrm{grad}(\cdot))$. In particular, $\mathrm{grad}(\cdot)$ outputs a skew-symmetric adjacency matrix, where each entry specifies the potential difference between nodes. $\mathrm{Step}(\mathrm{grad}(\cdot))$ zeros out the negative potential differences (i.e. $p_i \leq p_j$), and only permits the edge direction from higher potential to the lower one (i.e. $p_i > p_j$). This implicitly defines a sorting operation based on the node potentials, which can be cast as a particular $\arg\max$ problem (Blondel et al., 2020; Kuhn, 1955; Mena et al., 2018; Niculae et al., 2018) involving a permutation matrix.

Alternative formulation We define $\boldsymbol{L} \in \{0,1\}^{d\times d}$ as a matrix with lower triangular part to be 1, and vector $\boldsymbol{o} = [1,\dots,d]$. We propose the following formulation:

$$\boldsymbol{G} = \boldsymbol{W} \odot \left[\boldsymbol{\sigma}(\boldsymbol{p})\boldsymbol{L}\boldsymbol{\sigma}(\boldsymbol{p})^T\right] \ , \boldsymbol{\sigma}(\boldsymbol{p}) = \arg\max_{\boldsymbol{\sigma}'\in\boldsymbol{\Sigma}_d} \boldsymbol{p}^T(\boldsymbol{\sigma}'\boldsymbol{o}) \tag{5}$$

Here, $\boldsymbol{\Sigma}_d$ represents the space of all d dimensional permutation matrices. The following theorem states the equivalence of this formulation to Equation (3).

Theorem 3.2 (Equivalence to NoCurl formulation)**.** *Assuming the conditions in Theorem 3.1 are satisfied. Then, for a given $(\boldsymbol{W},\boldsymbol{p})$, we have*

$$\boldsymbol{G} = \boldsymbol{W} \odot \mathrm{Step}(\mathrm{grad}\,\boldsymbol{p}) = \boldsymbol{W} \odot \left[\boldsymbol{\sigma}(\boldsymbol{p})\boldsymbol{L}\boldsymbol{\sigma}(\boldsymbol{p})^T\right]$$

where $\boldsymbol{G}$ is a DAG and $\boldsymbol{\sigma}(\boldsymbol{p})$ is defined in Equation (5).

This theorem translates our proposed operator $\mathrm{Step}(\mathrm{grad}(\boldsymbol{p}))$ into finding a corresponding permutation matrix $\boldsymbol{\sigma}(\boldsymbol{p})$. Although this does not directly solve the uninformative gradient, it opens the door for approximating this gradient with the tools from the differentiable permutation literature (Blondel et al., 2020; Mena et al., 2018; Niculae et al., 2018). For simplicity, we adopt the relaxed Gumbel-Sinkhorn approach (Mena et al., 2018), but we emphasize that this equivalence is general enough that any past or future approximation methods can be easily applied. To get the binary permutation matrix, we apply the Hungarian algorithm (Munkres, 1957) and use a straight-through estimator for $\boldsymbol{p}$.

Some of the previous works (Charpentier et al., 2022; Cundy et al., 2021) have leveraged the Sinkhorn operator to model variational distributions over permutation matrices. However, they start with a full rank $\boldsymbol{M}$, which has been reported to require over **1000** Sinkhorn iterations to converge (Cundy et al., 2021). However, our formulation, based on explicit node potential $\boldsymbol{p}\boldsymbol{o}^T$, generates a rank-1 matrix, requiring much fewer Sinkhorn steps (around **300**) in practice, saving two-thirds of the computational cost.

4. Bayesian Causal Discovery via Sampling

In this section, we delve into two inference algorithms that are derived from the proposed framework. The first one, which will be our main focus, combines SG-MCMC and VI in a Gibbs sampling manner. The second one, which is based entirely on SG-MCMC with continuous relaxation, is also derived, but we include its details in Appendix A due to its inferior empirical performance.

Model Formulation We build upon the model formulation of (Geffner et al., 2022), which combines the additive noise model with neural networks. Specifically, $X_i := f_i(\boldsymbol{X}_{\mathbf{Pa}^i}) + \epsilon_i$, where f_i adheres to the adjacency relation specified by $\boldsymbol{G}$, i.e. $\partial f_i(\boldsymbol{x})/\partial x_j = 0$ if no edge exists between nodes i and j. We define f_i as

$$f_i(\boldsymbol{x}) = \zeta_i\left(\sum_{j=1}^{d} G_{ji} l_j(x_j)\right) \tag{6}$$

where ζ_i and l_i are neural networks with parameters $\boldsymbol{\Theta}$, and $\boldsymbol{G}$ serves as a mask disabling non-parent values. To reduce the number of neural networks, we adopt a weight-sharing mechanism: $\zeta_i(\cdot) = \zeta(\boldsymbol{u}_i,\cdot)$ and $l_i(\cdot) = l(\boldsymbol{u}_i,\cdot)$, with trainable node embeddings $\boldsymbol{u}_i$.

Likelihood of SCM The likelihood can be evaluated through the noise variable (Geffner et al., 2022):

$$p(\boldsymbol{x}|\boldsymbol{G}) = \prod_{i=1}^{d} p_{\epsilon_i}(x_i - f_i(\boldsymbol{x}_{\mathbf{Pa}_G^i})) \tag{7}$$

Prior design We implicitly define the prior $p(\boldsymbol{G})$ via $p(\boldsymbol{p},\boldsymbol{W})$. We propose the following for the joint prior:

$$p(\boldsymbol{W},\boldsymbol{p},\boldsymbol{\Theta}) \propto \mathcal{N}(\boldsymbol{\Theta};\boldsymbol{0},\boldsymbol{I})\mathcal{N}(\boldsymbol{p};\boldsymbol{0},\alpha\boldsymbol{I})\mathcal{N}(\boldsymbol{W};\boldsymbol{0},\boldsymbol{I})\exp(-\lambda_s\|\tau(\boldsymbol{W},\boldsymbol{p})\|_F^2)$$

where α controls the initialization scale of $\boldsymbol{p}$ and λ_s controls the sparseness of $\boldsymbol{G}$.

4.1. Bayesian Inference of W, p, Θ

The main challenge lies in the binary nature of $\boldsymbol{W} \in \{0,1\}^{d\times d}$. We propose a combination of SG-MCMC for $\boldsymbol{p}, \boldsymbol{\Theta}$ and VI for $\boldsymbol{W}$. It should be noted that our framework can incorporate any suitable discrete sampler if needed. We employ a Gibbs sampling procedure (Casella & George, 1992), which iteratively applies (1) sampling $\boldsymbol{p}, \boldsymbol{\Theta} \sim p(\boldsymbol{p}, \boldsymbol{\Theta}|\boldsymbol{D}, \boldsymbol{W})$ with SG-MCMC; (2) updating the variational posterior $q_\phi(\boldsymbol{W}|\boldsymbol{p}, \boldsymbol{D}) \approx p(\boldsymbol{W}|\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D})$.

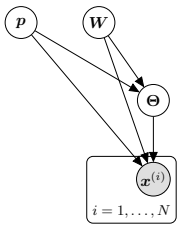

Figure 1: Graphical model of the inference problem.

We define the posterior $p(\boldsymbol{p}, \boldsymbol{\Theta}|\boldsymbol{D}, \boldsymbol{W}) \propto \exp(-U(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}))$, where $U(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) = -\log p(\boldsymbol{p}, \boldsymbol{D}, \boldsymbol{W}, \boldsymbol{\Theta})$. SG-MCMC in continuous time defines a specific form of Itô diffusion that maintains the target distribution invariant (Ma et al., 2015) without the expensive computation of the MH step. We adopt the Euler-Maruyama discretization for simplicity and employ the sampler based on (Gong et al., 2019), which is inspired by Adam (Kingma & Ba, 2014). Detailed update equations and gradient derivations can be found in Appendix C.

Variational inference for $\boldsymbol{W}$ We use the variational posterior $q_\phi(\boldsymbol{W}|\boldsymbol{p})$ to approximate the true posterior $p(\boldsymbol{W}|\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D})$. Specifically, we select an independent Bernoulli distribution with logits defined by the output of a neural network $\mu_\phi(\boldsymbol{p})$:

$$q_\phi(\boldsymbol{W}|\boldsymbol{p}) = \prod_{ij} Ber(\mu_\phi(\boldsymbol{p})_{ij}) \tag{8}$$

To train q_ϕ, we derive the corresponding *evidence lower bound* (ELBO):

$$\text{ELBO}(\phi) = \mathbb{E}_{q_\phi(\boldsymbol{W}|\boldsymbol{p})}\left[\log p(\boldsymbol{D}, \boldsymbol{p}, \boldsymbol{\Theta}|\boldsymbol{W})\right] - D_{\text{KL}}\left[q_\phi(\boldsymbol{W}|\boldsymbol{p}) \| p(\boldsymbol{W})\right] \tag{9}$$

where D_{KL} is the Kullback-Leibler divergence. The derivation is in Appendix B.6. Algorithm 2 summarizes this inference procedure.

SG-MCMC with continuous relaxation Furthermore, we explore an alternative formulation that circumvents the need for variational inference. Instead, we employ SG-MCMC to sample $\tilde{\boldsymbol{W}}$, a continuous relaxation of $\boldsymbol{W}$, facilitating a fully sampling-based approach. For a detailed formulation, please refer to Appendix A. We report its performance in Appendix G.2, which surprisingly is inferior to SG-MCMC+VI. We hypothesize that coupling $\boldsymbol{W}, \boldsymbol{p}$ through μ_ϕ is important since changes in $\boldsymbol{p}$ results in changes of the permutation matrix $\boldsymbol{\sigma}(\boldsymbol{p})$, which should also influence $\boldsymbol{W}$ accordingly during posterior inference. However, through sampling $\tilde{\boldsymbol{W}}$ with few SG-MCMC steps, this change cannot be immediately reflected, resulting in inferior performance. Thus, we focus only on the performance of SG-MCMC+VI for our experiments.

Computational complexity Our proposed SG-MCMC+VI offers a notable improvement in computational cost compared to existing approaches, such as DIBS (Lorch et al., 2021). The computational complexity of our method is $O(BN_p + N_p d^3)$, where B represents the batch size and N_p is the number of parallel SG-MCMC chains. This former term stems from the forward and backward passes, and the latter comes from the Hungarian algorithm, which can be parallelized to further reduce computational cost. In comparison, DIBS has a complexity of $O(N_p^2 N + N_p d^3)$ with $N \gg B$ being the full dataset size. This is due to the kernel computation involving the entire dataset and the evaluation of the matrix exponential in the DAG regularizer (Zheng et al., 2018). As a result, our approach provides linear scalability w.r.t. N_p with substantially smaller batch size B. Conversely, DIBS exhibits quadratic scaling in terms of N_p and lacks support for mini-batch gradients.

5. Experiments

We aim to study empirically the following aspects: (1) posterior inference quality of `BayesDAG` in high dimensional nonlinear causal models with synthetic datasets and (2) performance in semi-synthetic and real world applications.

Baselines and Metrics. We mainly compare `BayesDAG` with the following baselines: Bootstrap GES (**BGES**) (Chickering, 2002; Friedman et al., 2013), **BCD** Nets (Cundy et al., 2021), Differentiable DAG Sampling (**DDS** (Charpentier et al., 2022)) and **DIBS** (Lorch et al., 2021). For evaluation, we compute the expected SHD ($\mathbb{E}$**-SHD**), expected orientation F1 score (**Edge F1**) and negative log-likelihood of the held-out data (**NLL**). Our synthetic data generation and evaluation protocol follows prior work (Annadani et al., 2021; Geffner et al., 2022; Lorch et al., 2021). All the experimental details, including hyperparameters are provided in Appendix F.

5.1. Evaluation on Synthetic Data

We evaluate our method on synthetic data with nonlinear functional relations, where ground truth graphs are known. We generate data by randomly sampling DAGs from Erdos-Rènyi (ER) (Erdős et al., 1960) or Scale-Free (SF) (Barabási & Albert, 1999) graphs and drawing at random ground truth MLP parameters for nonlinear functions. We assess perfor-

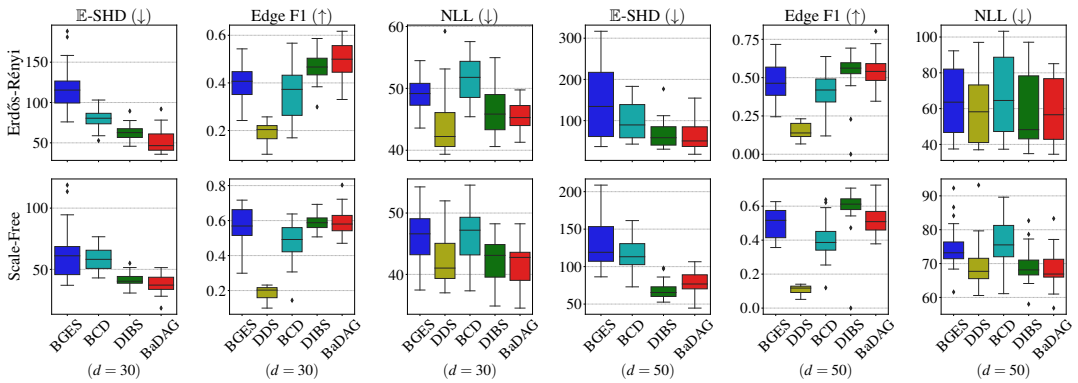

Figure 2: Posterior inference of both graph and nonlinear functional parameters on synthetic datasets of nonlinear causal models with $d = 30$ and $d = 50$ variables. `BayesDAG` gives best results across most metrics and outperforms other permutation based approaches (BCD and DDS). ↓ denotes lower is better and ↑ denotes higher is better.

mance on 30 random datasets.

Table 1: $\mathbb{E}$-SHD (with 95% CI) for ER graphs in higher dimensional nonlinear causal models. DIBS becomes computationally prohibitive for $d > 50$.

	$d = 70$	$d = 100$
BGES	355.77 ± 18.02	563.02 ± 27.21
BCD	217.05 ± 9.58	362.66 ± 29.18
DIBS	N/A	N/A
BaDAG	$\mathbf{143.70 \pm 11.61}$	$\mathbf{295.92 \pm 24.67}$

Our approach is the first to attempt full posterior inference in nonlinear models using permutation-based inference. Results for $d = 30$ variables in Figure 2 demonstrate that `BayesDAG` significantly outperforms other *permutation-based approaches* and DIBS in most of the metrics. For $d = 50$, `BayesDAG` performs comparably to DIBS in ER but a little worse in SF. However, our method achieves better NLL on held-out data compared to most baselines including DIBS for $d = 30, 50$, ER and SF settings.

We additionally evaluate on $d \in \{70, 100\}$ variables. We find that our method consistently outperforms the baselines with $d = 70$ and in terms of $\mathbb{E}$-SHD with $d = 100$. Full results are presented in Appendix G.1. Competitive performance for $d > 50$ in nonlinear settings further demonstrates the applicability and computational efficiency of the proposed approach. In contrast, the only fully Bayesian nonlinear method, DIBS, is not computationally efficient to run for $d > 50$.

5.2. Applications

Evaluation on Semi-Synthetic Data We evaluate our method on the SynTReN simulator (Van den Bulcke et al., 2006). This simulator creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximates real experimental data. Table 2 presents the results of all the methods. We find that our method recovers the true network much better in terms of $\mathbb{E}$-SHD as well as Edge F1 compared to baselines.

Table 2: Results (with 95% confidence intervals) on Syntren (semi-synthetic) and Sachs Protein Cells (real-world) datasets. ↓ denotes lower is better and ↑ denotes higher is better.

	Syntren ($d = 20$)		Sachs Protein Cells ($d = 11$)	
	$\mathbb{E}$-SHD (↓)	Edge F1 (↑)	$\mathbb{E}$-SHD (↓)	Edge F1 (↑)
BGES	66.18 ± 9.47	$\mathbf{0.21 \pm 0.05}$	$\mathbf{16.61 \pm 0.44}$	0.22 ± 0.02
DDS	134.37 ± 4.58	0.13 ± 0.02	34.90 ± 0.73	0.21 ± 0.02
BCD	38.38 ± 7.12	0.15 ± 0.07	17.05 ± 1.93	0.20 ± 0.08
DIBS	46.43 ± 4.12	0.16 ± 0.02	22.3 ± 0.31	0.20 ± 0.01
BaDAG	$\mathbf{34.21 \pm 2.82}$	$\mathbf{0.20 \pm 0.02}$	18.92 ± 1.0	$\mathbf{0.26 \pm 0.04}$

Evaluation on Real Data We also evaluate on a real dataset which measures the expression level of different proteins and phospholipids in human cells (called the Sachs Protein Cells Dataset) (Sachs et al., 2005). The data corresponds to a network of protein-protein interactions of 11 different proteins with 17 edges in total among them. There are 853 observational samples in total, from which we bootstrap 800 samples of 5 different datasets. Results in Table 2 demonstrate that our method performs well as compared to the baselines, proving the suitability of the proposed sampling framework to real-world settings.

6. Discussion

In this work, we propose `BayesDAG`, a novel, scalable Bayesian causal discovery framework that employs SG-MCMC (and VI) to infer causal models. We establish the validity of performing Bayesian inference in the augmented $(\boldsymbol{W}, \boldsymbol{p})$ space and demonstrate its connection to permutation-based DAG learning. Our method offers direct DAG sampling in nonlinear models while demonstrating superior inference accuracy.

References

Adams, R. P. and Zemel, R. S. Ranking via sinkhorn propagation. *arXiv preprint arXiv:1106.1925*, 2011.

Agrawal, R., Squires, C., Yang, K., Shanmugam, K., and Uhler, C. Abcd-strategy: Budgeted experimental design for targeted causal structure discovery. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3400–3409. PMLR, 2019.

Annadani, Y., Rothfuss, J., Lacoste, A., Scherrer, N., Goyal, A., Bengio, Y., and Bauer, S. Variational causal networks: Approximate bayesian inference over causal structures. *arXiv preprint arXiv:2106.07635*, 2021.

Annadani, Y., Tigas, P., Ivanova, D. R., Jesson, A., Gal, Y., Foster, A., and Bauer, S. Differentiable multi-target causal bayesian experimental design. *arXiv preprint arXiv:2302.10607*, 2023.

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Bengio, Y., Lahlou, S., Deleu, T., Hu, E. J., Tiwari, M., and Bengio, E. Gflownet foundations. *arXiv preprint arXiv:2111.09266*, 2021.

Blondel, M., Teboul, O., Berthet, Q., and Djolonga, J. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pp. 950–959. PMLR, 2020.

Casella, G. and George, E. I. Explaining the gibbs sampler. *The American Statistician*, 46(3):167–174, 1992.

Charpentier, B., Kibler, S., and Günnemann, S. Differentiable dag sampling. *arXiv preprint arXiv:2203.08509*, 2022.

Chickering, D. M. Optimal structure identification with greedy search. *Journal of machine learning research*, 3 (Nov):507–554, 2002.

Cooper, G. F. and Herskovits, E. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9:309–347, 1992.

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

Deleu, T., Góis, A., Emezue, C., Rankawat, M., Lacoste-Julien, S., Bauer, S., and Bengio, Y. Bayesian structure learning with generative flow networks. In *Uncertainty in Artificial Intelligence*, pp. 518–528. PMLR, 2022.

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

Erdős, P., Rényi, A., et al. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci*, 5(1):17–60, 1960.

Friedman, N. and Koller, D. Being bayesian about network structure. a bayesian approach to structure discovery in bayesian networks. *Machine learning*, 50(1):95–125, 2003.

Friedman, N., Goldszmidt, M., and Wyner, A. Data analysis with bayesian networks: A bootstrap approach. *arXiv preprint arXiv:1301.6695*, 2013.

Geffner, T., Antoran, J., Foster, A., Gong, W., Ma, C., Kiciman, E., Sharma, A., Lamb, A., Kukla, M., Pawlowski, N., et al. Deep end-to-end causal inference. *arXiv preprint arXiv:2202.02195*, 2022.

Gong, W. *Advances in approximate inference: combining VI and MCMC and improving on Stein discrepancy*. PhD thesis, University of Cambridge, 2022.

Gong, W., Tschiatschek, S., Nowozin, S., Turner, R. E., Hernández-Lobato, J. M., and Zhang, C. Icebreaker: Element-wise efficient information acquisition with a bayesian deep latent gaussian model. *Advances in neural information processing systems*, 32, 2019.

Grzegorczyk, M. and Husmeier, D. Improving the structure mcmc sampler for bayesian networks by introducing a new edge reversal move. *Machine Learning*, 71(2-3):265, 2008.

Heckerman, D., Meek, C., and Cooper, G. A bayesian approach to causal discovery. *Innovations in Machine Learning: Theory and Applications*, pp. 1–28, 2006.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kuhn, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Kuipers, J. and Moffa, G. Partition mcmc for inference on acyclic digraphs. *Journal of the American Statistical Association*, 112(517):282–299, 2017.

Li, C., Chen, C., Carlson, D., and Carin, L. Preconditioned stochastic gradient langevin dynamics for deep neural networks. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

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

Ma, Y.-A., Chen, T., and Fox, E. A complete recipe for stochastic gradient mcmc. *Advances in neural information processing systems*, 28, 2015.

Madigan, D., York, J., and Allard, D. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pp. 215–232, 1995.

Mena, G., Belanger, D., Linderman, S., and Snoek, J. Learning latent permutations with gumbel-sinkhorn networks. *arXiv preprint arXiv:1802.08665*, 2018.

Munkres, J. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957.

Murphy, K. P. Active learning of causal bayes net structure. Technical report, technical report, UC Berkeley, 2001.

Niculae, V., Martins, A., Blondel, M., and Cardie, C. Sparsemap: Differentiable sparse structured inference. In *International Conference on Machine Learning*, pp. 3799–3808. PMLR, 2018.

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

Pearl, J. *Causality*. Cambridge university press, 2009.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Pe'er, D., Regev, A., Elidan, G., and Friedman, N. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17(suppl_1):S215–S224, 2001.

Robinson, R. W. Counting labeled acyclic digraphs. *New directions in the theory of graphs*, pp. 239–273, 1973.

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.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, prediction, and search*. MIT press, 2000.

Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. Bayesian optimization with robust bayesian neural networks. *Advances in neural information processing systems*, 29, 2016.

Tigas, P., Annadani, Y., Jesson, A., Schölkopf, B., Gal, Y., and Bauer, S. Interventions, where and how? experimental design for causal models at scale. *arXiv preprint arXiv:2203.02016*, 2022.

Tong, S. and Koller, D. Active learning for structure in bayesian networks. In *International joint conference on artificial intelligence*, volume 17, pp. 863–869. Citeseer, 2001.

Trippe, B. and Turner, R. Overpruning in variational bayesian neural networks. *arXiv preprint arXiv:1801.06230*, 2018.

Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B., and Marchal, K. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics*, 7:1–12, 2006.

Van Koten, C. and Gray, A. An application of bayesian network for predicting object-oriented software maintainability. *Information and Software Technology*, 48(1): 59–67, 2006.

Yu, Y., Gao, T., Yin, N., and Ji, Q. Dags with no curl: An efficient dag structure learning approach. In *International Conference on Machine Learning*, pp. 12156–12166. PMLR, 2021.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Zheng, X., Aragam, B., Ravikumar, P. K., and Xing, E. P. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Appendix – BayesDAG: Gradient-Based Posterior Sampling for Causal Discovery

A. Joint Inference with SG-MCMC

In this section, we propose an alternative formulation that enables a joint inference framework for $\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}$ using SG-MCMC, thereby avoiding the need for variational inference for $\boldsymbol{W}$.

We adopt a continuous relaxation of $\boldsymbol{W}$, similar to (Lorch et al., 2021), by introducing a latent variable $\tilde{\boldsymbol{W}}$. The graphical model is illustrated in Figure 3. We can define

$$p(\boldsymbol{W}|\tilde{\boldsymbol{W}}) = \prod_{i,j} p(W_{ij}|\tilde{W}_{ij}) \tag{10}$$

with $p(W_{ij} = 1|\tilde{W}_{ij}) = \sigma(\tilde{W}_{ij})$ where $\sigma(\cdot)$ is the sigmoid function. In other words, $\tilde{W}ij$ defines the existence logits of Wij.

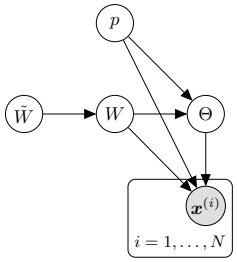

Figure 3: Graphical model with latent variable $\tilde{\boldsymbol{W}}$.

With the introduction of $\tilde{\boldsymbol{W}}$, the original posterior expectations of $p(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}|\boldsymbol{D})$, e.g. during evaluation, can be translated using the following proposition.

Proposition A.1 (Equivalence of posterior expectation)**.** *Under the generative model Figure 3, we have*

$$\mathbb{E}_{p(\boldsymbol{p},\boldsymbol{W},\boldsymbol{\Theta}|\boldsymbol{D})}\left[\boldsymbol{f}(\boldsymbol{G} = \tau(\boldsymbol{p}, \boldsymbol{W}), \boldsymbol{\Theta})\right] = \mathbb{E}_{p(\boldsymbol{p},\tilde{\boldsymbol{W}},\boldsymbol{\Theta})}\left[\frac{\mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})}\left[f(\boldsymbol{G}, \boldsymbol{\Theta})p(\boldsymbol{D}, \boldsymbol{\Theta}|\boldsymbol{p}, \boldsymbol{W})\right]}{\mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})}\left[p(\boldsymbol{D}, \boldsymbol{\Theta}|\boldsymbol{p}, \boldsymbol{W})\right]}\right] \tag{11}$$

where $\boldsymbol{f}$ is the target quantity.

This proof is in Appendix B.3.

With this proposition, instead of sampling $\boldsymbol{W}$, use SG-MCMC to draw $\tilde{\boldsymbol{W}}$ samples. Similar to Section 4.1, to use SG-MCMC for $\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta}$, we need their gradient information. The following proposition specifies the required gradients.

Proposition A.2. *With the generative model defined as Figure 3, we have*

$$\begin{aligned}\nabla_{\boldsymbol{p},\boldsymbol{\Theta},\tilde{\boldsymbol{W}}} U(\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta}) = &- \nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}) - \nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta}) \\ &- \nabla_{\tilde{\boldsymbol{W}}} \log p(\tilde{\boldsymbol{W}}) - \nabla_{\boldsymbol{p},\boldsymbol{\Theta},\tilde{\boldsymbol{W}}} \log \mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})}[p(\boldsymbol{D}|\boldsymbol{W}, \boldsymbol{p}, \boldsymbol{\Theta})]\end{aligned} \tag{12}$$

The proof is in Appendix B.4.

With these gradients, we can directly plug in existing SG-MCMC samplers to draw samples for $\boldsymbol{p}, \tilde{\boldsymbol{W}}$, and $\boldsymbol{\Theta}$ in joint inference (Algorithm 1).

B. Theory

B.1. Proof of Theorem 3.1

For completeness, we recite the theorem here.

Theorem 3.1 (Equivalence of inference in $(\boldsymbol{W}, \boldsymbol{p})$ and binary DAG space). Assume graph $\boldsymbol{G}$ is a binary adjacency matrix representing a DAG and node potential $\boldsymbol{p}$ does not contain the same values, i.e. $p_i \neq p_j \ \forall i, j$. Then, with the induced joint observational distribution $p(\boldsymbol{D}, \boldsymbol{G})$, dataset $\boldsymbol{D}$ and a corresponding prior $p(\boldsymbol{G})$, we have

$$p(\boldsymbol{G}|\boldsymbol{D}) = \int p_\tau(\boldsymbol{p}, \boldsymbol{W}|\boldsymbol{D})\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}d\boldsymbol{p} \tag{13}$$

if $p(\boldsymbol{G}) = \int p_\tau(\boldsymbol{p}, \boldsymbol{W})\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}d\boldsymbol{p}$, where $p_\tau(\boldsymbol{W}, \boldsymbol{p})$ is the prior, $\mathbb{1}(\cdot)$ is the indicator function and $p_\tau(\boldsymbol{p}, \boldsymbol{W}|D)$ is the posterior distribution over $\boldsymbol{p}, \boldsymbol{W}$.

Algorithm 1 Joint inference

Input: dataset $\boldsymbol{D}$, prior $p(\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta})$, SG-MCMC sampler update $\mathrm{Sampler}(\cdot)$; sampler hyperparameter Ψ; training steps T.
Output: posterior samples $\{\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta}\}$
Initialize $\boldsymbol{p}_0, \tilde{\boldsymbol{W}}_0, \boldsymbol{\Theta}_0$
for $t = 1, \ldots, T$ **do**
  Evaluate gradient $\nabla_{\boldsymbol{p}_{t-1}, \tilde{\boldsymbol{W}}_{t-1}, \boldsymbol{\Theta}_{t-1}} U$ based on Equation (12).
  Update samples $\boldsymbol{p}_t, \tilde{\boldsymbol{W}}_t, \boldsymbol{\Theta}_t = \mathrm{Sampler}(\nabla_{\boldsymbol{p}_{t-1}, \tilde{\boldsymbol{W}}_{t-1}, \boldsymbol{\Theta}_{t-1}} U; \Psi)$
  if storing condition met **then**
    $\{\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta}\} \leftarrow \boldsymbol{p}_t, \tilde{\boldsymbol{W}}_t, \boldsymbol{\Theta}_t$
  end if
end for

Algorithm 2 `BayesDAG` SG-MCMC+VI Inference

Input: dataset $\boldsymbol{D}$; prior $p(\boldsymbol{p}, \boldsymbol{W}), p(\boldsymbol{\Theta})$; SG-MCMC sampler $\mathrm{Sampler}$; sampler hyperparameters Ψ; network $\mu_\phi(\cdot)$; training iteration T.
Output: samples $\{\boldsymbol{\Theta}, \boldsymbol{p}\}$ and variational posterior q_ϕ
Initialize $\boldsymbol{\Theta}^{(0)}, \boldsymbol{p}^{(0)}, \phi$
for $t = 1 \ldots, T$ **do**
  Sample $\boldsymbol{W}^{(t-1)} \sim q_\phi(\boldsymbol{W}|\boldsymbol{p}^{(t-1)})$
  Evaluate $\nabla_{\boldsymbol{p}, \boldsymbol{\Theta}} U$ (Equations (24) and (25)) with $\boldsymbol{\Theta}^{(t-1)}, \boldsymbol{p}^{(t-1)}, \boldsymbol{W}^{(t-1)}$
  $\boldsymbol{\Theta}^{(t)}, \boldsymbol{p}^{(t)} = \mathrm{Sampler}(\nabla_{\boldsymbol{p}, \boldsymbol{\Theta}} U; \Psi)$
  if storing condition met **then**
    $\{\boldsymbol{p}, \boldsymbol{\Theta}\} \leftarrow \boldsymbol{p}^{(t)}, \boldsymbol{\Theta}^{(t)}$
  end if
  Maximize ELBO (Equation (9)) w.r.t. ϕ with $\boldsymbol{p}^{(t)}, \boldsymbol{\Theta}^{(t)}$
end for

To prove this theorem, we first prove the following lemma stating the equivalence of τ (Equation (3)) to binary DAG space.

Lemma B.1 (Equivalence of τ to DAG space)**.** *Consider d random variables, a node potential vector $\boldsymbol{p} \in \mathbb{R}^d$ and a binary matrix $\boldsymbol{W} \in \{0, 1\}^{d \times d}$. Then the following holds:*

(a) For any $\boldsymbol{W} \in \{0, 1\}^{d \times d}$, $\boldsymbol{p} \in \mathbb{R}^d$, $\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p})$ is a DAG.

(b) For any DAG $\boldsymbol{G} \in \mathbb{D}$, where $\mathbb{D}$ is the space of all DAGs, there exists a corresponding $(\boldsymbol{W}, \boldsymbol{p})$ such that $\tau(\boldsymbol{W}, \boldsymbol{p}) = \boldsymbol{G}$.

Proof. The main proof directly follows the theorem 2.1 in (Yu et al., 2021). For (a), we show the output from $\tau(\boldsymbol{W}, \boldsymbol{p})$ must be a DAG. By leveraging the Lemma 3.4 in (Yu et al., 2021), we can easily obtain that $\mathrm{Step}(\mathrm{grad}\, \boldsymbol{p})$ emits a binary adjacency matrix representing a DAG. The only difference is that we replace the $\mathrm{ReLU}(\cdot)$ with $\mathrm{Step}(\cdot)$ but the conclusion can be directly generalized.

For (b), we show that for any DAG $\boldsymbol{G}$, there exists a $(\boldsymbol{W}, \boldsymbol{p})$ pair s.t. $\tau(\boldsymbol{W}, \boldsymbol{p}) = \boldsymbol{G}$. To see this, we can observe that $\boldsymbol{p}$ implicitly defines a topological order in the mapping τ. For any $p_i > p_j$, we have $j \to i$ after the mapping $\mathrm{Step}(\mathrm{grad}\, \boldsymbol{p})$. Thus, by leveraging Theorem 3.7 in (Yu et al., 2021), we obtain that there exists a potential vector $\boldsymbol{p} \in \mathbb{R}^d$ for any DAG $\boldsymbol{G}$ such that

$$(\mathrm{grad}\, \boldsymbol{p})(i, j) > 0 \quad \text{when } G_{ij} = 1$$

Thus, we can choose $\boldsymbol{W}$ in the following way:

$$\boldsymbol{W} = \begin{cases} W_{ij} = 0 & \text{if } G_{ij} = 0 \\ W_{ij} = 1 & \text{if } G_{ij} = 1 \end{cases}$$

□

Next, let's prove the Theorem 3.1.

Proof of Theorem 3.1. From Lemma B.1, we see that the mapping is complete. Namely, the $(\boldsymbol{W}, \boldsymbol{p})$ space can represent the entire DAG space. Next, we show that performing Bayesian inference in $(\boldsymbol{W}, \boldsymbol{p})$ space can also correspond to the inference in DAG space.

Assume we have the prior $p_\tau(\boldsymbol{W}, \boldsymbol{p})$. Then through mapping τ, we implicitly define a prior over the DAG $\boldsymbol{G}$ in the following:

$$p_\tau(\boldsymbol{G}) = \int p_\tau(\boldsymbol{W}, \boldsymbol{p})\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}d\boldsymbol{p} \tag{14}$$

This basically states that the corresponding prior over $\boldsymbol{G}$ is an accumulation of the corresponding probability associated with $(\boldsymbol{W}, \boldsymbol{p})$ pairs.

Similarly, we can define a corresponding posterior $p_\tau(\boldsymbol{G}|\boldsymbol{D})$:

$$p_\tau(\boldsymbol{G}|\boldsymbol{D}) = \int p_\tau(\boldsymbol{W}, \boldsymbol{p}|\boldsymbol{D})\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}d\boldsymbol{p} \tag{15}$$

Now, let's show that this posterior $p_\tau(\boldsymbol{G}|\boldsymbol{D}) = p(\boldsymbol{G}|\boldsymbol{D})$ if prior matches, i.e. $p(\boldsymbol{G}) = p_\tau(\boldsymbol{G})$. From Bayes's rule, we can easily write down

$$p_\tau(\boldsymbol{W}, \boldsymbol{p}|\boldsymbol{D}) = \frac{p(\boldsymbol{D}|\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))p(\boldsymbol{p}, \boldsymbol{W})}{\sum_{\boldsymbol{G}' \in \mathbb{D}} p(\boldsymbol{D}, \boldsymbol{G}')} \tag{16}$$

Then, by substituting Equation (16) into Equation (15), we have

$$\begin{aligned}
p_\tau(\boldsymbol{G}|\boldsymbol{D}) &= \int \frac{p(\boldsymbol{D}|\boldsymbol{G})p_\tau(\boldsymbol{W}, \boldsymbol{p})}{\sum_{\boldsymbol{G}' \in \mathbb{D}} p(\boldsymbol{D}, \boldsymbol{G}')}\, \mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))d\boldsymbol{W}d\boldsymbol{p} \\
&= \frac{\int p(\boldsymbol{D}|\boldsymbol{G})p_\tau(\boldsymbol{W}, \boldsymbol{p})\, \mathbb{1}(\boldsymbol{G} = \tau)d\boldsymbol{W}d\boldsymbol{p}}{\sum_{\boldsymbol{G}' \in \mathbb{D}} p(\boldsymbol{D}, \boldsymbol{G}')} && (17) \\
&= \frac{p(\boldsymbol{D}|\boldsymbol{G}) \int p_\tau(\boldsymbol{W}, \boldsymbol{p})\, \mathbb{1}(\boldsymbol{G} = \tau)d\boldsymbol{W}d\boldsymbol{p}}{\sum_{\boldsymbol{G}' \in \mathbb{D}} p(\boldsymbol{D}, \boldsymbol{G}')} && (18) \\
&= \frac{p(\boldsymbol{D}|\boldsymbol{G})p_\tau(\boldsymbol{G})}{\sum_{\boldsymbol{G}' \in \mathbb{D}} p(\boldsymbol{D}|\boldsymbol{G}')p_\tau(\boldsymbol{G}')} \\
&= p(\boldsymbol{G}|\boldsymbol{D}) && (19)
\end{aligned}$$

where Equation (17) is from the fact that $\sum_{\boldsymbol{G}' \in \mathbb{D}} p(\boldsymbol{D}, \boldsymbol{G}')$ is independent of $(\boldsymbol{W}, \boldsymbol{p})$ due to marginalization. Equation (18) is obtained because $p(\boldsymbol{D}|\boldsymbol{G})$ is also independent of $(\boldsymbol{W}, \boldsymbol{p})$ due to (1) $\mathbb{1}(\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}))$ and (2) $p(\boldsymbol{D}|\boldsymbol{G})$ is a constant when fixing $\boldsymbol{G}$. Equation (19) is obtained by applying Bayes's rule and $p_\tau(\boldsymbol{G}) = p(\boldsymbol{G})$. □

B.2. Proof of Theorem 3.2

Theorem 3.2 (Equivalence of NoCurl formulation). Assuming the conditions in Theorem 3.1 are satisfied. Then, for a given $(\boldsymbol{W}, \boldsymbol{p})$, we have

$$\boldsymbol{G} = \boldsymbol{W} \odot \mathrm{Step}(\mathrm{grad}\, \boldsymbol{p}) = \boldsymbol{W} \odot \left[\boldsymbol{\sigma}^*(\boldsymbol{p})\boldsymbol{L}\boldsymbol{\sigma}^*(\boldsymbol{p})^T\right]$$

where $\boldsymbol{G}$ is a DAG and $\boldsymbol{\sigma}^*(\boldsymbol{p})$ is defined in Equation (5).

To prove this theorem, we need to first prove the following lemma.

Lemma B.2. *For any permutation matrix $\boldsymbol{M} \in \boldsymbol{\Sigma}_d$, we have*

$$\mathrm{grad}(\boldsymbol{M}\boldsymbol{p}) = \boldsymbol{M}^T \mathrm{grad}(p)\boldsymbol{M}$$

where grad *is the operator defined in* **??**.

Proof. By definition of $\mathrm{grad}(\cdot)$, we have

$$\begin{aligned}\mathrm{grad}(\boldsymbol{M}\boldsymbol{p}) &= (\boldsymbol{M}\boldsymbol{p})_i - (\boldsymbol{M}\boldsymbol{p})_j \\ &= \mathbf{1}(i)^T\boldsymbol{M}\boldsymbol{p} - \mathbf{1}(j)^T\boldsymbol{M}\boldsymbol{p} \\ &= \boldsymbol{M}_{i,:}\boldsymbol{p} - \boldsymbol{M}_{j,:}\boldsymbol{p}\end{aligned}$$

where $\mathbf{1}(i)$ is a one-hot vector with i^{th} entry 1, and $\boldsymbol{M}_{i,:}$ is the i^{th} row of matrix $\boldsymbol{M}$. The above is equivalent to computing the grad with new labels obtained by permuting $\boldsymbol{p}$ with $\boldsymbol{M}$. Therefore, we can see that $\mathrm{grad}(\boldsymbol{M}\boldsymbol{p})$ can be computed by permuting the original $\mathrm{grad}(\boldsymbol{p})$ by matrix $\boldsymbol{M}$.

$$\mathrm{grad}(\boldsymbol{M}\boldsymbol{p}) = \boldsymbol{M}^T\,\mathrm{grad}(\boldsymbol{p})\boldsymbol{M}$$

□

Proof of Theorem 3.2. Since $\boldsymbol{W}$ plays the same role in both formulations, we focus on the equivalence of $\mathrm{Step}(\mathrm{grad}(\cdot))$.

Define a sorted $\tilde{\boldsymbol{p}} = \boldsymbol{\sigma}\boldsymbol{p}$, where $\boldsymbol{\sigma} \in \boldsymbol{\Sigma}_d$, such that for $i < j$, we have $\tilde{p}_i > \tilde{p}_j$. Namely, $\boldsymbol{\sigma}$ is a permutation matrix. Thus, we have

$$\mathrm{grad}(\boldsymbol{p}) = \mathrm{grad}(\boldsymbol{\sigma}^T\tilde{\boldsymbol{p}}).$$

By Lemma B.2, we have

$$\mathrm{grad}(\boldsymbol{\sigma}^T\tilde{\boldsymbol{p}}) = \boldsymbol{\sigma}\,\mathrm{grad}(\tilde{\boldsymbol{p}})\boldsymbol{\sigma}^T.$$

Since $\tilde{\boldsymbol{p}}$ is an ordered vector. Therefore, $\mathrm{grad}(\tilde{\boldsymbol{p}})$ is a skew-symmetric matrix with a positive lower half part.

Therefore, we have

$$\mathrm{Step}(\mathrm{grad}(\boldsymbol{p})) = \mathrm{Step}(\boldsymbol{\sigma}\,\mathrm{grad}(\tilde{\boldsymbol{p}})\boldsymbol{\sigma}^T) = \boldsymbol{\sigma}\,\mathrm{Step}(\mathrm{grad}(\tilde{p}))\boldsymbol{\sigma}^T = \boldsymbol{\sigma}\boldsymbol{L}\boldsymbol{\sigma}^T$$

This is true because $\boldsymbol{\sigma}$ is just a permutation matrix that does not alter the sign of $\mathrm{grad}(\tilde{\boldsymbol{p}})$.

Since $\boldsymbol{\sigma}$ is a permutation matrix that sort $\boldsymbol{p}$ value in a ascending order, from Lemma 1 in (Blondel et al., 2020), we have

$$\boldsymbol{\sigma} = \underset{\boldsymbol{\sigma}' \in \boldsymbol{\Sigma}_d}{\arg\max}\, \boldsymbol{p}^T(\boldsymbol{\sigma}'\boldsymbol{o})$$

□

B.3. Proof of Proposition A.1

Proof.

$$\begin{aligned}&\mathbb{E}_{p(\boldsymbol{p},\boldsymbol{W},\boldsymbol{\Theta}|\boldsymbol{D})}\left[f(\boldsymbol{G}=\tau(\boldsymbol{p},\boldsymbol{W}),\boldsymbol{\Theta})\right] \\ =&\int p(\boldsymbol{p},\boldsymbol{W},\boldsymbol{\Theta},\tilde{\boldsymbol{W}}|\boldsymbol{D})f(\boldsymbol{G},\boldsymbol{\Theta})d\boldsymbol{p}d\boldsymbol{W}d\boldsymbol{\Theta}d\tilde{\boldsymbol{W}} \\ =&\int p(\boldsymbol{p},\tilde{\boldsymbol{W}},\boldsymbol{\Theta}|\boldsymbol{D})p(\boldsymbol{W}|\boldsymbol{p},\boldsymbol{\Theta},\tilde{\boldsymbol{W}},\boldsymbol{D})f(\boldsymbol{G},\boldsymbol{\Theta})d\boldsymbol{p}d\boldsymbol{W}d\boldsymbol{\Theta}d\tilde{\boldsymbol{W}} \\ =&\mathbb{E}_{p(\boldsymbol{p},\tilde{\boldsymbol{W}},\boldsymbol{\Theta}|\boldsymbol{D})}\left[\frac{\int p(\boldsymbol{D}|\boldsymbol{p},\boldsymbol{\Theta},\boldsymbol{W})p(\boldsymbol{p})p(\tilde{\boldsymbol{W}})p(\boldsymbol{W}|\tilde{\boldsymbol{W}})p(\boldsymbol{\Theta}|\boldsymbol{p},\boldsymbol{W})f(\boldsymbol{G},\boldsymbol{\Theta})d\boldsymbol{W}}{\int p(\boldsymbol{D}|\boldsymbol{p},\boldsymbol{\Theta},\boldsymbol{W})p(\boldsymbol{p})p(\tilde{\boldsymbol{W}})p(\boldsymbol{W}|\tilde{\boldsymbol{W}})p(\boldsymbol{\Theta}|\boldsymbol{p},\boldsymbol{W})d\boldsymbol{W}}\right] \\ =&\mathbb{E}_{p(\boldsymbol{p},\tilde{\boldsymbol{W}},\boldsymbol{\Theta})}\left[\frac{\mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})}\left[f(\boldsymbol{G},\boldsymbol{\Theta})p(\boldsymbol{D},\boldsymbol{\Theta}|\boldsymbol{p},\boldsymbol{W})\right]}{\mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})}\left[p(\boldsymbol{D},\boldsymbol{\Theta}|\boldsymbol{p},\boldsymbol{W})\right]}\right]\end{aligned}$$

□

B.4. Proof of Proposition A.2

Proof.

$$\begin{aligned}
\nabla_{\boldsymbol{p}} U(\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta}) &= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}, \tilde{\boldsymbol{W}}, \boldsymbol{\Theta}, \boldsymbol{D}) \\
&= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}) - \nabla_{\boldsymbol{p}} \log p(\tilde{\boldsymbol{W}}, \boldsymbol{\Theta}, \boldsymbol{D}|\boldsymbol{p}) \\
&= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}) - \frac{\nabla_{\boldsymbol{p}} \int p(\boldsymbol{D}|\boldsymbol{W}, \boldsymbol{p}, \boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{p}, \boldsymbol{W}) p(\boldsymbol{W}|\tilde{\boldsymbol{W}}) p(\tilde{\boldsymbol{W}}) d\boldsymbol{W}}{\int p(\boldsymbol{D}|\boldsymbol{W}, \boldsymbol{p}, \boldsymbol{\Theta}) p(\boldsymbol{\Theta}|\boldsymbol{p}, \boldsymbol{W}) p(\boldsymbol{W}|\tilde{\boldsymbol{W}}) p(\tilde{\boldsymbol{W}}) d\boldsymbol{W}} \\
&= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}) - \frac{\nabla_{\boldsymbol{p}} \mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})} \left[p(\boldsymbol{D}|\boldsymbol{W}, \boldsymbol{p}, \boldsymbol{\Theta}) \right]}{\mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})} \left[p(\boldsymbol{D}|\boldsymbol{W}, \boldsymbol{p}, \boldsymbol{\Theta}) \right]} \\
&= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}) - \nabla_{\boldsymbol{p}} \log \mathbb{E}_{p(\boldsymbol{W}|\tilde{\boldsymbol{W}})} \left[p(\boldsymbol{D}|\boldsymbol{W}, \boldsymbol{p}, \boldsymbol{\Theta}) \right]
\end{aligned}$$

Other gradient $\nabla_{\tilde{\boldsymbol{W}}} U$ and $\nabla_{\boldsymbol{\Theta}} U$ can be derived using the similar approach, which concludes the proof. □

B.5. Proof of Proposition C.1

Proof of Proposition C.1. By definition, we have easily have

$$\begin{aligned}
\nabla_{\boldsymbol{p}} U &= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}, \boldsymbol{D}) \\
&= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}, \boldsymbol{W}) - \nabla_{\boldsymbol{p}} \log p(\boldsymbol{D}, \boldsymbol{\Theta}|\tau(\boldsymbol{W}, \boldsymbol{p})) \\
&= -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}, \boldsymbol{W}) - \nabla_{\boldsymbol{p}} \log p(\boldsymbol{D}|\boldsymbol{\Theta}, \tau(\boldsymbol{W}, \boldsymbol{p})) + \underbrace{\nabla_{\boldsymbol{p}} \log p(\boldsymbol{\Theta}|\tau(\boldsymbol{p}, \boldsymbol{W}))}_{0}
\end{aligned}$$

Similarly, we have

$$\begin{aligned}
\nabla_{\boldsymbol{\Theta}} U &= -\nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}, \boldsymbol{D}) \\
&= -\nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{D}|\boldsymbol{\Theta}, \tau(\boldsymbol{W}, \boldsymbol{p})) - \nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta}|\boldsymbol{p}, \boldsymbol{W}) - \underbrace{\nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{p}, \boldsymbol{W})}_{0} \\
&= -\nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{D}|\boldsymbol{\Theta}, \tau(\boldsymbol{W}, \boldsymbol{p})) - \nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta})
\end{aligned}$$

□

B.6. Derivation of ELBO

$$\begin{aligned}
\log p(\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D}) &= \log \int p(\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D}, \boldsymbol{W}) d\boldsymbol{W} \\
&= \log \int \frac{q_\phi(\boldsymbol{W}|\boldsymbol{p})}{q_\phi(\boldsymbol{W}|\boldsymbol{p})} p(\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D}, \boldsymbol{W}) d\boldsymbol{W} \\
&\geq \int q_\phi(\boldsymbol{W}|\boldsymbol{p}) \log p(\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D}|\boldsymbol{W}) d\boldsymbol{W} + \int q_\phi(\boldsymbol{W}|\boldsymbol{p}) \log \frac{p(\boldsymbol{W})}{q_\phi(\boldsymbol{W}|\boldsymbol{p})} d\boldsymbol{W} \qquad (20) \\
&= \mathbb{E}_{q_\phi(\boldsymbol{W}|\boldsymbol{p})} \left[\log p(\boldsymbol{p}, \boldsymbol{\Theta}, \boldsymbol{D}|\boldsymbol{W}) \right] - D_{\mathrm{KL}} \left[q_\phi(\boldsymbol{W}|\boldsymbol{p}) \| p(\boldsymbol{W}) \right]
\end{aligned}$$

where the Equation (20) is obtained by Jensen's inequality.

C. SG-MCMC Update

Assume we want to draw samples $\boldsymbol{p} \sim p(\boldsymbol{p}|\boldsymbol{D}, \boldsymbol{W}, \boldsymbol{\Theta}) \propto \exp(-U(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}))$ with $U(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) = -\log p(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta})$, we can compute U by

$$U(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) = -\sum_{n=1}^{N} \log p(\boldsymbol{x}_n|\boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}), \boldsymbol{\Theta}) - \log p(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) \qquad (21)$$

In practice, we typically use mini-batches $\mathcal{S}$ instead of the entire dataset $\boldsymbol{D}$. Therefore, an approximation is

$$\tilde{U}(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) = -\frac{|\boldsymbol{D}|}{|\mathcal{S}|} \sum_{n \in \mathcal{S}} \log p(\boldsymbol{x}_n | \boldsymbol{G} = \tau(\boldsymbol{W}, \boldsymbol{p}), \boldsymbol{\Theta}) - \log p(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) \tag{22}$$

where $|\mathcal{S}|$ and $|\boldsymbol{D}|$ are the minibatch and dataset sizes, respectively.

(Gong et al., 2019) uses the preconditioning technique on *stochastic gradient Hamiltonian Monte Carlo*(SG-HMC), similar to the preconditioning technique in (Li et al., 2016). In particular, they use a moving-average approximation of diagonal Fisher information to adjust the momentum. The transition dynamics at step t with EM discretization is

$$\begin{aligned}
B &= \frac{1}{2} l \\
\boldsymbol{V}_t &= \beta_2 \boldsymbol{V}_{t-1} + (1 - \beta_2) \nabla_{\boldsymbol{p}} \tilde{U}(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) \odot \nabla_{\boldsymbol{p}} \tilde{U}(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) \\
g_t &= \frac{1}{\sqrt{1 + \sqrt{\boldsymbol{V}_t}}} \\
\boldsymbol{r}_t &= \beta_1 \boldsymbol{r}_{t-1} - l g_t \nabla_{\boldsymbol{p}} \tilde{U}(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta}) + l \frac{\partial g_t}{\partial \boldsymbol{p}_t} + s \sqrt{2l(\frac{1 - \beta_1}{l} - B)} \eta \\
\boldsymbol{p}_t &= \boldsymbol{p}_{t-1} + l g_t \boldsymbol{r}_t
\end{aligned} \tag{23}$$

where l^2 is the learning rate; (β_1, β_2) controls the preconditioning decay rate, η is the Gaussian noise with 0 mean and unit variance, and s is the hyperparameter controlling the level of injected noise to SG-MCMC. Throughout the paper, we use $(\beta_1, \beta_2) = (0.9, 0.99)$ for all experiments.

The following proposition specifies the gradients required by SG-MCMC: $\nabla_{\boldsymbol{p}, \boldsymbol{\Theta}} U(\boldsymbol{p}, \boldsymbol{W}, \boldsymbol{\Theta})$.

Proposition C.1. *We have the following gradient equations for SG-MCMC:*

$$\nabla_{\boldsymbol{p}} U = -\nabla_{\boldsymbol{p}} \log p(\boldsymbol{p}) - \nabla_{\boldsymbol{p}} \log p(\boldsymbol{D} | \boldsymbol{\Theta}, \tau(\boldsymbol{W}, \boldsymbol{p})) \tag{24}$$

and

$$\nabla_{\boldsymbol{\Theta}} U = -\nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{\Theta}) - \nabla_{\boldsymbol{\Theta}} \log p(\boldsymbol{D} | \boldsymbol{\Theta}, \tau(\boldsymbol{p}, \boldsymbol{W})) \tag{25}$$

Refer to Appendix B.5 for details.

D. Gumbel-Sinkhorn Operator

The Sinkhorn operator $\mathcal{S}(\boldsymbol{M})$ on a matrix $\boldsymbol{M}$ (Adams & Zemel, 2011) is defined as a sequence of row and column normalizations, each is called Sinkhorn iteration. (Mena et al., 2018) showed that the non-differentiable $\arg\max$ problem

$$\boldsymbol{\sigma} = \arg\max_{\boldsymbol{\sigma}' \in \Sigma_d} \langle \boldsymbol{\sigma}', \boldsymbol{M} \rangle \tag{26}$$

can be relaxed through an entropy regularizer with its solution being expressed by $\mathcal{S}(\cdot)$. In particular, they showed that $\mathcal{S}(\boldsymbol{M}/t) = \arg\max_{\boldsymbol{\sigma}' \in \mathcal{B}_d} \langle \boldsymbol{\sigma}', \boldsymbol{M} \rangle + t h(\boldsymbol{\sigma}')$, where $h(\cdot)$ is the entropy function. This regularized solution converges to the solution of Equation (26) when $t \to 0$, i.e. $\lim_{t \to 0} \mathcal{S}(\boldsymbol{M}/t)$. Since the Sinkhorn operator is differentiable, $\mathcal{S}(\boldsymbol{M}/t)$ can be viewed as a differentiable approximation to Equation (26), which can be used to obtain the solution of Equation (5). Specifically, we have

$$\arg\max_{\boldsymbol{\sigma}' \in \boldsymbol{\Sigma}_d} \boldsymbol{p}^T (\boldsymbol{\sigma}' \boldsymbol{o}) = \arg\max_{\boldsymbol{\sigma}' \in \boldsymbol{\Sigma}_d} \langle \boldsymbol{\sigma}', \boldsymbol{p}\boldsymbol{o}^T \rangle = \lim_{t \to 0} \mathcal{S}(\frac{\boldsymbol{p}\boldsymbol{o}^T}{t}) \tag{27}$$

In practice, we approximate it wth $t > 0$, resulting in a doubly stochastic matrix. To get the binary permutation matrix, we apply the Hungarian algorithm (Munkres, 1957). During the backward pass, we use a straight-through estimator for $\boldsymbol{p}$.

Some of the previous works (Charpentier et al., 2022; Cundy et al., 2021) have leveraged the Sinkhorn operator to model variational distributions over permutation matrices. However, they start with a full rank $\boldsymbol{M}$, which has been reported to require over **1000** Sinkhorn iterations to converge (Cundy et al., 2021). However, our formulation, based on explicit node potential $\boldsymbol{p}\boldsymbol{o}^T$, generates a rank-1 matrix, requiring much fewer Sinkhorn steps (around **300**) in practice, saving two-thirds of the computational cost.

E. Related Work

Bayesian causal discovery literature has primarily focused on inference in linear models with closed-form posteriors or marginalized parameters. Early works considered sampling directed acyclic graphs (DAGs) for discrete (Cooper & Herskovits, 1992; Madigan et al., 1995; Heckerman et al., 2006) and Gaussian random variables (Friedman & Koller, 2003; Tong & Koller, 2001) using Markov chain Monte Carlo (MCMC) in the DAG space. However, these approaches exhibit slow mixing and convergence (Eaton & Murphy, 2012; Grzegorczyk & Husmeier, 2008), often requiring restrictions on number of parents (Kuipers & Moffa, 2017).

Recent advances in variational inference (Zhang et al., 2018) have facilitated graph inference in DAG space, with gradient-based methods employing the NOTEARS DAG penalty (Zheng et al., 2018).(Annadani et al., 2021) samples DAGs from autoregressive adjacency matrix distributions, while (Lorch et al., 2021) utilizes Stein variational approach (Liu & Wang, 2016) for DAGs and causal model parameters. (Cundy et al., 2021) proposed a variational inference framework on node orderings using the gumbel-sinkhorn gradient estimator (Mena et al., 2018). (Deleu et al., 2022; Nishikawa-Toomey et al., 2022) employ the GFlowNet framework (Bengio et al., 2021) for inferring the DAG posterior. Most methods, except(Lorch et al., 2021) are restricted to linear models, while (Lorch et al., 2021) has high computational costs and lacks DAG generation guarantees compared to our method.

In contrast, *quasi-Bayesian* methods, such as DAG bootstrap (Friedman et al., 2013), demonstrate competitive performance. DAG bootstrap resamples data and estimates a single DAG using PC (Spirtes et al., 2000), GES (Chickering, 2002), or similar algorithms, weighting the obtained DAGs by their unnormalized posterior probabilities. Recent neural network-based works employ variational inference to learn DAG distributions and point estimates for nonlinear model parameters (Charpentier et al., 2022; Geffner et al., 2022).

F. Experimental Settings

F.1. Baselines

For all the experimental settings, we compare with the following baselines:

- Bootstrap GES (**BGES**) (Friedman et al., 2013; Chickering, 2002) is a bootstrap based quasi-Bayesian approach for linear Gaussian models which first resamples with replacement data points at random and then estimates a linear SCM using the GES algorithm (Chickering, 2002) for each bootstrap set. GES is a score based approach to learn a point estimate of a linear Gaussian SCM. For all the experimental settings, we use 50 bootstrap sets.
- Differentiable DAG Sampling (**DDS**) is a VI based approach to learn distribution over DAGs and a point estimate over the nonlinear functional parameters. DDS performs inference on the node permutation matrices, thus directly generating DAGs. Gumbel-sinkhorn (Mena et al., 2018) is used for obtaining valid gradients and Hungarian algorithm is used for the straight-through gradient estimator. In the author provided implementation, for evaluation, a single permutation matrix is sampled and the logits of the edge beliefs are directly thresholded. In this work, in order to make the comaprison fair to Bayesian learning methods, we directly sample the binary adjacency matrix based on the edge logits.
- **BCD** Nets (Cundy et al., 2021) is a VI based fully Bayesian structure learning approach for linear causal models. BCD performs inference on both the node permutations through the Gumbel-sinkhorn (Mena et al., 2018) operator as well as the model parameters through a VI distribution. Both DDS and BCD nets operate directly on full rank initializations to the Gumbel-sinkhorn operator, unlike our rank-1 initialization, which saves computations in practice.
- **DIBS** (Lorch et al., 2021) uses SVGD (Liu & Wang, 2016) with the DAG regularizer (Zheng et al., 2018) and bilinear embeddings to perform inference over both linear and nonlinear causal models. As our data generation process involves SCM with unequal noise variance, we extend DIBS framework with an inference over noise variance using SVGD, similar to the original paper.

While DIBS and DDS can handle nonlinear parameterization, approaches like BGES and BCD, which are primarily designed for linear models still give competitive results when applied on nonlinear data. Given that there are limited number of baselines in the nonlinear case, and DIBS being the only fully Bayesian nonlinear baseline, we compare with BGES and BCD for all settings despite their model misspecification.

F.2. Evaluation Metrics

For higher dimensional settings with nonlinear models, the true posterior is intractable. While in general it is hard to evaluate the posterior inference quality in high dimensions, prior work has suggested to evaluate on proxy metrics which we adopt in this work as well (Lorch et al., 2021; Geffner et al., 2022; Annadani et al., 2023). In particular, we evaluate the following metrics:

- $\mathbb{E}$**-SHD**: Structural Hamming Distance (SHD) measures the hamming distance between graphs. In particular, it is a measure of number of edges that are to be added, removed or reversed to get the ground truth from the estimated graph. Since we have a posterior distribution $q(\boldsymbol{G})$ over graphs, we measure the *expected* SHD:

$$\mathbb{E}\text{-SHD} \coloneqq \mathbb{E}_{\boldsymbol{G}\sim q(\boldsymbol{G})}[\text{SHD}(\boldsymbol{G}, \boldsymbol{G}^{GT})] \approx \frac{1}{N_e}\sum_{i=1}^{N_e}[\text{SHD}(\boldsymbol{G}^{(i)}, \boldsymbol{G}^{GT})] \quad \text{, with} \quad \boldsymbol{G}^{(i)} \sim q(\boldsymbol{G})$$

  where $\boldsymbol{G}^{GT}$ is the ground-truth causal graph.

- **Edge F1**: It is F1 score of each edge being present or absent in comparison to the true edge set, averaged over all edges.

- **NLL**: We also measure the negative log-likelihood of the held-out data.

The first two metrics measure the goodness of the graph posterior while the NLL measures the goodness of the joint posterior over the entire causal model.

F.3. Synthetic Data

As knowledge of ground truth graph is not possible in many real world settings, it is standard across causal discovery to benchmark in synthetic data settings. Following prior work, we generate synthetic data by first sampling a DAG at random from either Erdos-Rènyi (ER) (Erdős et al., 1960) or Scale-Free (SF) (Barabási & Albert, 1999) family. We ensure that the graphs have $2d$ edges in expectation. For nonlinear models, the nonlinear functions are defined by randomly initialized Multi-Layer Perceptrons (MLP) with a single hidden layer of 5 nodes and ReLU nonlinearity. The variance of the exogenous Gaussian noise variable is drawn from an Inverse Gamma prior with concentration $\alpha = 1.5$ and rate $\beta = 1$. For higher dimensional settings, we consider $N = 5000$ random samples for training and $N = 1000$ samples for held-out evaluation. For all settings, we evaluate on 30 random datasets.

F.4. Hyperparameter Selection

In this section, we will give the details our how to select the hyperparameters for our method and all the baseline models.

We employ a cross-validation-like procedure for hyperparameter tuning in BayesDAG and DIBS to optimize $\mathbb{E}-$SHD value (for nonlinear setting). For each ER and SF dataset with varying dimensions, we initially generate five tuning datasets. After determining the optimal hyperparameters, we fix them and evaluate the models on 30 test datasets. For DDS, we adopt the hyperparameters provided in the original paper (Charpentier et al., 2022). BCD and BGES do not necessitate hyperparameter tuning since BCD already incorporates the correct prior graph for ER and SF datasets. For semi-synthetic Syntren and real world Sachs protein cells datasets, we assume the number of edges in the ground truth graphs are known and we tune our hyperparameters to produce roughly correct number of edges. BCD and DIBS also assume access to the ground truth edge number and use the graph prior to enforce the number of edges.

Network structure We use one hidden layer MLP with hidden size of $\max(4 * d, 64)$ for the nonlinear functional relations, where d is the dimensionalilty of dataset. We use **LeakyReLU** as the activation function. We also enable the **LayerNorm** and **residual connections** in the network. In particular, for variational network μ_ϕ in `BayesDAG`, we apply the **LayerNorm** on $\boldsymbol{p}$ before inputting it to the network. We use 2 hidden layer MLP with size 48, **LayerNorm** and **residual connections** for μ_ϕ.

Sparse initialization for `BayesDAG` For `BayesDAG`, we additionally allow sparse initialization by sampling a sparse $\boldsymbol{W}$ from the μ_ϕ. This can be achieved by substracting a constant 1 from the existing logits (i.e. the output from μ_ϕ).

BayesDAG				
	λ_s	Scale $\boldsymbol{p}$	Scale $\boldsymbol{\Theta}$	Sparse Init.
nonlinear ER $d = 20$	300	0.01	0.01	False
nonlinear SF $d = 20$	200	0.1	0.1	False
nonlinear ER $d = 30$	500	1	0.01	False
nonlinear SF $d = 30$	300	0.01	0.01	False
nonlinear ER $d = 50$	500	0.01	0.01	True
nonlinear SF $d = 50$	300	0.1	0.01	False
nonlinear ER $d = 70$	700	0.1	0.01	True
nonlinear SF $d = 70$	300	0.01	0.01	False
nonlinear ER $d = 100$	700	0.1	0.01	False
nonlinear SF $d = 100$	700	0.1	0.01	False
SynTren	300	0.1	0.01	False
Sachs Protein Cells	1200	0.1	0.01	False

Table 3: The hyperparameter selection for BayesDAG for each setting.

DIBS				
	α	h latent	h_θ	h_σ
nonlinear ER $d = 20$	0.02	5	1500	10
nonlinear SF $d = 20$	0.2	5	1500	10
nonlinear ER $d = 30$	0.2	5	500	1
nonlinear SF $d = 30$	0.2	5	1000	1
nonlinear ER $d = 50$	0.2	5	500	10
nonlinear SF $d = 50$	0.2	5	1500	1
SynTren	0.2	5	500	10
Sachs Protein Cells	0.2	5	500	10

Table 4: The hyperparameter selection for DIBS for each setting.

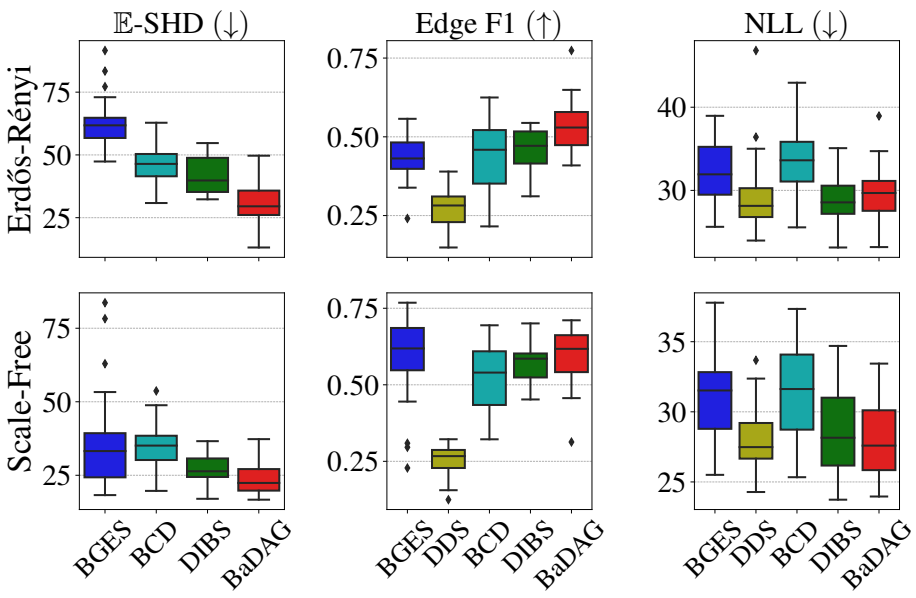

Figure 4: Posterior inference of both graph and functional parameters on synthetic datasets of nonlinear causal models with $d = 20$ variables. `BayesDAG` gives best results across all metrics. $\downarrow$ denotes lower is better and $\uparrow$ denotes higher is better. For the sake of clarity, DDS has been omitted for $\mathbb{E}$-SHD due to its significantly inferior performance on this metric.

Other hyperparameters For `BayesDAG`, we run 10 parallel SG-MCMC chains for $\boldsymbol{p}$ and $\boldsymbol{\Theta}$. We implement an adaptive sinkhorn iteration where the iteration automatically stops when the sum of rows and columns are closed to 1 within the threshold 0.001 (upto a maximum of 3000 iterations). Typically, we found this to require only around 300 iterations. We set the sinkhorn temperature t to be 0.2. For the reparametrization of $\boldsymbol{W}$ matrix with Gumbel-softmax trick, we use temperature 0.2. During evaluation, we use 100 SG-MCMC particles extracted from the particle buffer. We use 0.0003 for SG-MCMC learning rate l and batch size 512. We run 700 epochs to make sure the model is fully converged.

For DIBS, we can only use 20 SVGD particles for evaluation due to the quadratic scaling with the number of particles. We use 0.1 for Gumbel-softmax temperature. We run 10000 epochs for convergence. The learning rate is selected as 0.01.

Table 3 shows the hyperparameter selection for `BayesDAG`. Table 4 shows the hyperparameter selection for DIBS.

G. Additional Results

G.1. Performance with higher dimensional datasets

Full results for all the metrics for settings $d = 20$, $d = 70$ and $d = 100$ for nonlinear settings are presented in Figure 4, Figure 5 and Figure 6. We find that our method consistently outperforms the baselines with $d = 70$ and in terms of $\mathbb{E}$-SHD with $d = 100$. Competitive performance for $d > 50$ in nonlinear settings further demonstrates the applicability and computational efficiency of the proposed approach. In contrast, the only fully Bayesian nonlinear method, DIBS, is not computationally efficient to run for $d > 50$.

G.2. Performance of SG-MCMC with Continuous Relaxation

We compare the performance of SG-MCMC+VI and SG-MCMC with $\tilde{\boldsymbol{W}}$ on $d = 10$ ER and SF graph settings. Figure 7 shows the performance comparison. We can observe that SG-MCMC+VI generally outperforms its counterpart in most of the metrics. We hypothesize that this is because VI network μ_ϕ couples $\boldsymbol{p}$ and $\boldsymbol{W}$. This coupling effect is crucial since the changes in $\boldsymbol{p}$ results in the change of permutation matrix, where the $\boldsymbol{W}$ can immediately respond to this change through μ_ϕ. On the other hand, $\tilde{\boldsymbol{W}}$ can only respond to this change through running SG-MCMC steps on $\tilde{\boldsymbol{W}}$ with fixed $\boldsymbol{p}$. In theory, this is the most flexible approach since this coupling do not requires parametric form like μ_ϕ. However in practice, we cannot

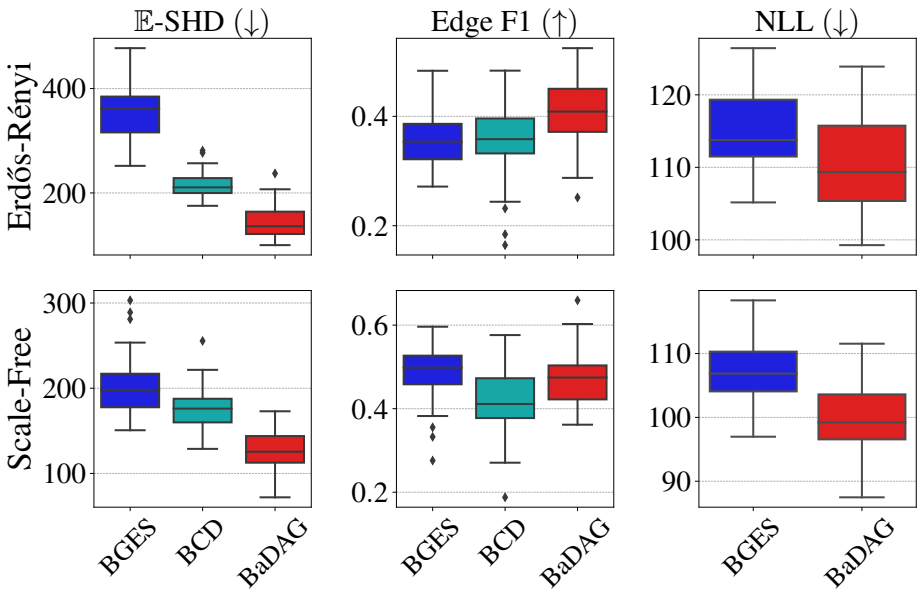

Figure 5: Posterior inference of both graph and functional parameters on synthetic datasets of nonlinear causal models with $d = 70$ variables. `BayesDAG` gives best results across most metrics. ↓ denotes lower is better and ↑ denotes higher is better. As DIBS and DDS are computationally prohibitive to run for this setting, it has been omitted. BCD has been omitted for NLL as we observed that it performs significantly worse.

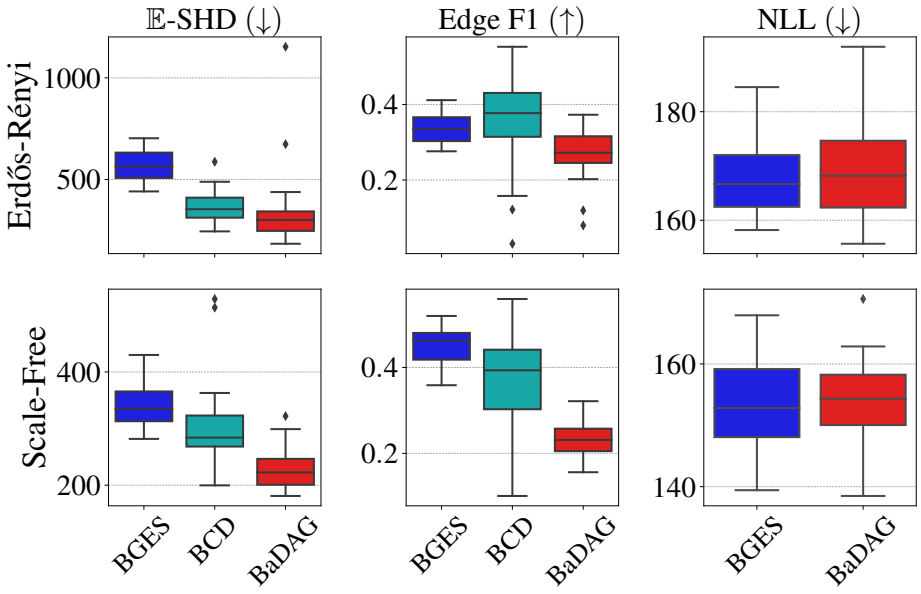

Figure 6: Posterior inference of both graph and functional parameters on synthetic datasets of nonlinear causal models with $d = 100$ variables. `BayesDAG` gives best results across 𝔼-SHD, comparable across NLL but slightly worse for Edge F1. ↓ denotes lower is better and ↑ denotes higher is better. As DIBS and DDS are computationally prohibitive to run for this setting, it has been omitted. BCD has been omitted for NLL as we observed that it performs significantly worse.

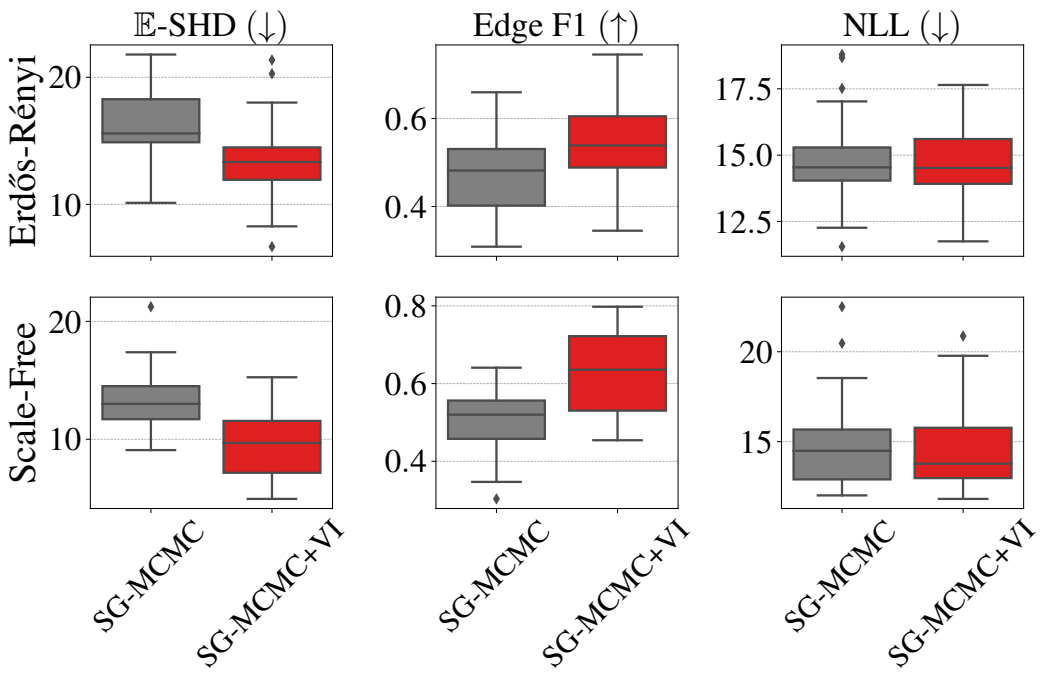

Figure 7: Performance comparison of SG-MCMC+VI v.s. fully SG-MCMC with $\tilde{\boldsymbol{W}}$ for $d = 10$ variables.

run many SG-MCMC steps with fixed $\boldsymbol{p}$ for convergence, which results in the inferior performance.

G.3. Ablation Study

We conduct ablation studies on our method using the nonlinear ER $d = 30$ dataset. Key findings are summarized below. All ablation studies are conducted using ER $d = 30$ datasets with the same hyperparameters reported in Appendix F.4.

Initialized $\boldsymbol{p}$ scale Figure 8 investigates the influence of the initialized scale of $\boldsymbol{p}$. We found that the performance is the best with $\alpha = 0.01$ or 10^{-5}, and deteriorates with increasing scales. This is because with larger initialization scale, the absolute value of the $\boldsymbol{p}$ is large. Longer SG-MCMC updates are needed to reverse the node potential order, which hinders the exploration of possible permutations, resulting in the convergence to poor local optima.

Number of SG-MCMC chains We examine the impact of the number of parallel SG-MCMC chains in Figure 9. We observe that the number of chains does not have a significant impact on the $\mathbb{E}$-SHD and Edge F1 scores.

Injected noise level for SG-MCMC In Figures 10 and 11, we study the performance differences arising from various injected noise levels for $\boldsymbol{p}$ and $\boldsymbol{\Theta}$ in the SG-MCMC algorithm (i.e. s of the SG-MCMC formulation in Appendix C). Interestingly, the noise level of $\boldsymbol{p}$ does not impact the performance as much as the level of $\boldsymbol{\Theta}$. Injecting noise helps improve the performance, but a smaller noise level should be chosen for $\boldsymbol{\Theta}$ to avoid divergence from optima.

H. Code and License

For the baselines, we use the code from the following repositories:

- BGES: We use the code from (Agrawal et al., 2019) from the repository https://github.com/agrawalraj/active_learning (No license included).
- DDS: We use the code from the official repository https://github.com/sharpenb/Differentiable-DAG-Sampling (No

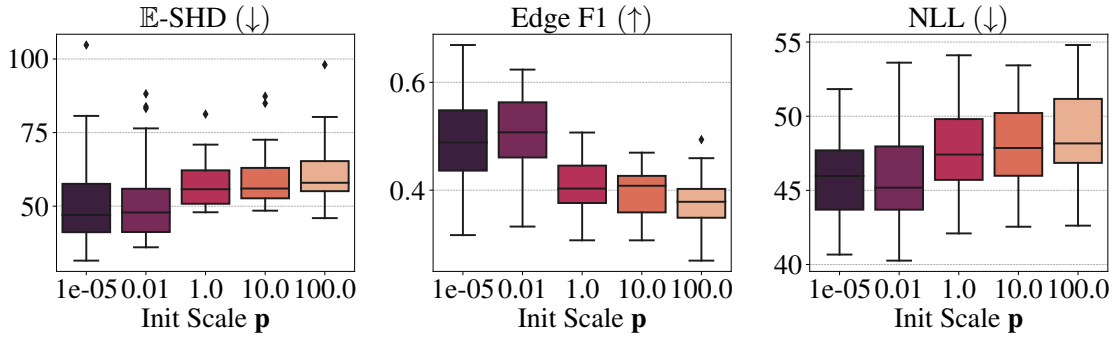

Figure 8: Posterior inference quality for $d = 30$ ER synthetic datasets with different initialized $\boldsymbol{p}$ scale. Best performance is obtained for 0.01, with larger scales leading to progressively worse performance.

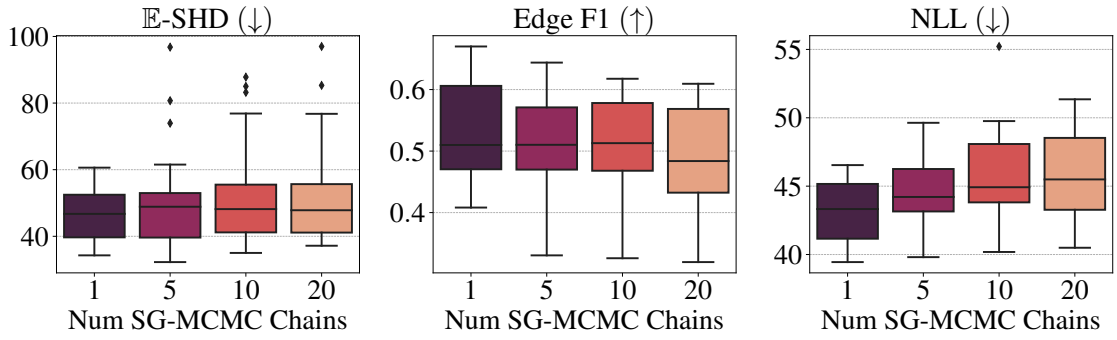

Figure 9: Posterior inference quality for $d = 30$ ER synthetic datasets with different number of parallel SG-MCMC chains. We find that the method performs the best with fewer chains, possibly due to easier optimization and computational complexity.

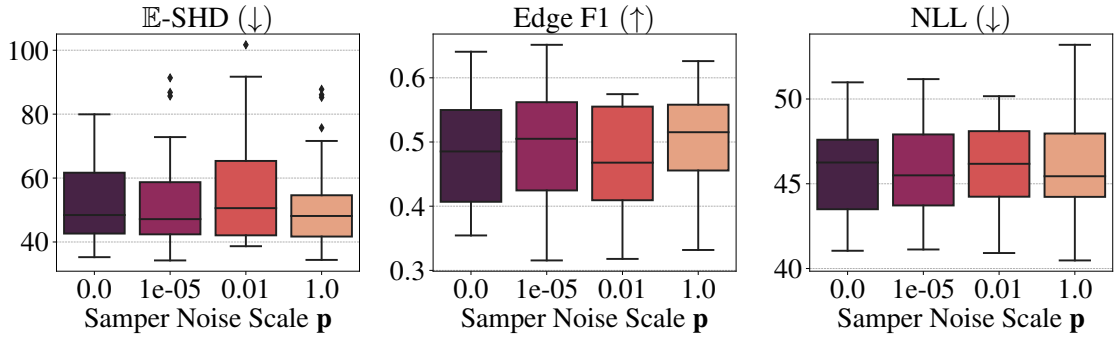

Figure 10: Posterior inference quality for $d = 30$ ER synthetic datasets with different level of injected noise scale for $\boldsymbol{p}$. We notice that the method is fairly insensitive to the level of noise injection.

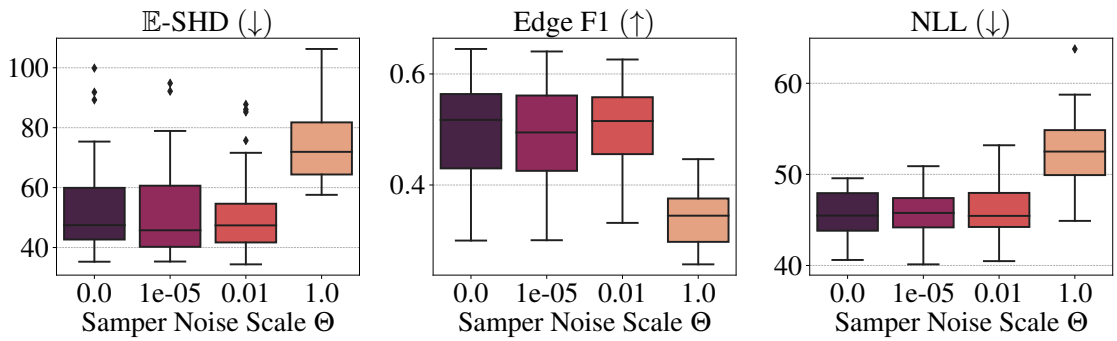

Figure 11: Posterior inference quality for $d = 30$ ER synthetic datasets with different level of injected noise scale for $\boldsymbol{\Theta}$. We notice that the method converges to good solutions with low level of noise injections, leading to superior performance.

license included).

- BCD: We use the code from the official repository https://github.com/ermongroup/BCD-Nets (No license included).
- DIBS: We use the code from the official repository https://github.com/larslorch/dibs (MIT license).

Additionally for the Syntren (Van den Bulcke et al., 2006) and Sachs Protein Cells (Sachs et al., 2005) datasets, we use the data provided with repository https://github.com/kurowasan/GraN-DAG (MIT license).