Incorporating Expert Knowledge into Bayesian Causal Discovery of Mixtures of Directed Acyclic Graphs

Zachris Björkman
Aalto University

Jorge Loría
Aalto University

Sophie Wharrie
University of Melbourne

Samuel Kaski
Aalto University
Manchester University
Ellis Institute Finland

Abstract

Bayesian causal discovery benefits from prior information elicited from domain experts, and in heterogeneous domains any prior knowledge would be badly needed. However, so far prior elicitation approaches have assumed a single causal graph and hence are not suited to heterogeneous domains. We propose a causal elicitation strategy for heterogeneous settings, based on Bayesian experimental design (BED) principles, and a *variational mixture structure learning* (VaMSL) method—extending the earlier *differentiable Bayesian structure learning* (DiBS) method—to iteratively infer mixtures of causal Bayesian networks (CBNs). We construct an informative graph prior incorporating elicited expert feedback in the inference of mixtures of CBNs. Our proposed method successfully produces a set of alternative causal models (mixture components or clusters), and achieves an improved structure learning performance on heterogeneous synthetic data when informed by a simulated expert. Finally, we demonstrate that our approach is capable of capturing complex distributions in a breast cancer database.

1 INTRODUCTION

Prior elicitation has been called "a central part of a Bayesian workflow" (Mikkola et al. 2024) and is especially important in settings where there are few observations but practitioners have considerable expertise. Domains of low data/high expertise are common in causal modeling which has recently been stressed in the machine learning field (Kitson et al. 2023). Indeed, the need to involve domain experts when determining and validating causal models has been stressed, as the modeling may involve strong assumptions that are difficult to falsify (Hasan et al. 2023, Brouillard et al. 2024). Others have listed further benefits, including reducing noise (Kitson et al. 2023) and improved inference (Eggeling et al. 2019).

When correctly specified, causal models remain valid even under *interventions*, changes in the system that alter the joint distribution—which have been used to conceptualize several problems faced by the machine learning field (e.g., distribution shifts; Schölkopf et al. 2012, Budhathoki et al. 2021). To infer such models, causal discovery methods aim to encode causal relationships as causal Bayesian networks (CBN) or structural causal models (Vowels et al. 2022). Additionally, many have proposed Bayesian inference (e.g., Lorch et al. 2021, Faria et al. 2022, Heckerman 2022) to yield a posterior over probable causal models, addressing various modeling uncertainties, and offering a formal way to incorporate background knowledge through the prior. Even so, to date, no priors over graph structure have reached "universal acclaim" (Eggeling et al. 2019). To leverage any information an expert might have about the causal relationships in the graph, we therefore propose both a novel informative graph prior to incorporate expert beliefs in the form of *hard* and *soft constraints* (Kitson et al. 2023) and a querying strategy based on Bayesian experimental design (BED) principles (Rainforth et al. 2024) to elicit causal information from an expert.

However, even with an informative prior, the assumption of a single causal graph may be too restrictive to encode an expert's beliefs. Recent surveys (Brouillard et al. 2024, Markham et al. 2022) in domains such as biology and medicine have pointed out that the common assumption of a *single* underlying causal

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s).

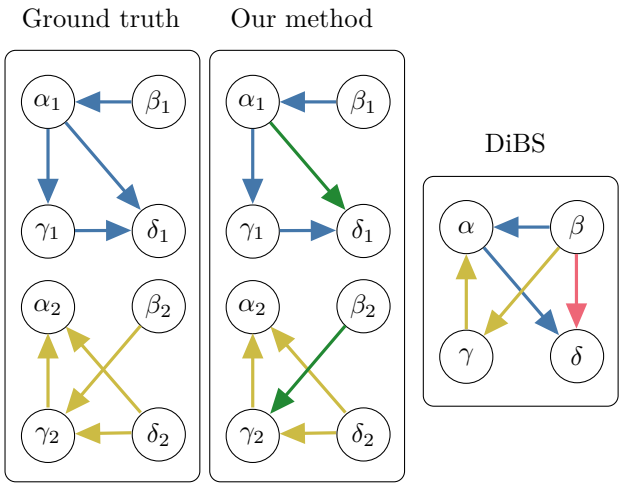

Figure 1: Example of inference using a mixture model as compared to a non-mixture model with ground truth graphs with two components (*left*), maximum-a-posteriori (MAP) graphs from our proposed mixture method (*center*) with elicited edges (in **green**), and MAP graph of non-mixture method DiBS (*right*) without edge elicitation. Edges from the first graph are shown in **blue**, and edges from the second graph are in **yellow**. Incorrectly inferred edges are shown in **red**. As shown, our method recovers the separate graphs while the non-mixture method finds one graph that tries to satisfy the contradictory information that the heterogeneous data encodes. For details see Appendix E.2.

graph may not hold. Simultaneously, the promising alternative of mixtures of causal graphs needs more study (Varıcı et al. 2024). To address the "heterogeneous" (Glymour et al. 2019) setting in a Bayesian way, we propose a *variational mixture structure learning* (VaMSL)[1] method to infer mixtures of CBNs. Our method extends the inference of the differentiable Bayesian structure learning (DiBS) method (Lorch et al. 2021) from single CBNs to mixtures, enabling the specification of priors based on expert beliefs on heterogeneity. Although mixtures of causal graphs have been gaining interest (Saeed et al. 2020, Strobl 2023, Varıcı et al. 2024) and Rittel and Tschiatschek (2025) also conclude the increased expressivity of mixtures can be beneficial for Bayesian causal discovery, few methods carry out Bayesian inference in the joint space of graphs structures and model parameters. These consist of works by Castelletti et al. (2020) and Marchant et al. (2025); however, their methods are limited to linear BNs and neither incorporates expert knowledge, making our method the first to carry out joint inference on both linear and non-linear BNs while incorporating expert knowledge. As is common, we assume causal faithfulness and causal sufficiency (Brouillard et al. 2024), within each component.

By combining VaMSL with the queried causal relationships, our method can learn mixtures of either linear or nonlinear CBNs in heterogeneous settings that arise in real-world scenarios. Figure 1 demonstrates this with a toy example using a two-component mixture with four nodes $(\alpha, \beta, \gamma, \delta)$. Further, the proposed mixture method incorporates expert knowledge of the data generating process (DGP) to aid in identifying the graphs. Our results show that our proposed method can identify homogeneous subsets in heterogeneous data with improved structure learning. To summarize, our contributions are the following:

- A novel informative (latent) graph prior to incorporate expert knowledge into causal discovery.
- A BED principled elicitation strategy to select optimal queries.
- A variational method for inference of mixtures of linear BNs **and**, for the first time, mixtures of nonlinear BNs.

Notation and Context of our Contributions within the Literature Consider a directed acyclic graph (DAG) G and parameters Θ. The tuple (G, Θ) encodes a Bayesian network (BN), when the joint distribution factorizes according to the directed relationships in G, that is: $p(\mathbf{x} \mid G, \Theta) = \prod_i p(x_i \mid \mathrm{pa}_G(x_i), \Theta)$, where $\mathrm{pa}_G(x_i)$ corresponds to the nodes that have a directed edge towards x_i—the child. Causal BNs are BNs where the edges are *causal* relationships, meaning that each child (x_i) is an *effect* for which each parent is a *direct cause*. With Bayesian inference we want to leverage observations $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbb{R}^d$, to infer posteriors used to compute expectations for graph features of the form $p(f \mid \mathcal{D}) = \sum_G p(G \mid \mathcal{D}) f(G)$, as described by Friedman and Koller (2000); although in this work we focus on the more general case of inferring both graphs and parameters as $p(G, \Theta \mid \mathcal{D})$. Inference in the graph space is challenging, since it is a discrete space. To this end, Lorch et al. (2021) proposed DiBS, building on the work by Zheng et al. (2018), which enables the use of gradient optimization methods for Bayesian inference in the (discrete) graph space via a latent continuous graph embedding space $\mathbf{Z} = [\mathbf{U}, \mathbf{V}]$, with $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{\ell \times d}$, where ℓ is a latent embedding dimensionality and d is the number of variables in the BN. A sample from $\mathbf{Z}$ parametrizes a temperature-dependent distribution over graphs according to $p_\omega(G \mid \mathbf{Z}) = \prod_{i \neq j} \mathrm{Ber}(G_{ij} \mid G_\omega(\mathbf{Z})_{ij})$ with

[1]Source code in Python JAX (Bradbury et al. 2018) available at https://github.com/ZacBjo/VaMSL_proj.

$G_\omega(\mathbf{Z})_{ij} = \sigma_\omega(\mathbf{U}_{\cdot i} \cdot \mathbf{V}_{\cdot j})$, where ω is an inverse temperature and $\sigma_\omega(\lambda) = 1/(1+\exp(-\omega\lambda))$ is the logistic function. We refer to the matrix of edge probabilities encoded by $G(\mathbf{Z})$ as a *soft graph* and suppress the dependence on ω when possible. The use of gradient methods in DAG search is enabled by incorporating a function h that characterizes the cyclicity of a given graph G, such that $h(G) = 0$ if and only if G is acyclic. Additionally, to infer DAGs, DiBS uses another temperature parameter β that is annealed over the optimization scheme, degenerating the prior probability of generating a cyclic graph, such that $p_\beta(\mathbf{Z}')|_{h(G')\neq 0,\ G'\sim p(G|\mathbf{Z}')}$ goes to zero when $\beta \to \infty$. For more details see Lorch et al. (2021).

A standard (finite) mixture model consists of distributions over a simplex of mixing weights $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$, *one-hot* assignment variables $\boldsymbol{c}_n = \{c_{nk}\}_{k=1}^K, n = 1, \ldots, N$, and a *mixture likelihood* for the observations. In the case of a mixture of BNs, consisting of K tuples of graphs and parameters denoted $\{(G_k, \Theta_k)\}_{k=1}^K$, these distributions are:

$$\boldsymbol{\pi} \sim \mathrm{Dir}(\boldsymbol{\alpha}),\ \ \boldsymbol{c}_n \sim \mathrm{Cat}(\boldsymbol{\pi}),\ \text{and}$$
$$p(\mathbf{x}_n \mid G, \Theta, \boldsymbol{c}_n) = \prod\nolimits_k p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}, \qquad (1)$$

respectively. A common technique for inference in these models is through variational approximations. An optimization scheme is then used to maximize the *evidence lower bound* (ELBO), which corresponds to minimizing the Kullback-Leibler divergence between the variational distribution and the posterior. As the standard approach of DiBS, we use Stein variational gradient descent (see, Liu and Wang 2016) in each component; for a review of variational methods we refer the reader to Zhang et al. (2018).

Inferring the true causal DGP from covariates in an observational setting is a challenging problem. Indeed, the alternative of compensating for a lack of data by making use of expert knowledge in causal discovery (structure learning) methods (e.g., Lorch et al. 2021) has long been acknowledged in the field (Cano et al. 2011). Early works combining causal discovery from observations with expert information to infer directed graphical models include Buntine (1991) and Heckerman et al. (1995). Here we consider structured informative priors. These priors are usually distinguished by whether they consist of *hard* or *soft* constraints (Angelopoulos and Cussens 2008, Constantinou et al. 2023). Hard constraints require (forbid) given edges, and soft constraints merely bias the search to including (excluding) them. Angelopoulos and Cussens (2008) review Bayesian methods incorporating informative priors on structure, noting that "*most BN learning approaches make no significant attempt to integrate prior knowledge into the learning process*" and that most priors are chosen for "pragmatic reasons". Based on a lack of "systematic comparisons", Eggeling et al. (2019) conduct a general investigation into common structure priors, favoring sparsity over the standard uniform choice. They support their arguments both analytically and experimentally by comparing multiple common structure priors. Constantinou et al. (2023) review a variety of methods to incorporate prior knowledge in causal discovery. Notably, they do not include soft constraints based on elicited probabilities. Further causal discovery methods incorporating hard constraints include Wang et al. (2020), Hasan and Gani (2022), Hasan and Gani (2023), and Sun et al. (2023). However, our method incorporates *both* hard and soft constraints as the latter are less strict. Rittel and Tschiatschek (2023) propose several priors to use on inference of a causal graph (including a prior for DiBS) which can themselves be expert-informed. However, they do not consider any possibility of actively learning the graph, and their implementations are not publicly available. Most recently, Marchant et al. (2025) propose a model (with an accompanying inference method) for mixtures of BNs, where the mixture probabilities depend on covariates, though the possibility of expert-informed priors is only addressed by indicating which of the covariates are "modifiable".

To optimally select edges to elicit from an expert, we use a BED approach (Rainforth et al. 2024). Previously, BED has been applied to causal discovery to pick maximally informative intervention experiments to run (see, Toth et al. 2022, Zhang et al. 2025, Zemplenyi and Miller 2023, Tigas et al. 2023), our goal is instead prior elicitation. While this is related to the work by Ibrahim et al. (2022), they do not try to infer full causal graphs instead focusing on inferring necessary interventions to regulate a target variable.

Applying BED requires a model; in our case of expert elicitation, a model of the expert. As such, our model includes two parts: the currently inferred latent graph posterior $p(\mathbf{Z} \mid \mathcal{D})$ and a simulator of responses from the expert. The former is a probability distribution over parameters for the latter. We incorporate the simulator by using the *expected information gain* (EIG) (Bernardo 1979, Lindley 1956) as a utility function, that we maximize over the set of possible queries and prompt the selected query to the expert. Maximizing the EIG corresponds to obtaining the largest (Shannon) information between the prior and (potential) posterior distributions, thereby optimally leveraging the expert.

The rest of the article is organized as follows. In Section 2 we present and explain the importance of elicitation in causal settings and our proposed priors. Section 3 exemplifies our methodological contributions in

the learning of a mixture of structured graphs. We demonstrate the performance of our approach in Section 4, by simulations and an application to a cancer identification problem in the UCI machine learning repository. We finalize with pointing out conclusions and future directions in Section 5.

2 ELICITATION OF CAUSAL-EDGE INFORMATION

For elicitation in the causal setting we assume there is an underlying CBN which corresponds to the DGP. Additionally, we assume that the queried expert has, possibly uncertain, knowledge of the underlying CBN (otherwise, why ask them?). We will query the expert on their beliefs regarding the edges in the graph of the CBN; we denote these edge-specific beliefs by $p(\psi_{ij} \mid \mathcal{K}_{ij})$, for each edge $i \to j$, where $\mathcal{K}_{ij}$ is the knowledge the expert is basing their beliefs on. Further, we assume the distribution which gives rise to their beliefs factorizes over the edges of the graph. The joint distribution of the edge beliefs is then

$$p(\Psi \mid \mathcal{K}) = \prod_{i \neq j} p(\psi_{ij} \mid \mathcal{K}_{ij}).$$

The difficulty of eliciting beliefs as a probability distribution is well-documented (O'Hagan 2006, Mikkola et al. 2024); we therefore only elicit a probability which we interpret as the mode of the distribution of edge-beliefs—denoted by $\psi^*_{ij} = \text{argmax}_{\psi_{ij} \in [0,1]}\, p(\psi_{ij} \mid \mathcal{K}_{ij})$. We distinguish between hard and soft constraints, and denote the hard constraint responses by $\mathcal{D}_H = \{\psi^*_{ij} \in \{0,1\} : \text{ the edge } i \to j \text{ has been elicited}\}$ and the soft constraint responses as $\mathcal{D}_S = \{\psi^*_{ij} \in (0,1) : \text{ the edge } i \to j \text{ has been elicited}\}$. We do **not** claim to observe the random variable ψ_{ij} nor its distribution $p(\psi_{ij} \mid \mathcal{K}_{ij})$—in general we cannot observe either. Rather we interpret the response we obtain as the mode of the distribution $p(\psi_{ij} \mid \mathcal{K}_{ij})$. This approach can be extended to other summary statistics than the mode, but that is left for future work.

Our aim is to incorporate the expert's responses $(\mathcal{D}_H, \mathcal{D}_S)$ into the inference of the data generating CBN through an informative graph *prior*. We want to minimize the number of queries since the act of querying is time-consuming and potentially expensive. Moreover, an exhaustive query of all the edges is infeasible even for graphs with a moderate number of nodes. As mentioned by Rittel and Tschiatschek (2023), the potential issue of an expert expressing cyclic beliefs (forbidden by the DAG assumption), is mitigated by the cyclicity regularizer that the DiBS method leverages as part of its prior. The cyclicity regularizer assures the inferred graphs are DAGs and allows us define to our querying strategy in terms of edge beliefs, rather than orderings (e.g., Buntine 1991) or other DAG-ensuring constructs. This significantly reduces the burden of the expert while the interpretability of the query can contribute to the quality of the responses. We address the need to minimize queries by relying on BED principles to select the most informative edges for our prior.

2.1 Selecting Elicitation Queries by Bayesian Experimental Design

We elicit the expert responses $(\mathcal{D}_H, \mathcal{D}_S)$ that are used to construct the elicitation prior in the following way. Suppose we have access to a posterior distribution over latent graph embeddings $p(\mathbf{Z} \mid \mathcal{D})$, for instance using DiBS, where each $\mathbf{Z}$ encodes probabilities $G(\mathbf{Z})_{ij} \in [0,1]$ for each edge $i \to j$ in a causal graph. Consider a query of a specified edge as an experiment $\xi_{ij} \in \Xi$, which results in a response $\psi^*_{ij} \in [0,1]$. This response is the expert's belief (probability) for the edge existing in the causal graph G. Depending on the application, the design space Ξ can vary. However, we will assume it consists of all the edges in the causal graph (barring self-loops). For instance, each observation could contain variables from two snapshots at times $t_1 < t_2$, logically we would not want any variable at time t_2 to have an edge directed towards variables at time t_1; such possibilities can be excluded from Ξ.

The BED framework (Rainforth et al. 2024) requires a parameter prior as well as a simulator for experiment outcomes. In our case these will be $p(\mathbf{Z} \mid \mathcal{D})$ and $p(\psi^*_{ij} \mid \mathbf{Z}, \xi_{ij})$, respectively. To simulate querying the expert about their edge belief, we propose the simulator

$$\begin{aligned} &p(\psi^*_{ij} \mid \mathbf{Z}, \xi_{ij}) \\ &= \text{Beta}(\alpha_s G(\mathbf{Z})_{ij} + 1, \beta_s(1 - G(\mathbf{Z})_{ij}) + 1), \end{aligned}$$

where setting $\alpha_s = \beta_s$ yields a distribution with a mode of $G(\mathbf{Z})_{ij}$ and the variance of the simulator is fully determined by the choice of value for α_s. In our experiments we set $\alpha_s = \beta_s = 10$.

We propose to use the EIG to evaluate and select queries to pose to the expert. The EIG serves as a utility function, and we can simulate the possible future responses from the expert given by the current model $p(\mathbf{Z} \mid \mathcal{D})$. Maximizing the EIG finds the edge that is expected to provide the most information for posterior inference; this enables an efficient convergence to the causal graph of interest. Specifically, we pick the

optimal query ξ^*_{ij} that satisfies

$$\begin{aligned}&\underset{\xi_{ij}\in\Xi}{\operatorname{argmax}}\ \mathrm{EIG}(\xi_{ij}) \\ &= \underset{\xi_{ij}\in\Xi}{\operatorname{argmax}}\ \mathbb{E}_{p(\mathbf{Z}|\mathcal{D})p(\psi^*_{ij}|\mathbf{Z},\xi_{ij})}\left[\ln p(\psi^*_{ij}\mid \mathbf{Z},\xi_{ij})\right. \\ &\quad \left.-\ln p(\psi^*_{ij}\mid \xi_{ij})\right].\end{aligned} \tag{2}$$

For this we use a nested Monte Carlo approximation $\hat{\mu}_{\mathrm{NMC}}$ (see Appendix D for details). This requires sampling $\mathbf{Z}^{(n)} \sim p(\mathbf{Z} \mid \mathcal{D})$ from the parameter prior. This is trivial to do from the particle posterior that the Stein variational gradient descent (SVGD) instantiation of DiBS provides. We note that, in case we are only interested in binary responses $\psi^*_{ij} \in \{0,1\}$, we can use a Bernoulli distributed simulator and tractably compute the outer expectation of Eq. (2) across ψ^*_{ij}, enabling the use of the Rao-Blackwellized EIG estimator $\hat{\mu}_{\mathrm{RB}}$. Full details of how these Monte Carlo approximations are computed, and alternatives for EIG approximations, are described by Rainforth et al. (2024).

2.2 Elicited Informative Graph Prior

Our goal is to infer CBNs conditional on elicited responses $(\mathcal{D}_H, \mathcal{D}_S)$ from the expert in the form of hard and soft constraints, respectively, that were selected as described above. *Hard constraints* directly shrink the considered graph space and have been called "the most significant form of prior knowledge" (Angelopoulos and Cussens 2008). Allowing experts to fully rule out edges that do not coincide with their fundamental understanding of the causal system under investigation (e.g., in biomedical applications a biomarker cannot "cause" the age of a patient) or require edges they deem necessary. Imposing hard constraints of absence (existence) requires $p(G) = 0$ whenever $G_{ij} = 1$ ($G_{ij} = 0$), which is achieved by setting the prior of such graphs to satisfy this condition. Notably, in DiBS this is done by 0/1-masking the appropriate edges when converting latent graph embeddings $\mathbf{Z}$ to (soft) graphs G, restricting the graph search to only the space of graphs satisfying the hard constraints. To incorporate *soft constraints* we construct an informative graph prior based on Bayesian prior elicitation (Mikkola et al. 2024).

To impose an informative prior on the latent space $\mathbf{Z}$, we propose a user model for the expert inspired by the *imaginary observations* device (Consonni et al. 2018, Good 1950) and generate a hypothetical set of observations which the expert would condition on to form their opinion. We define a mapping $f : [0,1] \to \mathbb{N}^2$ from expert responses ψ^*_{ij} to the tuple $\mathcal{K}_{ij} = (n_{ij}, k_{ij})$, representing a number of *imagined observations* (trials) n_{ij} and a number of successes k_{ij} (observations of the existence of the causal edge $i \to j$). These imagined observations are part of a data generating story for an idealized experiment the expert has conducted to investigate the existence (absence) of a causal edge to form their posterior edge beliefs, such that for all $i \neq j$, $p(\psi_{ij} \mid \mathcal{K}_{ij}) \propto p(\mathcal{K}_{ij} \mid \psi_{ij})p(\psi_{ij})$.

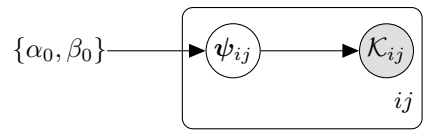

Figure 2: Generative model for expert edge beliefs (ψ_{ij}), with hyperparameters α_0, β_0, and trials observed by the expert ($\mathcal{K}_{ij}$).

The generative model for the expert's beliefs follows the diagram shown in Figure 2. We assign beta priors for $i \neq j$, $p(\psi_{ij}) = \mathrm{Beta}(\psi_{ij} \mid \alpha_0, \beta_0)$ to represent the beliefs of the expert before they observed any causal edges (letting each edge prior share parameters). Additionally, we model the update of the expert's beliefs about an edge's existence as a beta-binomial conjugate distribution. For simplicity, we assume the idealized experiment the expert uses to verify a causal edge's existence is deterministic. Hence, if i is a direct cause of j (i.e., $i \to j$) then all trials confirm it (i.e., $k_{ij} = n_{ij}$), and vice versa (i.e., $k_{ij} = 0$). Given that the queried responses $\psi^*_{ij} \in \mathcal{D}_S$ are the modes of the expert's posterior beliefs, conjugacy yields a unique mapping from an expert's prior $\mathrm{Beta}(\alpha_0, \beta_0)$ to their posterior $p(\psi_{ij} \mid \mathcal{K}_{ij}) = \mathrm{Beta}(\alpha_{ij}, \beta_{ij})$, with the (elicited) mode $\psi^*_{ij} = \frac{\alpha_{ij}-1}{\alpha_{ij}+\beta_{ij}-2}$. As the mode must have moved from the mode of the prior $\psi^*_{ij,0} = \frac{\alpha_0-1}{\alpha_0+\beta_0-2}$ in the direction supported by observations, we derive that:

$$\begin{aligned}&f_{\alpha_0,\beta_0}(\psi^*_{ij}) \\ &= \begin{cases}\mathcal{K}_{ij} = \left(n_{ij} = \lfloor \frac{\psi^*_{ij}(\alpha_0+\beta_0-2)-\alpha_0+1}{1-\psi^*_{ij}} \rfloor, k_{ij} = n_{ij}\right), \\ \quad \text{if} \quad \psi^*_{ij} > \psi^*_{ij,0}; \\ \mathcal{K}_{ij} = \left(n_{ij} = \lfloor \frac{\alpha_0-1-\psi^*_{ij}(\alpha_0+\beta_0-2)}{\psi^*_{ij}} \rfloor, k_{ij} = 0\right), \\ \quad \text{if} \quad \psi^*_{ij} < \psi^*_{ij,0},\end{cases}\end{aligned} \tag{3}$$

where the flooring ensures that we only consider integer observations and the number of imaginary observations can be identified by rearranging the posterior mode. For more details, see Appendix B.

We denote the set of all mappings $\mathcal{D}_{\mathcal{K}} = f(\mathcal{D}_S)$ and construct the *elicited informative graph prior* on the latent graph embeddings $\mathbf{Z}$:

$$\begin{aligned}p(\mathbf{Z} \mid \mathcal{D}_S) &\propto p(\mathbf{Z})p(f(\mathcal{D}_S) \mid \mathbf{Z}) \\ &= p(\mathbf{Z}) \prod\nolimits_{i\neq j} p(\mathcal{K}_{ij} \mid G(\mathbf{Z})_{ij}),\end{aligned} \tag{4}$$

where the *elicitation likelihood* (Mikkola et al. 2024) $p(\mathcal{K}_{ij} \mid G(\mathbf{Z})_{ij})$ is evaluated by parameterizing a bi-

nomial distribution with the soft graph encoded by $\mathbf{Z}$, with trials confirming (refuting) an edge viewed as k successes ($n-k$ failures):

$$p(\mathcal{K}_{ij} \mid G(\mathbf{Z})_{ij}) = \mathrm{Bin}(n = n_{ij}, k = k_{ij} \mid p = G(\mathbf{Z})_{ij}),$$

where, in each case, both the binomial coefficient and one of the factors in the binomial probability mass function evaluate to 1 (as either $k_{ij} = n_{ij}$ or $k_{ij} = 0$, see Eq. (3)). Note that the informativeness of the prior is set by the choice of α_0, β_0, as they determine how many imagined observations a given response from the expert maps to. Potential cyclic beliefs are not an issue, as the cyclicity regularizer in the DiBS prior $p(\mathbf{Z})$ collapses the latent graph space unto the DAG space during optimization, eliminating any possible cyclic beliefs in the posterior. For more details of our implementation, see Appendix C.

2.3 Incorporating Expert Responses to Model Updates

In our querying scheme, the expert is queried after a set number of optimization steps, using the currently inferred distribution as the parameter prior for the BED selected queries. Once the querying is completed, the structure learning method can continue, conditional on the new responses. We here assume the problem of label switching (Celeux 1998) is addressed by the expert being able to indicate which component refers to which set of beliefs (e.g., by visual inspection of the current clustering). Moreover, expert information is a way to specify non-exchangeable priors, which prevents label switching (Betancourt 2017).

The number of times the optimization is interrupted to query the expert depends on how many times the expert can be queried as well as the total computational budget. Notably, in the case of our proposed model, described in Section 3, the iterative coordinate-ascent variational inference (CAVI) procedure naturally enables the expert to be queried every time the clustering is updated and new information about identified subpopulations can be provided. Thus, the inference of our proposed mixture method, when paired with our elicited graph prior and used by a domain expert, is informed by responses to queries about edges they supply.

3 (DIFFERENTIABLE) VARIATIONAL MIXTURE STRUCTURE LEARNING

Classically, a single graph has been assumed to be sufficient to explain the causal mechanisms of a real-world phenomena, however the possibility of several graphs explaining the data should not be ignored as the real world is often complex and cannot be easily described. Correspondingly, learning mixtures of models is more demanding than a single model, both computationally, and statistically. For graphs this difficulty compounds (Varıcı et al. 2024). To alleviate this difficulty, we incorporate expert beliefs about the causal relationships into the inference procedure through the construction of component- and edge-specific informative graph priors.

To address data heterogeneity, we use a mixture of CBNs (Eq. (1)). We leverage the method proposed by Lorch et al. (2021) to infer a posterior with respect to CBNs with graphs in the (continuous) latent graph embedding space as well as the mixing weights and the assignment variables. This enables the use of gradient optimization to carry out the (otherwise discrete) search across the graph space. Our method applies variational inference to approximate the posterior as

$$q(\boldsymbol{\pi}, \mathbf{Z}, \Theta, C) \approx p(\boldsymbol{\pi}, \mathbf{Z}, \Theta, C \mid \mathcal{D}).$$

We refer to Lorch et al. (2021) for a detailed account of how inference over the latent variables $\mathbf{Z}_k$ corresponds to inference over graphs G_k ($K = 1$ in our setting). However, any inference algorithm that yields a posterior over CBNs can be used for VaMSL. As is common in VI, we make the mean-field assumption, approximating the posterior as the factorizing distribution:

$$\begin{aligned} q(\boldsymbol{\pi}, \mathbf{Z}, \Theta, C) &= q(\boldsymbol{\pi}) q(\mathbf{Z}, \Theta) q(C) \\ &= q(\boldsymbol{\pi}) \prod\nolimits_k q(\mathbf{Z}_k, \Theta_k) \prod\nolimits_n q(\boldsymbol{c}_n). \end{aligned}$$

The factorizing distributions are iteratively optimized using CAVI. To optimize the variational CBN distributions $q(G, \Theta)$, we need to employ a Bayesian structure learning framework. In this work, we use VaMSL with the DiBS framework (Lorch et al. 2021), replacing inference over G with the continuous relaxation $\mathbf{Z}$ and enabling the use of gradient-based optimization methods to carry out posterior inference. Before introducing the CAVI-steps employed to optimize the variational approximation of the posterior, we describe how we use the continuous relaxation over the graph space.

3.1 (Differentiable) Generative Graph Mixture Model

Figure 3 shows the DGP of our mixture model, using DiBS, and it corresponds to

$$
\begin{aligned}
&p(\boldsymbol{\pi}, \mathbf{Z}, G, \Theta, C, \mathcal{D} \mid \mathcal{D}_S, \mathcal{D}_H) \\
&= p(\boldsymbol{\pi}) p(\mathbf{Z} \mid \mathcal{D}_S) p(G \mid \mathbf{Z}, \mathcal{D}_H) p(\Theta \mid G) \\
&\quad \times p(C \mid \boldsymbol{\pi}) p(\mathcal{D} \mid C, G, \Theta) \\
&= p(\boldsymbol{\pi}) \prod_k \left[p(\mathbf{Z}_k \mid \mathcal{D}_{S,k}) p(G_k \mid \mathbf{Z}_k, \mathcal{D}_{H,k}) p(\Theta_k \mid G_k) \right] \\
&\quad \times \prod_n \left[p(\boldsymbol{c}_n \mid \boldsymbol{\pi}) \prod_k p(\mathbf{x}_n \mid c_{nk}, G_k, \Theta_k) \right],
\end{aligned}
$$

where graphs with d variables are represented by adjacency matrices $G_k \in \{0,1\}^{d \times d}$. Each component of the graph embeddings $\mathbf{Z}_k = [\mathbf{U}^{(k)}, \mathbf{V}^{(k)}]$ with $\mathbf{U}^{(k)}, \mathbf{V}^{(k)} \in \mathbb{R}^{\ell \times d}$ takes the form described in Section 1. The parameter distribution over Θ is a modeling choice dependent on the chosen family of CBNs, where we consider both the linear and non-linear case. The data-likelihood corresponds to the mixture likelihood in Eq. (1).

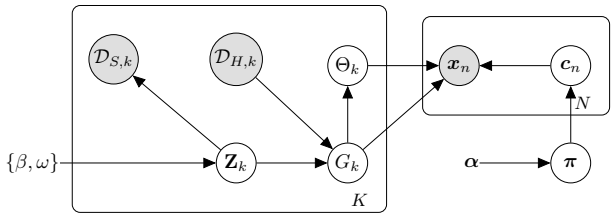

Figure 3: Generative model for VaMSL using DiBS.

The generative graph prior of the latent graph embeddings $\mathbf{Z}_k$, encodes a distribution over directed graphs via the following generative model:

$$
\begin{aligned}
p_\omega(G_k \mid \mathbf{Z}_k, \mathcal{D}_{H,k}) &= \prod_{ij : \psi^*_{ij} \in \mathcal{D}_H} \psi^*_{ij} \\
&\times \prod_{ij : \psi^*_{ij} \notin \mathcal{D}_H} \mathrm{Ber}\left(G_{k,(ij)} \mid \sigma_\omega(\mathbf{U}^{(k)}_{\cdot i} \cdot \mathbf{V}^{(k)}_{\cdot j}) \right),
\end{aligned} \tag{5}
$$

where $G_{k,(ij)}$ refers to the edge $i \to j$ in component k. Recall that the *hard* constraints $\psi^*_{ij} \in \mathcal{D}_H$ can only take values of zero or one, meaning that they rule out any graphs that violate them. Additionally, Lorch et al. (2021) define the *soft graph*, the matrix of edge probabilities encoded by the latent graph $\mathbf{Z}_k$, as: $G_\omega(\mathbf{Z}_k)_{ij} = \sigma_\omega(\mathbf{U}^{(k)}_{\cdot i} \cdot \mathbf{V}^{(k)}_{\cdot j})$. As the temperature is annealed $\omega \to \infty$, the prior in Eq. (5) collapses onto the DAG space and $G_\infty(\mathbf{Z}_k)$ defines an adjacency matrix.

The prior for the latent graph embeddings $\mathbf{Z}_k$ is the product of four terms, each of which ensures specific properties of the generative graph prior. Namely,

$$
\begin{aligned}
&p_{\beta,\omega}(\mathbf{Z}_k \mid \mathcal{D}_S) \\
&\propto p\left(\mathcal{D}_{S,k} \mid G_\omega(\mathbf{Z}_k)\right) \\
&\quad \times \exp\left(-\beta \, \mathbb{E}_{p_\omega(G_k \mid \mathbf{Z}_k)}[h(G_k)]\right) \\
&\quad \times \prod_{i \neq j} \mathcal{N}(\mathbf{U}^{(k)}_{\cdot i} \mid \mathbf{0}, \sigma_z I) \, \mathcal{N}(\mathbf{V}^{(k)}_{\cdot j} \mid \mathbf{0}, \sigma_z I) \\
&\quad \times p\left(G_\omega(\mathbf{Z}_k)\right).
\end{aligned} \tag{6}
$$

The first term is our proposed *elicited informative graph prior*, ensuring that the obtained embedding aligns with the expert's beliefs (see Section 2.2). The last three terms, introduced by Lorch et al. (2021), respectively constitute: a term that penalizes cyclicity in samples from the generative graph prior, priors ensuring that the inner products $\mathbf{U}^{(k)}_{\cdot i} \cdot \mathbf{V}^{(k)}_{\cdot j}$ remain "well-behaved", and a term that promotes specified random graph structures. Specifically, for Erdős-Rényi graphs, Lorch et al. (2021) suggest the prior $p\left(G_\omega(\mathbf{Z}_k)\right) \propto q^{\|G_\omega(\mathbf{Z}_k)\|_1} (1-q)^{\binom{d}{2} - \|G_\omega(\mathbf{Z}_k)\|_1}$, while scale-free graphs are promoted using $p\left(G_\omega(\mathbf{Z}_k)\right) \propto \prod_{i=1}^d \left(1 + \|G_\omega(\mathbf{Z}_k)_i\|_1\right)^{-3}$, where $G_\omega(\mathbf{Z}_k)_i$ is the ith row in the soft graph. In Appendix A.1 we discuss implementation details with a full procedure to approximate the posterior distribution of $p(\pi, \mathbf{Z}, G, \Theta, C \mid \mathcal{D})$. Algorithm 1 describes how to incorporate the expert's edge-beliefs to the approach described in this section.

Algorithm 1 VaMSL + expert

Require: Number of queries $M \times B$; observations $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$; number of components K

1: Obtain $q_{\beta,\omega}(\pi, \mathbf{Z}, G, \Theta, C)$ that approximates $p(\pi, \mathbf{Z}, G, \Theta, C \mid \mathcal{D})$, using Algorithm 2.
2: Choose the top M queries for each of the K components that maximizes the EIG (see: Section 2.1).
3: Query the edge beliefs from the expert $(\mathcal{D}_S, \mathcal{D}_H)$.
4: Update $q_{\beta,\omega}(\pi, \mathbf{Z}, G, \Theta, C)$ to approximate $p(\pi, \mathbf{Z}, G, \Theta, C \mid \mathcal{D}, \mathcal{D}_S)$, using Algorithm 2.
5: Go to Step 2 until out of query budget.

4 NUMERICAL EXPERIMENTS AND CANCER LABELING

We demonstrate that incorporating soft constraints through our proposed prior improves the inference, even when the expert can make mistakes. For this, we present experiments on single-component, two-component, and a breast-cancer data set. The experiments compare four approaches: VaMSL with expert information, VaMSL without expert information, a Gaussian mixture model (GMM; using `scikit-learn`, Pedregosa et al. 2011), a causal k-means kernel

(Markham et al. 2022), DiBS without label information, and DiBS with label information. We remark that the last approach is an ideal scenario (the labels ensure it has the correct clusters), as such it provides an unrealistic bound which no unsupervised method should be able to accomplish. In principle we could compare to the approach by Marchant et al. (2025), but we did not find an implementation of their method.

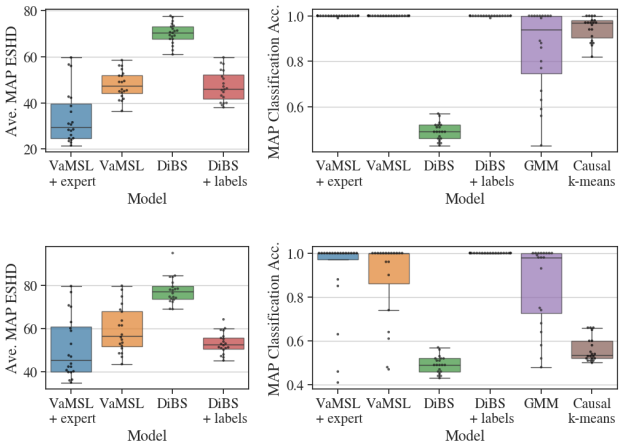

Figure 4: Boxplot (and values) of average ESHD between learned graphs and true graphs (*left*) and classification accuracy (*right*) by method, in data generated from a mixture of two **linear** (*top*) and **non-linear** (*bottom*) ER BNs with Gaussian errors. Learned graphs not available for Causal k-means and GMMs.

4.1 Data Generation and Metrics

To validate VaMSL on synthetic settings, we generate data from Gaussian BNs with Erdős-Rényi (ER) ground truth graphs parametrized by linear or nonlinear mean functions. These graphs are generated with an expectation of two edges per node. Nonlinearity is instantiated using shallow 2-layer neural networks (NN) with 5 neurons per hidden layer. All the parameters for the NNs and linear models are randomly set with a standard normal distribution.

We assess the performance of all methods in terms of classification accuracy (for correct labeling) with the *maximum a posteriori* (MAP) component on held-out observations ($\mathcal{D}_{ho}$) as the percentage of correctly classified observations. When a graph is learned by the method (namely, DiBS and VaMSL variants), we assess the structure learning by the *expected structural Hamming-distance* (ESHD), given by

$$\text{ESHD}(p(G_k \mid \mathcal{D}), G^*) = \mathbb{E}_{p(G_k|\mathcal{D})}\left[\text{SHD}(G_k, G^*)\right],$$

where $\text{SHD}(G_k, G^*)$ is the number of edge changes needed to move from G_k to G^*. Additionally, in Appendix E we provide sensitivity studies, simulations with scale-free priors, other simulation results, and assessment of predictive power where the same conclusion of this section is echoed: learning the graphs by incorporating an expert helps the inference process compared to the competing baselines. An implementation of our algorithm is freely available in https://github.com/ZacBjo/VaMSL_proj.

Further ablation studies are provided in Appendices E.9, E.10, and E.11, where we compare against a hard-encoding strategy (using the method of Sun et al. 2023), the scalability of our method to higher dimensions, and the behaviour with a misspecified number of components, respectively.

4.2 Validation with Mixture of Graphs

To investigate the heterogeneous setting we generate a two-component dataset $\mathcal{D} = \{\mathbf{x}_n \mid \mathbf{x}_n \in \mathbb{R}^{20}\}_{n=1}^{100}$ with mixing probabilities $(0.5, 0.5)$, and each DAG is generated from the random graph prior. We repeat the simulations 20 times and evaluate on out-of-sample simulations. We apply the competing methods to the generated datasets. In Fig. 4 we show the effectivity of both VaMSL as well as VaMSL paired with expert information (labeled VaMSL + expert) via our proposed prior for both linear and nonlinear models, respectively; in contrast to the competing methods.

The results in Fig. 4 indicate how DiBS is *unable* to accurately describe the underlying DGP—given by the graph-structures, as it tries to find a single graph to describe two separate graphs. VaMSL, on the other hand, is able to cluster the observations into homogeneous subpopulations (right panel) and infer a graph posterior comparable to those achieved when knowing the labels (left panel). Further, the VaMSL approaches outperform the GMM and the causal k-means methods in terms of classification accuracy. We recall that the k-means and GMM methods do not have a structure learning component so we cannot assess their ESHD.

4.3 Querying Strategy Validation

We consider a single-component setting, with different querying strategies and graph parametrizations (linear and nonlinear), for validation of our expert elicitation strategy (Section 2.1). Fig. 5 (top) shows how increasing the number of queries to an expert (90% reliable) leads to improved structure learning, as expected. Additionally, our BED strategy outperforms random querying of an expert. In Fig. 5 (bottom) we display the effect of the reliability of the expert, showing that more reliable opinions impact the inference more dramatically. As an expert would be the only

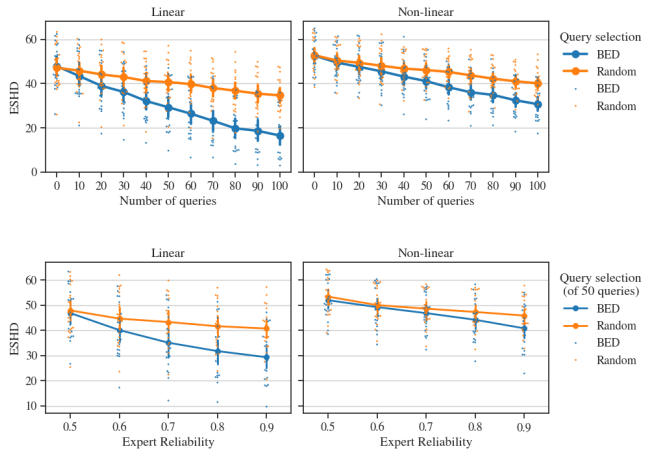

Figure 5: Comparison between BED and random approaches for query selection by mean and bootstrapped 95% confidence interval for homogeneous experiments with varying number of queries (*top*) and varying expert reliability (*bottom*) when inferring Gaussian ER network in the **linear** (*left*) and **non-linear** case (*right*).

one to reject an inferred model (regardless of predictive performance), it is indeed desirable that our prior emphasizes expert opinion proportionally to how strongly said opinion is held. In Appendix E.3 we detail how the expert responses are simulated.

4.4 Label Assignment in Breast Cancer Data

As an application we use the Diagnostic Wisconsin Breast Cancer Database[2]. The data set consists of 569 observations with binary labels (for classification) and 30 real-valued features. We use the labels as the mixture components and interpret the problem as an unsupervised problem, with task to infer the labels. As Fig. 6 shows, VaMSL+expert outperforms all the other approaches, and VaMSL alone performs as good as a GMM. We provide details of the expert simulation as well as other relevant metrics in Appendix E.6. In Apppendix E.6.1 we showcase how the performance of the method improves as the number of queries to the expert increases. We simulate the expert with a reliability of 0.9, more details in Appendix E.3.

5 CONCLUSIONS

We have presented a strategy for eliciting expert beliefs about causal relationships as well as an elicited informative graph prior. Further, we have addressed the heterogeneous setting by proposing a mixture method,

[2]from: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic.

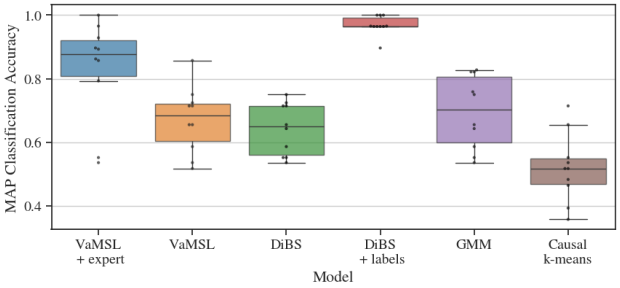

Figure 6: Boxplot and values of classification accuracy by method in out-of-sample data for the breast cancer dataset, in ten splits of the data.

which when paired with the expert responses in the form of non-exchangeable component-wise priors, infers mixtures of CBNs. The method has been demonstrated to improve the structure learning and allows finding correct clusters with an unsupervised method, through an expert-in-the-loop.

We consider that our work can be expanded in at least four main directions. First, the proposed method we call "VaMSL" uses DiBS for the soft-graph inference part. This leaves open the possibility of using other graph discovery methods within VaMSL—different from DiBS, such as by Annadani et al. (2023), Charpentier et al. (2022), and Tran et al. (2023), to encode the soft-graph distribution. Second, the assignment probabilities do not depend on covariates, a way to drop this assumption is to use a similar approach as Marchant et al. (2025). Third, Darvariu et al. (2024) propose querying a large language model for expert information, when an expert is not available, an idea which is easy to adapt to our proposed method. Finally, although our work is primarily concerned with the low data-regime (where prior information is most valuable), ensuring that our approach can scale to high dimensions — where causal discovery algorithms become prohibitively expensive — is of interest and is an active area of research in Bayesian causal discovery.

Acknowledgements

This work was supported by the Research Council of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI, 358958 and 359567), ELISE Networks of Excellence Centres (EU Horizon: 2020 grant agreement 951847) and UKRI Turing AI World-Leading Researcher Fellowship, EP/W002973/1. We acknowledge the computational resources provided by the Aalto Science-IT Project from Computer Science IT. We thank Yasir Barlas, Sammie Katt, and reviewers for comments on early versions of the manuscript.

References

Angelopoulos, N. and Cussens, J. (2008). Bayesian learning of Bayesian networks with informative priors. *Annals of Mathematics and Artificial Intelligence*, 54(1):53–98.

Annadani, Y., Pawlowski, N., Jennings, J., Bauer, S., Zhang, C., and Gong, W. (2023). BayesDAG: Gradient-Based Posterior Inference for Causal Discovery. *Advances in Neural Information Processing Systems*, 36:1738–1763.

Bernardo, J. M. (1979). Expected information as expected utility. *the Annals of Statistics*, pages 686–690.

Betancourt, M. (2017). Identifying Bayesian mixture models. https://mc-stan.org/users/documentation/case-studies/identifying_mixture_models.html.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: Composable transformations of Python+NumPy programs.

Brouillard, P., Squires, C., Wahl, J., Kording, K. P., Sachs, K., Drouin, A., and Sridhar, D. (2024). The Landscape of Causal Discovery Data: Grounding Causal Discovery in Real-World Applications.

Budhathoki, K., Janzing, D., Bloebaum, P., and Ng, H. (2021). Why did the distribution change? In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pages 1666–1674. PMLR.

Buntine, W. (1991). Theory refinement on Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'91, pages 52–60, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Cano, A., Masegosa, A. R., and Moral, S. (2011). A Method for Integrating Expert Knowledge When Learning Bayesian Networks From Data. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(5):1382–1394.

Castelletti, F., La Rocca, L., Peluso, S., Stingo, F. C., and Consonni, G. (2020). Bayesian learning of multiple directed networks from observational data. *Statistics in Medicine*, 39(30):4745–4766.

Celeux, G. (1998). Bayesian Inference for Mixture: The Label Switching Problem. In Payne, R. and Green, P., editors, *COMPSTAT*, pages 227–232, Heidelberg. Physica-Verlag HD.

Charpentier, B., Kibler, S., and Günnemann, S. (2022). Differentiable DAG Sampling. In *International Conference on Learning Representations 10*.

Consonni, G., Fouskakis, D., Liseo, B., and Ntzoufras, I. (2018). Prior distributions for objective Bayesian analysis. *Bayesian Analysis*, 13:627–679.

Constantinou, A. C., Guo, Z., and Kitson, N. K. (2023). The impact of prior knowledge on causal structure learning. *Knowledge and Information Systems*, 65(8):3385–3434.

Crouse, D. F. (2016). On implementing 2D rectangular assignment algorithms. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4):1679–1696.

Darvariu, V.-A., Hailes, S., and Musolesi, M. (2024). Large Language Models are Effective Priors for Causal Graph Discovery.

Eggeling, R., Viinikka, J., Vuoksenmaa, A., and Koivisto, M. (2019). On Structure Priors for Learning Bayesian Networks. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, pages 1687–1695. PMLR.

Faria, G. R. A., Martins, A., and Figueiredo, M. A. T. (2022). Differentiable Causal Discovery Under Latent Interventions. In *Proceedings of the First Conference on Causal Learning and Reasoning*, pages 253–274. PMLR.

Friedman, N. and Koller, D. (2000). Being Bayesian about Network Structure. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 201–210.

Geiger, D. and Heckerman, D. (1994). Learning gaussian networks. In *Uncertainty in Artificial Intelligence*, pages 235–243. Elsevier.

Gelman, A., Hwang, J., and Vehtari, A. (2014). Understanding predictive information criteria for Bayesian models. *Statistics and Computing*, 24(6):997–1016.

Glymour, C., Zhang, K., and Spirtes, P. (2019). Review of Causal Discovery Methods Based on Graphical Models. *Frontiers in Genetics*, 10.

Good, I. J. (1950). *Probability and the Weighing of Evidence*. C. Griffin London.

Hasan, U. and Gani, M. O. (2022). Kcrl: A prior knowledge based causal discovery framework with reinforcement learning. In *Machine Learning for Healthcare Conference*, pages 691–714. PMLR.

Hasan, U. and Gani, M. O. (2023). KGS: causal discovery using knowledge-guided greedy equivalence search. *CoRR*.

Hasan, U., Hossain, E., and Gani, M. O. (2023). A Survey on Causal Discovery Methods for I.I.D. and Time Series Data. *Trans. Mach. Learn. Res.*, 2023.

Heckerman, D. (2022). A Tutorial on Learning With Bayesian Networks. (arXiv:2002.00269).

Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243.

Ibrahim, N., John, S. T., Guo, Z., and Kaski, S. (2022). Targeted causal elicitation. In *NeurIPS 2022 Workshop on Causality for Real-world Impact*.

Kitson, N. K., Constantinou, A. C., Guo, Z., Liu, Y., and Chobtham, K. (2023). A survey of Bayesian Network structure learning. *Artificial Intelligence Review*, 56(8):8721–8814.

Lindley, D. V. (1956). On a Measure of the Information Provided by an Experiment. *The Annals of Mathematical Statistics*, 27(4):986–1005.

Liu, Q. and Wang, D. (2016). Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Lorch, L., Rothfuss, J., Schölkopf, B., and Krause, A. (2021). DiBS: Differentiable Bayesian Structure Learning. In *Advances in Neural Information Processing Systems*, volume 34, pages 24111–24123. Curran Associates, Inc.

Marchant, R., Draca, D., Francis, G., Assadzadeh, S., Varidel, M., Iorfino, F., and Cripps, S. (2025). Covariate Dependent Mixture of Bayesian Networks. (arXiv:2501.05745).

Markham, A., Das, R., and Grosse-Wentrup, M. (2022). A Distance Covariance-based Kernel for Nonlinear Causal Clustering in Heterogeneous Populations. In *Proceedings of the First Conference on Causal Learning and Reasoning*, pages 542–558. PMLR.

Mikkola, P., Martin, O. A., Chandramouli, S., Hartmann, M., Pla, O. A., Thomas, O., Pesonen, H., Corander, J., Vehtari, A., Kaski, S., Bürkner, P.-C., and Klami, A. (2024). Prior Knowledge Elicitation: The Past, Present, and Future. *Bayesian Analysis*, 19(4):1129–1161.

O'Hagan, A., editor (2006). *Uncertain Judgements: Eliciting Experts' Probabilities*. Statistics in Practice. Wiley, London Hoboken, NJ.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12(null):2825–2830.

Rainforth, T., Foster, A., Ivanova, D. R., and Smith, F. B. (2024). Modern Bayesian Experimental Design. *Statistical Science*, 39(1):100–114.

Rittel, S. and Tschiatschek, S. (2023). Specifying Prior Beliefs over DAGs in Deep Bayesian Causal Structure Learning. *ECAI 2023*, pages 1962–1969.

Rittel, S. and Tschiatschek, S. (2025). Expressiveness of Parametrized Distributions over DAGs for Causal Discovery. *Transactions on Machine Learning Research*.

Saeed, B., Panigrahi, S., and Uhler, C. (2020). Causal Structure Discovery from Distributions Arising from Mixtures of DAGs. In *Proceedings of the 37th International Conference on Machine Learning*, pages 8336–8345. PMLR.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., and Mooij, J. (2012). On Causal and Anticausal Learning. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1255–1262, Edinburgh, Scotland.

Strobl, E. V. (2023). Causal discovery with a mixture of DAGs. *Machine Learning*, 112(11):4201–4225.

Sun, X., Schulte, O., Liu, G., and Poupart, P. (2023). NTS-NOTEARS: Learning Nonparametric DBNs With Prior Knowledge. arXiv:2109.04286 [cs].

Tigas, P., Annadani, Y., Ivanova, D. R., Jesson, A., Gal, Y., Foster, A., and Bauer, S. (2023). Differentiable multi-target causal bayesian experimental design. In *International Conference on Machine Learning*, pages 34263–34279. PMLR.

Toth, C., Lorch, L., Knoll, C., Krause, A., Pernkopf, F., and Peharz, R. (2022). Active Bayesian Causal Inference. *Advances in Neural Information Processing Systems*, 35:16261–16275.

Tran, Q.-D., Nguyen, P., Duong, B., and Nguyen, T. (2023). Differentiable Bayesian Structure Learning with Acyclicity Assurance. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 598–607.

Varıcı, B., Katz, D., Wei, D., Sattigeri, P., and Tajer, A. (2024). Separability Analysis for Causal Discovery in Mixture of DAGs. *Transactions on Machine Learning Research*.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski,

E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., Van Mulbregt, P., SciPy 1.0 Contributors, Vijaykumar, A., Bardelli, A. P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C. N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D. A., Hagen, D. R., Pasechnik, D. V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G. A., Ingold, G.-L., Allen, G. E., Lee, G. R., Audren, H., Probst, I., Dietrich, J. P., Silterra, J., Webber, J. T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J. L., De Miranda Cardoso, J. V., Reimer, J., Harrington, J., Rodríguez, J. L. C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N. J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P. A., Lee, P., McGibbon, R. T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T. J., Robitaille, T. P., Spura, T., Jones, T. R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y. O., and Vázquez-Baeza, Y. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Vowels, M. J., Camgoz, N. C., and Bowden, R. (2022). D'ya Like DAGs? A Survey on Structure Learning and Causal Discovery. *ACM Computing Surveys*, 55(4):1–36.

Wang, W., Hu, G., Yuan, B., Ye, S., Chen, C., Cui, Y., Zhang, X., and Qian, L. (2020). Prior-knowledge-driven local causal structure learning and its application on causal discovery between type 2 diabetes and bone mineral density. *IEEE Access*, 8:108798–108810.

Zemplenyi, M. and Miller, J. W. (2023). Bayesian optimal experimental design for inferring causal structure. *Bayesian Analysis*, 18(3):929–956.

Zhang, C., Bütepage, J., Kjellström, H., and Mandt, S. (2018). Advances in Variational Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(8):2008–2026.

Zhang, Z., Dong, J., Liu, J., and Huan, X. (2025). Goal-Oriented Sequential Bayesian Experimental Design for Causal Learning. arXiv:2507.07359 [cs].

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

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes, see Appendix A.1.]
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes, our source code is available at https://github.com/ZacBjo/VaMSL_proj, and included in the Supplementary Material.]

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   (b) Complete proofs of all theoretical results. [Yes]
   (c) Clear explanations of any assumptions. [Yes]

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes, we use boxplots for uncertainty quantification, instead of simple error bars.]
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes, experiments were run on an internal cluster with NVIDIA H100 GPUs.]

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes]

(b) The license information of the assets, if applicable. [Not Applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Not Applicable]

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Incorporating Expert Knowledge into Bayesian Causal Discovery of Mixtures of Directed Acyclic Graphs: Supplementary Materials

A APPROXIMATING FAMILIES OF DISTRIBUTIONS

We split the approximation of p into three separate parts: approximate the component assignment ($\mathbf{c}_n$), the mixture probabilities ($\boldsymbol{\pi}$), and on each component the latent embeddings with the parameters ($\mathbf{Z}_k, \Theta_k$). First, to approximate the component assignment we use a categorical distribution, initialized as uninformative. Next, for the mixture probabilities we use a Dirichlet distribution with prior parameter α, with a default value of 1. Finally, to approximate the embeddings and parameters we use a particle approximation, as the one done by Lorch et al. (2021).

A.1 Description of VaMSL

Algorithm 2 describes the procedure we employ for learning mixtures of structures. The algorithm relies on six equations, which we provide next. The full derivation of these equations is given in Appendix A.2. The algorithm consists of two different types of updates, which is enabled by a coordinate-ascent procedure. First, the updates on the mixing probabilities ($\boldsymbol{\pi}$) and assignment probabilities ($q(\mathbf{c}_n)$)—also termed responsibilities. These updates can be done explicitly as we describe in Appendix A.2.1. Next, the updates of the latent embeddings ($\mathbf{Z}_k$) — which encode soft graphs and the parameters (Θ_k). These updates are done using SVGD, and fully described in Appendix A.2.2.

Denoting by ζ the digamma function, the update of the responsibilities $q(\mathbf{c}_n)$ is given by

$$\hat{q}^*(c_{nk}=1) = \frac{\exp\left(\frac{1}{P}\sum_p \left[\ln p(\mathbf{x}_n \mid G_k^{(p)}, \Theta_k^{(p)})\right] + \zeta(\alpha_k) - \zeta(\sum_h^K \alpha_h)\right)}{\sum_{h=1}^K \exp\left(\frac{1}{P}\sum_p \left[\ln p(\mathbf{x}_n \mid G_h^{(p)}, \Theta_k^{(p)})\right] + \zeta(\alpha_h) - \zeta(\sum_m^K \alpha_m)\right)}. \tag{7}$$

The mixing weights are updated according to

$$\ln q^*(\boldsymbol{\pi}) = \sum\nolimits_k \ln \pi_k \left[\left(\alpha_k + \sum\nolimits_n \hat{q}^*(c_{nk}=1)\right) - 1\right] + \text{constant}. \tag{8}$$

The particle updates within each component $k = 1, \ldots, K$, are

$$\mathbf{Z}_k^{(p),t+1} = \mathbf{Z}_k^{(p),t} + \eta_t \phi_t^Z(\Gamma_k^{(p),t}) \tag{9}$$

$$\Theta_k^{(p),t+1} = \Theta_k^{(p),t} + \eta_t \phi_t^\Theta(\Gamma_k^{(p),t}), \tag{10}$$

where

$$\phi_t^Z(\Gamma) := \frac{1}{P}\sum_p \left[\kappa(\Gamma_k^{(p),t}, \Gamma)\nabla_{Z_k^{(p),t}} \ln q(\mathbf{Z}_k^{(p),t}, \Theta_k^{(p),t}) + \nabla_{Z_k^{(p),t}} \kappa(\Gamma_k^{(p),t}, \Gamma)\right], \tag{11}$$

$$\phi_t^\Theta(\Gamma) := \frac{1}{P}\sum_p \left[\kappa(\Gamma_k^{(p),t}, \Gamma)\nabla_{\Theta_k^{(p),t}} \ln q(\mathbf{Z}_k^{(p),t}, \Theta_k^{(p),t}) + \nabla_{\Theta_k^{(p),t}} \kappa(\Gamma_k^{(p),t}, \Gamma)\right], \tag{12}$$

using κ as a squared exponential additive kernel and $\Gamma_k^{(p),t} = (\mathbf{Z}_k^{(p),t}, \Theta_k^{(p),t})$.

Additionally, since VaMSL is dependent on initialization and component collapsing is possible, we include random restarts to ensure better results. We set two conditions for randomly restarting with reinitialized particles, 1)

Algorithm 2 Variational Mixture Structure Learning

Require: Observations $\mathcal{D} = \{\mathbf{x}_n\}_{n=1}^N$, with $\mathbf{x}_n \in \mathbb{R}^d$, number of components K, expert information $\mathcal{D}_\Psi$
Hyperparameters: mixing prior $\boldsymbol{\alpha}$, kernel κ, schedules for ω_t, β_t, and η_t, number of optimization steps T, number of coordinate-ascent updates U.
Output: Approximate distribution of $\{\mathbf{Z}_k, \Theta_k\}_{k=1}^K$; distributions $\mathrm{q}(C)$ and $\mathrm{q}(\boldsymbol{\pi})$.

1: Initialize particles $\{\mathbf{Z}_k^{(p),0}, \Theta_k^{(p),0}\}_{p=1}^P$
2: **for** $u = 1$ until $U + 1$ **do** ▷ Coordinate ascent
3: Update responsibilities $\mathrm{q}(\boldsymbol{c}_n)$ using Equation (7), for $n = 1, \ldots, N$.
4: Update mixing weights $\mathrm{q}(\boldsymbol{\pi})$ using Equation (8).
5: Sample assignments $\boldsymbol{c}'_n \sim \mathrm{q}(\boldsymbol{c}_n),\ n = 1, \ldots, N$.
6: **for** $k = 1$ until K **do**
7: **for** $t = 1$ until $T + 1$ **do** ▷ Optimization
8: Obtain particles $\{\mathbf{Z}_k^{(p),t+1}, \Theta_k^{(p),t+1}\}_{p=1}^P$ using Equations (9)-(12).
9: **end for**
10: **end for**
11: **end for**

if updating the responsibilities (line 3) numerically collapses one of the components to have zero responsibility, and 2) if after the final annealing of the particle distribution (line 11) none of the graphs $G_\infty(\mathbf{Z}_k^{(p)}), p = 1, \ldots, P$, for a given component k are acyclic. We set the maximum number of random restarts to 5 restarts, and, in the case of using an elicited informative prior, let the reinitialized model start with same prior.

The dominating step of our SVGD instantiation of VaMSL lies in computing the (approximate) gradients part of the SVGD updates in Equation (11) and Equation (12). This computational load can be addressed by instead computing the gradients using mini-batches, thereby avoiding a dominating complexity term dependent on the number of observations N. The SVGD updates also involve evaluating kernels which depend on the number of mixture components K, the number of particles P, number of covariates d, and the latent dimensionality l, and have the complexity $\mathcal{O}(KP^2dl)$. We also note that when the update of the responsibilities (given in Equation (7)) is carried out for each observation (as in our experiments) it scales linearly in N and depends on the number or components K and particles P, with a complexity of $\mathcal{O}(NKP)$.

A.2 Proof of Equalities Needed for VaMSL

As indicated in Appendix A.1, the Equations (7)-(12) are sufficient because of CAVI updates. For in depth explanations of CAVI, see Blei et al. (2017) or Bishop (2006, Chapter 10). The optimal settings (denoted by q^*) for the individual variational distributions, conditional on values of the other components are:

$$\ln q^*(\mathbf{Z}_k, \Theta_k) = \operatorname*{argmax}_{q \in \mathcal{Q}} \mathbb{E}_{q(\boldsymbol{\pi})q(C)q(Z_{\neg k}, \Theta_{\neg k})}[\ln p(\mathbf{Z}, \Theta, C, \boldsymbol{\pi}, \mathcal{D} \mid \mathcal{D}_S, \mathcal{D}_H)], \tag{13}$$

$$\ln q^*(\boldsymbol{c}_n) = \operatorname*{argmax}_{q \in \mathcal{Q}} \mathbb{E}_{q(\boldsymbol{\pi})q(C_{\neg n})q(\mathbf{Z}, \Theta)}[\ln p(\mathbf{Z}, \Theta, C, \boldsymbol{\pi}, \mathcal{D} \mid \mathcal{D}_S, \mathcal{D}_H)], \text{ and} \tag{14}$$

$$\ln q^*(\boldsymbol{\pi}) = \operatorname*{argmax}_{q \in \mathcal{Q}} \mathbb{E}_{q(C)q(\mathbf{Z}, \Theta)}[\ln p(\mathbf{Z}, \Theta, C, \boldsymbol{\pi}, \mathcal{D} \mid \mathcal{D}_S, \mathcal{D}_H)], \tag{15}$$

where $k = 1, \ldots, K$, $n = 1, \ldots, N$, and the indexing $\neg i$ refers to a product between all the distributions with indices $j \neq i$. As these terms are dependent on each other, we employ CAVI and Monte Carlo estimates to iteratively refine our variational approximations. Next we describe the computation of these terms, see Section A.2.3 for full derivations of each CAVI update.

For our approach to be valid, we make use of Proposition 1 and Equation (16) of Lorch et al. (2021) to ensure convergence of the samples we obtain with the CAVI updates. For completeness, we include them below, adapted to our notation.

Proposition 1 (Latent posterior expectation, Lorch et al. (2021)) *Under the generative model (5), with $K = 1$, it holds that:*

1. $\mathbb{E}_{p(G|\mathcal{D})}[f(G)] = \mathbb{E}_{p(\mathbf{Z}|\mathcal{D})}\left[\frac{\mathbb{E}_{p(G|\mathbf{Z})}[f(G)p(\mathcal{D}|G)]}{\mathbb{E}_{p(G|\mathbf{Z})}[p(\mathcal{D}|G)]}\right]$ *and*

2. $\mathbb{E}_{p(G,\Theta|\mathcal{D})}[f(G,\Theta)] = \mathbb{E}_{p(\mathbf{Z},\Theta|\mathcal{D})}\left[\frac{\mathbb{E}_{p(G|\mathbf{Z})}[f(G,\Theta)p(\Theta|G)p(\mathcal{D}|G,\Theta)]}{\mathbb{E}_{p(G|\mathbf{Z})}[p(\Theta|G)p(\mathcal{D}|G,\Theta)]}\right]$.

Additionally, Equation (16) of Lorch et al. (2021) indicates that when annealing $\omega \to \infty$ we obtain:

$$\begin{aligned}\mathbb{E}_{p(G|\mathcal{D})}[f(G)] &\to \mathbb{E}_{p(\mathbf{Z}|\mathcal{D})}[f(G_\infty(\mathbf{Z}))]\\ \mathbb{E}_{p(G,\Theta|\mathcal{D})}[f(G,\Theta)] &\to \mathbb{E}_{p(\mathbf{Z},\Theta|\mathcal{D})}[f(G_\infty(\mathbf{Z}),\Theta)].\end{aligned} \tag{16}$$

As a note, the equation numbers came out in a nice enough way that our label of Eq. (16) matches the label of Eq. (16) in Lorch et al. (2021). We make use of this coincidence.

Combining these results allows working with the annealed version of the soft-graph to obtain estimates on functions that depend on the graph G. However, we want to be able compute expectations with respect to latent graphs also in the mixture setting. We therefore also prove an equality for computing expectations in the graph space via latent graph embeddings in the mixture setting in Proposition 2.

Proposition 2 (Latent mixture posterior expectation) *Under the generative model (5), it holds that:*

$$\mathbb{E}_{p(G_k,\Theta_k|\mathcal{D})}[f(G,\Theta)] = \mathbb{E}_{p(\mathbf{Z}_k,\Theta_k,C|\mathcal{D})}\left[\frac{\mathbb{E}_{p(G_k|\mathbf{Z}_k)}[f(G_k,\Theta_k)p(\Theta_k \mid G_k)\prod_n p(\mathbf{x}_n \mid G_k,\Theta_k)^{c_{nk}}]}{\mathbb{E}_{p(G_k|\mathbf{Z}_k)}[p(\Theta_k \mid G_k)\prod_n p(\mathbf{x}_n \mid G_k,\Theta_k)^{c_{nk}}]}\right].$$

We provide a proof in Appendix A.2.4.

A.2.1 Exact Update of Responsibilities and Mixing Weights

Making use of Equation (16) (Lorch et al. 2021), the optimized distributions $q(\mathbf{Z},\Theta)$ can be rewritten as distributions $q(G,\Theta)$ over CBNs. The optimal update given in Equation (14) is simplified to

$$\ln q^*(\boldsymbol{c}_n) = \mathbb{E}_{q(G,\Theta)}\left[\ln p(\mathbf{x}_n \mid G,\Theta,\boldsymbol{c}_n))\right] + \mathbb{E}_{q(\boldsymbol{\pi})}\left[\ln p(\mathbf{c}_n \mid \boldsymbol{\pi})\right] + \text{constant},$$

and the optimal variational distribution for the assignment variables given current approximations $q(G,\Theta)q(\boldsymbol{\pi})$ satisfies

$$q^*(\boldsymbol{c}_n) \propto \prod\nolimits_k \exp\left(\mathbb{E}_{q(G_k,\Theta_k)}\left[\ln p(\mathbf{x}_n \mid G_k,\Theta_k)\right] + \mathbb{E}_{q(\boldsymbol{\pi})}\left[\ln \pi_k\right]\right)^{c_{nk}}, n = 1,\dots,N. \tag{17}$$

Normalizing across the components implies

$$q^*(c_{nk}=1) = \frac{\exp\left(\mathbb{E}_{q(G_k,\Theta_k)}\left[\ln p(\mathbf{x}_n \mid G_k,\Theta_k)\right] + \mathbb{E}_{q(\boldsymbol{\pi})}\left[\ln \pi_k\right]\right)}{\sum_{h=1}^K \exp\left(\mathbb{E}_{q(G_h,\Theta_h)}\left[\ln p(\mathbf{x}_n \mid G_h,\Theta_h)\right] + \mathbb{E}_{q(\boldsymbol{\pi})}\left[\ln \pi_h\right]\right)}. \tag{18}$$

Since $q(\boldsymbol{\pi})$ is a density of Dirichlet random variable (see Appendix A), use the standard result for the expectation of the logarithm a Dirichlet distributed variable: $\mathbb{E}_{q(\boldsymbol{\pi})}\left[\ln \pi_h\right] = \zeta(\alpha_k) - \zeta(\sum_h^K \alpha_h)$, where $\zeta(x) = \frac{\partial}{\partial x}\Gamma(x)$ is the digamma function. The expectation $\mathbb{E}_{q(G_k,\Theta_k)}$, appearing in both the numerator and denominator of Equation (18), is approximated with a Monte Carlo average

$$\hat{q}^*(c_{nk}=1) = \frac{\exp\left(\frac{1}{P}\sum_p\left[\ln p(\mathbf{x}_n \mid G_k^{(p)},\Theta_k^{(p)})\right] + \zeta(\alpha_k) - \zeta(\sum_h^K \alpha_h)\right)}{\sum_{h=1}^K \exp\left(\frac{1}{P}\sum_p\left[\ln p(\mathbf{x}_n \mid G_h^{(p)},\Theta_h^{(p)})\right] + \zeta(\alpha_h) - \zeta(\sum_m^K \alpha_m)\right)}. \tag{19}$$

We also note that, the expectation in Equation (18) with respect to the distribution $q(G_k,\Theta_k)$ requires the latent graph embeddings to be annealed as described above when presenting Proposition 1. If however, the assignments are updated before sufficient annealing, Proposition 2 can be employed to compute the expectations via the distributions $q(Z_k,\Theta_k)$. For our algorithms, we use Proposition 2 in this manner when updating the responsibilities, except after the final annealing of the particle distribution $q(Z_k,\Theta_k)$, in which case we compute Equation (18) using hard graphs from $q(G_\infty(Z_k),\Theta_k)$.

Due to our choice of prior $p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\alpha})$, the update of the variational distribution $q(\boldsymbol{\pi})$, given current distributions $q(\mathbf{Z},\Theta)q(C)$, is a standard result (see, for instance, Bishop 2006, Chapter 10) given by:

$$\ln q^*(\boldsymbol{\pi}) = \sum\nolimits_k \ln \pi_k\left[\left(\alpha_k + \sum\nolimits_n \hat{q}^*(c_{nk}=1)\right) - 1\right] + \text{constant}. \tag{20}$$

The optimal update is, therefore, also Dirichlet distributed and $q^*(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\alpha}^*)$, where $\alpha_k^* = \alpha_k + \sum_n \hat{q}^*(c_{nk} = 1)$.

A.2.2 Particle Algorithm for Latent Embeddings and Parameters

To approximate full posteriors using SVGD, we instantiate a set of P particles $\{\mathbf{Z}_k^{(p)}, \Theta_k^{(p)}\}_{p=1}^P$ for each component, $k = 1, \ldots, K$, and perturb each in the direction maximally decreasing the Kullback-Leibler divergence. Using $\Gamma_k^{(p),t} = (\mathbf{Z}_k^{(p),t}, \Theta_k^{(p),t})$, for steps $t = 1, \ldots, T$, we apply the following perturbation to each particle:

$$\mathbf{Z}_k^{(p),t+1} = \mathbf{Z}_k^{(p),t} + \eta_t \phi_t^Z(\Gamma_k^{(p),t}) \quad \text{and} \quad \Theta_k^{(p),t+1} = \Theta_k^{(p),t} + \eta_t \phi_t^\Theta(\Gamma_k^{(p),t}), \tag{21}$$

where $p = 1, \ldots, P$, $k = 1, \ldots, K$, and η_t is a stepsize. The transforms ϕ_t^Z and ϕ_t^Θ take the form:

$$\phi_t^Z(\cdot) := \frac{1}{P} \sum_p \left[\kappa(\Gamma_k^{(p),t}, \cdot) \nabla_{Z_k^{(p),t}} \ln \mathrm{q}(Z_k^{(p),t}, \Theta_k^{(p),t}) + \nabla_{Z_k^{(p),t}} \kappa(\Gamma_k^{(p),t}, \cdot) \right], \tag{22}$$

$$\phi_t^\Theta(\cdot) := \frac{1}{P} \sum_p \left[\kappa(\Gamma_k^{(p),t}, \cdot) \nabla_{\Theta_k^{(p),t}} \ln \mathrm{q}(Z_k^{(p),t}, \Theta_k^{(p),t}) + \nabla_{\Theta_k^{(p),t}} \kappa(\Gamma_k^{(p),t}, \cdot) \right], \tag{23}$$

where the gradients are estimated using samples $\boldsymbol{c}_n^{(s)} \sim q(\boldsymbol{c}_n), s = 1, \ldots, S, n = 1, \ldots, N$ as shown in Equation (25) and Equation (26). The kernel κ needs to be positive-definite and Lorch et al. (2021) suggest using the additive SE kernel:

$$\kappa\left((\mathbf{Z}, \Theta), (\mathbf{Z}', \Theta')\right) := \exp\left(-\frac{\|\mathbf{Z} - \mathbf{Z}'\|_2^2}{\gamma_Z}\right) + \exp\left(-\frac{\|\Theta - \Theta'\|_2^2}{\gamma_\Theta}\right), \tag{24}$$

where γ_Z and γ_Θ are length-scales.

Note that the particles $\mathbf{Z}_k^{(p)}$ can be interpreted as continuous relaxations for unique graphs $G_k^{(p)}$ as the temperature ω in Equation (5) goes to infinity. To ensure that these graphs are DAGs, the temperature β of the latent prior in Equation (6) is annealed $\beta \to \infty$ over the course of the optimization as well. The updated distributions $q(\mathbf{Z}_k, \Theta_k), k = 1, \ldots, K$ can thus be rewritten as distributions over CBNs: $q(G_k, \Theta_k), k = 1, \ldots, K$, where each graph particle $G_k^{(p)}$ is a DAG and satisfies $p_\infty(G_k^{(p)} \mid \mathbf{Z}_k^{(p)}, E_k) = 1$, by alluding to Proposition 1 (Lorch et al. 2021).

A.2.3 Derivations of CAVI Updates

The results presented above depend on the gradients $\nabla_{Z_k} \ln q(\mathbf{Z}_k, \Theta_k)$, $\nabla_{Z_k^{(p)}} \ln q(Z_k^{(p)}, \Theta_k^{(p)})$, $\nabla_{\Theta_k} \ln q(\mathbf{Z}_k, \Theta_k)$, $\nabla_{\Theta_k^{(p)}} \ln q(Z_k^{(p)}, \Theta_k^{(p)})$, as well as the updates for the variational responsibilities $q(\boldsymbol{c}_n)$ and mixing weights $q(\boldsymbol{\pi})$. We present the derivations of these gradients below. Note that the CAVI updates for the responsibilities and mixing weights are calculated with respect to graph distributions $q(G_k, \Theta_k)$ as (continuous) latent embedding space maps unto the discrete graph space as the temperature in equation (5) is annealed, $\omega \to \infty$. We suppress the conditioning on expert information $\mathcal{D}_{S,k}, \mathcal{D}_{H,k}, k = 1, \ldots, K$ for clarity.

Component Latent Graph Embedding Gradient The gradient of $\ln q(\mathbf{Z}_k, \Theta_k)$ with respect to the latent graph embeddings is obtained by:

$$\begin{aligned}
&\nabla_{Z_k} \ln q(\mathbf{Z}_k, \Theta_k) \\
&= \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(\mathbf{Z}, \Theta, C, \boldsymbol{\pi}, \mathcal{D})\right] \\
&= \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(\Theta, \mathcal{D} \mid \mathbf{Z}, C))\right] \quad \text{(Expand generative model.)} \\
&\quad + \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(\mathbf{Z})\right] \\
&\quad + \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(C \mid \boldsymbol{\pi})\right] \\
&\quad + \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(\boldsymbol{\pi})\right] \\
&= \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(\Theta, \mathcal{D} \mid \mathbf{Z}, C)\right] \quad \text{(Remove terms independent of } \nabla_{Z_k}.) \\
&\quad + \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C) \prod_{h \neq k} q(Z_h, \Theta_h)} \left[\ln p(\mathbf{Z})\right]
\end{aligned}$$

$$
\begin{aligned}
&= \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\sum\nolimits_m \ln p(\Theta_m, \mathcal{D} \mid Z_m, C)\right] \\
&\quad + \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\sum\nolimits_m \ln p(Z_m)\right] \\
&= \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)\right] \quad \text{(Terms where } m \neq k \text{ don't affect the gradient.)} \\
&\quad + \nabla_{Z_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\mathbf{Z}_k)\right] \\
&= \nabla_{Z_k} \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\ln p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)\right] + \nabla_{Z_k} \ln p(\mathbf{Z}_k) \quad \text{(Factor out independent expectations.)} \\
&= \nabla_{Z_k} \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\ln \left[\sum\nolimits_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k, \mathcal{D} \mid G_k, C)\right]\right] \quad \text{(From generative model.)} \\
&\quad + \nabla_{Z_k} \ln p(\mathbf{Z}_k) \\
&= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\nabla_{Z_k} \ln \left[\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k, \mathcal{D} \mid G_k, C)\right]\right]\right] \\
&\quad + \nabla_{Z_k} \ln p(\mathbf{Z}_k) \\
&= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\nabla_{Z_k} \ln \left[\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod\nolimits_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}\right]\right]\right] \\
&\quad + \nabla_{Z_k} \ln p(\mathbf{Z}_k) \\
&= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\frac{\nabla_{Z_k} \mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}\right]}{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}\right]}\right] \quad \left(\nabla \ln f(x) = \frac{\nabla f(x)}{f(x)}.\right) \\
&\quad + \nabla_{Z_k} \ln p(\mathbf{Z}_k)
\end{aligned}
$$

With samples $\{\boldsymbol{c}'_n\}_{n=1}^N, s = 1, \ldots, S$ drawn from the variational distribution $q(C)$, we can compute a Monte Carlo estimate of the gradient:

$$
\nabla_{Z_k} \ln q(\mathbf{Z}_k, \Theta_k) \approx \nabla_{Z_k} \ln p(\mathbf{Z}_k) + \frac{1}{S} \sum_s \frac{\nabla_{Z_k} \mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}^{(s)}}\right]}{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}^{(s)}}\right]}. \tag{25}
$$

Component Parameter Gradient The gradient of $\ln q(\mathbf{Z}_k, \Theta_k)$ with respect to the parameters is obtained by:

$$
\begin{aligned}
&\nabla_{\Theta_k} \ln q(\mathbf{Z}_k, \Theta_k) \\
&= \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\mathbf{Z}, \Theta, C, \boldsymbol{\pi}, \mathcal{D})\right] \\
&= \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\Theta, \mathcal{D} \mid \mathbf{Z}, C))\right] \quad \text{(Expand generative model.)} \\
&\quad + \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\mathbf{Z})\right] \\
&\quad + \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(C \mid \boldsymbol{\pi})\right] \\
&\quad + \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\boldsymbol{\pi})\right] \\
&= \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\Theta, \mathcal{D} \mid \mathbf{Z}, C)\right] \quad \text{(Remove terms independent of } \nabla_{\Theta_k}.) \\
&= \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\sum\nolimits_m \ln p(\Theta_m, \mathcal{D} \mid \mathbf{Z}_m, C)\right] \\
&= \nabla_{\Theta_k} \mathbb{E}_{q(\boldsymbol{\pi})q(C)\prod_{h\neq k} q(Z_h,\Theta_h)} \left[\ln p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)\right] \quad \text{(Terms where } m \neq k \text{ do not affect the gradient.)} \\
&= \nabla_{\Theta_k} \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\ln p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)\right] \quad \text{(Factor out independent expectations.)} \\
&= \nabla_{\Theta_k} \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\ln \left[\sum\nolimits_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k, \mathcal{D} \mid G_k, C)\right]\right] \quad \text{(From generative model.)} \\
&= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\nabla_{\Theta_k} \ln \left[\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k, \mathcal{D} \mid G_k, C)\right]\right]\right] \\
&= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\nabla_{\Theta_k} \ln \left[\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod\nolimits_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}\right]\right]\right]
\end{aligned}
$$

$$= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\frac{\nabla_{\Theta_k} \mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}} \right]}{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}} \right]} \right] \quad (\nabla \ln f(x) = \frac{\nabla f(x)}{f(x)}.)$$

$$= \mathbb{E}_{\prod_n q(\boldsymbol{c}_n)} \left[\frac{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[\nabla_{\Theta_k} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}} \right] \right]}{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}} \right]} \right]$$

With samples $\{\boldsymbol{c}'_n\}_{n=1}^N, s = 1, \ldots, S$ drawn from the variational distribution $q(C)$, we can compute a Monte Carlo estimate of the gradient:

$$\nabla_{\Theta_k} \ln q(\mathbf{Z}_k, \Theta_k) \approx \frac{1}{S} \sum_s \frac{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[\nabla_{\Theta_k} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}^{(s)}} \right] \right]}{\mathbb{E}_{p(G_k|\mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}^{(s)}} \right]}. \tag{26}$$

Variational Responsibilities The variational responsibilities are obtained by:

$$\begin{aligned}
&\ln q^*(\boldsymbol{c}_n) \\
&= \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(G, \Theta, C, \boldsymbol{\pi}, \mathcal{D}) \right] \\
&= \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(\mathcal{D} \mid G, \Theta, C)) \right] \quad \text{(Expand generative model.)} \\
&\quad + \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(\Theta \mid G) \right] \\
&\quad + \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(G) \right] \\
&\quad + \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(C \mid \boldsymbol{\pi}) \right] \\
&\quad + \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(\boldsymbol{\pi}) \right] \\
&= \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(\mathcal{D} \mid G, \Theta, C)) \right] \qquad \text{(Terms independent of } q(\boldsymbol{c}_n).) \\
&\quad + \mathbb{E}_{q(G,\Theta)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\ln p(C \mid \boldsymbol{\pi}) \right] + \text{constant} \\
&= \mathbb{E}_{\prod_k q(G_k,\Theta_k)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\sum_j \sum_k c_{jk} \ln p(x_j \mid G_k, \Theta_k) \right] \\
&\quad + \mathbb{E}_{\prod_k q(G_k,\Theta_k)q(\boldsymbol{\pi}) \prod_{i \neq n} q(\boldsymbol{c}_i)} \left[\sum_j \sum_k c_{jk} \ln \pi_k \right] + \text{constant} \\
&= \mathbb{E}_{\prod_k q(G_k,\Theta_k)} \left[\sum_k c_{nk} \ln p(\mathbf{x}_n \mid G_k, \Theta_k) \right] \quad \text{(Terms where } j \neq n \text{ are constant and} \\
&\quad + \mathbb{E}_{q(\boldsymbol{\pi})} \left[\sum_k c_{nk} \ln \pi_k \right] + \text{constant} \qquad \text{(Expectations independent of the terms can be factored out.)} \\
&= \sum_k c_{nk} \left[\mathbb{E}_{q(G_k,\Theta_k)} \left[\ln p(\mathbf{x}_n \mid G_k, \Theta_k) \right] + \mathbb{E}_{q(\boldsymbol{\pi})} \left[\ln \pi_k \right] \right] + \text{constant}
\end{aligned}$$

The above equality implies the proportionality:

$$q^*(\boldsymbol{c}_n) \propto \prod_k \exp \left[\mathbb{E}_{q(G_k,\Theta_k)} \left[\ln p(\mathbf{x}_n \mid G_k, \Theta_k) \right] + \mathbb{E}_{q(\boldsymbol{\pi})} \left[\ln \pi_k \right] \right]^{c_{nk}}.$$

From which the marginal assignment probabilities are computed by normalizing across the components using the softmax function, such that:

$$q^*(c_{nk} = 1) = \gamma(c_{nk}) = \frac{\exp \left[\mathbb{E}_{q(G_k,\Theta_k)} \left[\ln p(\mathbf{x}_n \mid G_k, \Theta_k) \right] + \mathbb{E}_{q(\boldsymbol{\pi})} \left[\ln \pi_k \right] \right]}{\sum_{h=1}^K \exp \left[\mathbb{E}_{q(G_h,\Theta_h)} \left[\ln p(\mathbf{x}_n \mid G_h, \Theta_h) \right] + \mathbb{E}_{q(\boldsymbol{\pi})} \left[\ln \pi_h \right] \right]}.$$

Complete Derivation of Mixing Weight Updates The mixing weights are updated by:

$$\begin{aligned}
&\ln q^*(\boldsymbol{\pi}) \\
&= \mathbb{E}_{q(G,\Theta)q(C)} \left[\ln p(G, \Theta, C, \boldsymbol{\pi}, \mathcal{D}) \right]
\end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}_{q(G,\Theta)q(C)}\left[\ln p(\Theta, \mathcal{D} \mid G, C))\right] \quad \text{(Expand generative model.)} \\
&\quad + \mathbb{E}_{q(G,\Theta)q(C)}\left[\ln p(G)\right] \\
&\quad + \mathbb{E}_{q(G,\Theta)q(C)}\left[\ln p(C \mid \boldsymbol{\pi})\right] \\
&\quad + \mathbb{E}_{q(G,\Theta)q(C)}\left[\ln p(\boldsymbol{\pi})\right] \\
&= + \mathbb{E}_{q(G,\Theta)q(C)}\left[\ln p(C \mid \boldsymbol{\pi})\right] \qquad \text{(Terms independent of } \boldsymbol{\pi} \text{ are constant w.r.t. to } q(\boldsymbol{\pi})\text{.)} \\
&\quad + \mathbb{E}_{q(G,\Theta)q(C)}\left[\ln p(\boldsymbol{\pi})\right] + \text{constant} \\
&= + \mathbb{E}_{q(C)}\left[\ln p(C \mid \boldsymbol{\pi})\right] \quad \text{(Expectations independent of the terms can be factored out.)} \\
&\quad + \ln p(\boldsymbol{\pi}) + \text{constant} \\
&= + \mathbb{E}_{\prod_i q(\boldsymbol{c}_i)}\left[\sum\nolimits_n \sum\nolimits_k c_{nk} \ln \pi_k\right] \qquad (p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\alpha})\text{.)} \\
&\quad + \ln \frac{1}{B(\boldsymbol{\alpha})} + \sum_k (\alpha_k - 1) \ln \pi_k + \text{constant} \\
&= + \sum\nolimits_n \sum\nolimits_k \mathbb{E}_{q(c_n)}\left[c_{nk} \ln \pi_k\right] \qquad \text{(Terms where } i \neq n \text{ and } \ln \frac{1}{B(\boldsymbol{\alpha})} \text{ are constant w.r.t. } q(\boldsymbol{\pi})\text{.)} \\
&\quad + \sum_k (\alpha_k - 1) \ln \pi_k + \text{constant} \\
&= \sum\nolimits_k \ln \pi_k \left[(\alpha_k - 1) + \sum\nolimits_n \mathbb{E}_{q(c_n)}[c_{nk}]\right] + \text{constant} \\
&= \sum\nolimits_k \ln \pi_k \left[(\alpha_k + \sum\nolimits_n \gamma(c_{nk})) - 1\right] + \text{constant} \quad \text{(As } \mathbb{E}_{q(c_n)}[c_{nk}] = q^*(c_{nk} = 1) = \gamma(c_{nk})\text{.)}
\end{aligned}
$$

Showing the update is a Dirichlet distribution $q^*(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\alpha}^*)$, where $\alpha_k^* = \alpha_k + \sum_n \gamma(c_{nk})$.

A.2.4 Derivation of Proposition 2

Below we derive the latent mixture posterior expectation which connects the expectation of a function with respect to a joint distribution over graphs G_k and parameters Θ_k with an expectation with respect to a distribution over latent graph embeddings $\mathbf{Z}_k$, parameters Θ_k, and assignments C.

$$
\begin{aligned}
&\mathbb{E}_{p(G_k,\Theta_k \mid \mathcal{D})}[f(G_k, \Theta_k)] \\
&= \int_{\Theta_k} \sum_{G_k} p(G_k, \Theta_k \mid \mathcal{D}) f(G_k, \Theta_k) \, \mathrm{d}\Theta_k \\
&= \int_{\Theta_k} \sum_{G_k} \sum_{C} p(G_k, \Theta_k, C \mid \mathcal{D}) f(G_k, \Theta_k) \, \mathrm{d}\Theta_k \\
&= \int_{\Theta_k} \sum_{G_k} \sum_{C} \frac{p(C) p(G_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C)}{p(\mathcal{D})} f(G_k, \Theta_k) \, \mathrm{d}\Theta_k \\
&= \int_{\Theta_k} \sum_{G_k} \sum_{C} \int_{\mathbf{Z}_k} \frac{p(C) p(\mathbf{Z}_k) p(G_k \mid \mathbf{Z}_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C)}{p(\mathcal{D})} f(G_k, \Theta_k) \, \mathrm{d}\Theta_k \, \mathrm{d}\mathbf{Z}_k \\
&\qquad \text{as } \; p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D}) = \tfrac{p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C) p(C) p(\mathbf{Z}_k)}{p(\mathcal{D})} \iff \tfrac{p(C) p(\mathbf{Z}_k)}{p(\mathcal{D})} = \tfrac{p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D})}{p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)}, \\
&= \int_{\Theta_k} \sum_{G_k} \sum_{C} \int_{\mathbf{Z}_k} \frac{p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D}) p(G_k \mid \mathbf{Z}_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C)}{p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)} f(G_k, \Theta_k) \, \mathrm{d}\Theta_k \, \mathrm{d}\mathbf{Z}_k \\
&= \int_{\mathbf{Z}_k} \int_{\Theta_k} \sum_{C} p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D}) \frac{\sum_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C) f(G_k, \Theta_k)}{p(\Theta_k, \mathcal{D} \mid \mathbf{Z}_k, C)} \, \mathrm{d}\Theta_k \, \mathrm{d}\mathbf{Z}_k
\end{aligned}
$$

$$= \int_{\mathbf{Z}_k} \int_{\Theta_k} \sum_C p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D}) \frac{\sum_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C) f(G_k, \Theta_k)}{\sum_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k, \mathcal{D} \mid G_k, C)} \, \mathrm{d}\Theta_k \, \mathrm{d}\mathbf{Z}_k$$

$$= \int_{\mathbf{Z}_k} \int_{\Theta_k} \sum_C p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D}) \frac{\sum_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C) f(G_k, \Theta_k)}{\sum_{G_k} p(G_k \mid \mathbf{Z}_k) p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, C)} \, \mathrm{d}\mathbf{Z}_k \, \mathrm{d}\Theta_k$$

$$= \int_{\mathbf{Z}_k} \int_{\Theta_k} \sum_C p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D}) \frac{\mathbb{E}_{p(G_k \mid \mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, \Theta_k, C) f(G_k, \Theta_k) \right]}{\mathbb{E}_{p(G_k \mid \mathbf{Z}_k)} \left[p(\Theta_k \mid G_k) p(\mathcal{D} \mid G_k, C) \right]} \, \mathrm{d}\mathbf{Z}_k \, \mathrm{d}\Theta_k$$

$$= \mathbb{E}_{p(\mathbf{Z}_k, \Theta_k, C \mid \mathcal{D})} \left[\frac{\mathbb{E}_{p(G_k \mid \mathbf{Z}_k)} [f(G_k, \Theta_k) p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}]}{\mathbb{E}_{p(G_k \mid \mathbf{Z}_k)} [p(\Theta_k \mid G_k) \prod_n p(\mathbf{x}_n \mid G_k, \Theta_k)^{c_{nk}}]} \right].$$

B USER MODEL FOR IMAGINARY OBSERVATIONS

We employ the device of *imaginary observations* (Consonni et al. 2018, Good 1950) to formalize the background knowledge $\mathcal{K}_{ij}$ an expert has about the existence of a causal effect $i \to j$ as a set of idealized trials $\mathcal{K}_{ij} = (k, n)$ confirming (refuting) said effect. Starting from a beta distributed prior belief $p(\psi_{ij}) = \mathrm{Beta}(\psi_{ij} \mid \alpha_0, \beta_0)$ with the mode $\psi^*_{ij,0} = \frac{\alpha_0 - 1}{\alpha_0 + \beta_0 - 2}$, where $\alpha_0, \beta_0 > 1$ for the mode to be defined, the expert's edge belief is formalized as the mode of the posterior belief $p(\psi_{ij} \mid \mathcal{K}_{ij}) \propto p(\mathcal{K}_{ij} \mid \psi_{ij}) p(\psi_{ij})$, conditioned on the aforementioned trials.

Due to our assumed beta-binomial conjugate model and imaginary observations from deterministic and correct experiments for confirming (refuting) causal effects, we know the mode of the posterior ψ^*_{ij} (expressed by the expert) must have moved in the direction supported by the observations. We can therefore derive a mapping from the posterior mode to imaginary observations by rearranging the conjugate update of the posterior mode, relating the posterior mode to the prior mode as well as the number of confirming and disconfirming trials:

$$\psi^*_{ij} = \frac{\alpha_0 + k - 1}{\alpha_0 + k + \beta_0 + (n - k) - 2}.$$

When the posterior mode has increased relative to the prior mode $\psi^*_{ij} > \psi^*_{ij,0}$, the trials are confirmatory and we can conclude $k = n$. While the opposite case $\psi^*_{ij} < \psi^*_{ij,0}$ indicates the trials disconfirm the existence of a causal effect and hence $k = 0$. Solving for these case we get the following mapping from expert belief to imaginary observations:

$$f_{\alpha_0, \beta_0}(\psi^*_{ij}) = \begin{cases} \mathcal{K}_{ij} = (n_{ij} = \left\lfloor \frac{\psi^*_{ij}(\alpha_0 + \beta_0 - 2) - \alpha_0 + 1}{1 - \psi^*_{ij}} \right\rfloor, k_{ij} = n_{ij}), & \text{if } \psi^*_{ij} > \psi^*_{ij,0}; \\ \mathcal{K}_{ij} = (n_{ij} = \left\lfloor \frac{\alpha_0 - 1 - \psi^*_{ij}(\alpha_0 + \beta_0 - 2)}{\psi^*_{ij}} \right\rfloor, k_{ij} = 0), & \text{if } \psi^*_{ij} < \psi^*_{ij,0}. \end{cases}$$

Note that, as the prior parameters α_0, β_0 determine how many observations it would take for the prior mode to shift to the mode expressed by the expert, they effectively determine the informativeness of the informative graph prior. With low values constituting a low informativeness and high values making the prior very informative. In our experiments we use the same prior parameters for all edges, if desired these could vary over the edges.

C IMPLEMENTATION OF INFERENCE FOR THE USER MODEL

The construction described in Section 2.2 yields an elicitation prior that deterministically drives particles in the direction of edge inclusion or exclusion that the expert has endorsed. This is comparable to the method proposed by Geiger and Heckerman (1994), which scores networks based on their distance from a reference prior network. However, to include stochasticity in a sample of particles from the posterior we propose sampling separate subsets of the elicited responses, termed elicitation matrices, for each particle. For each particle $\mathbf{Z}^{(p)}$ to be optimized, we sample an elicitation matrix $E^{(p)} \mid \mathcal{D}_S$, by independently sampling its entries $E^{(p)}_{ij}$ each of which takes one of two possible values: ψ^*_{ij} and 0.5, with $P(E^{(p)}_{ij} = \psi^*_{ij}) = \max(\psi^*_{ij}, 1 - \psi^*_{ij})$. In other words, the closer a response from the expert is to 0 or 1 (i.e., the stronger the opinion), the more probable it is to be part of a particle's elicitation matrix. Whereas weakly opinionated responses (i.e., close to 0.5) are more probable to not be included (i.e., labeled "unresponded" and $\equiv 0.5$). Rather than deterministically biasing the inference towards the expert opinion, this leads the proportion of particles in the particle distribution with priors to (exclude) include an edge

to correspond to the probability the expert has responded with. For a gradient based method of inference, such as the method proposed in Section 3, the elicited prior gradient for each particle is therefore:

$$\nabla_{\mathbf{Z}} \ln p(\mathbf{Z}^{(p)} \mid f(E^{(p)})) = \nabla_{\mathbf{Z}} \ln p(\mathbf{Z}^{(p)} \mid \mathcal{D}_T^{(p)}) = \nabla_{\mathbf{Z}} \ln p(\mathbf{Z}^{(p)}) + \nabla_{\mathbf{Z}} \ln p(\mathcal{D}_T^{(p)} \mid \mathbf{Z}^{(p)}),$$

which is straightforward to add to the SVGD implementation of DiBS provided by Lorch et al. (2021).

D APPROXIMATING THE EXPECTED INFORMATION GAIN

Bayesian experimental design (Rainforth et al. 2024) defines a method for choosing optimal experiments (with respect to a given utility) under uncertainty. Given a parameter distribution $p(\mathbf{Z} \mid \mathcal{D})$ and a simulator $p(\psi_{ij}^* \mid \mathbf{Z}, \xi_{ij})$, we apply the BED framework to pick optimal queries about causal edges for the expert based on the information theoretic utility of each query. The optimal query maximizes the expected information gain, which can be expressed as:

$$\underset{\xi_{ij} \in \Xi_k}{\operatorname{argmax}} \operatorname{EIG}(\xi_{ij}) = \underset{\xi_{ij} \in \Xi}{\operatorname{argmax}} \mathbb{E}_{p(\mathbf{Z}|\mathcal{D})p(\psi_{ij}^*|\mathbf{Z},\xi_{ij})} \left[\ln p(\psi_{ij}^* \mid \mathbf{Z}, \xi_{ij}) - \ln p(\psi_{ij}^* \mid \xi_{ij})\right].$$

However, computing the EIG is intractable and we instead need to approximate it. As the domain of our response variables $\psi_{ij}^* \in [0, 1]$ is continuous, we propose using the nested Monte Carlo estimator (NMC) $\hat{\mu}_{NMC}$ which can be computed by sampling from our parameter distribution and simulator. Using samples $\mathbf{Z}^{(p)} \sim p(\mathbf{Z} \mid \mathcal{D}),\ p = 1, \ldots, P$ and $\psi_{ij}^{*(s)}, \mathbf{Z}^{(s)} \sim p(\psi_{ij}^* \mid \mathbf{Z}, \xi_{ij}) p(\mathbf{Z} \mid \mathcal{D}),\ s = 1, \ldots, S$, the NMC-estimate can be computed as:

$$\hat{\mu}_{NMC} = \frac{1}{S} \sum_{s=1}^{S} \ln \frac{p(\psi_{ij}^{*(s)} \mid \mathbf{Z}^{(s)}, \xi_{ij})}{\frac{1}{P} \sum_{p=1}^{P} p(\psi_{ij}^{*(s)} \mid \mathbf{Z}^{(p)}, \xi_{ij})}. \tag{27}$$

We note that if the expert only needed to respond with hard constraints (e.g., $\psi_{ij}^* \in \{0, 1\}$), it would be possible to enumerate the expectation over $p(\psi_{ij}^* \mid \mathbf{Z}, \xi_{ij})$ in the EIG. It would then be possible to instead approximate the EIG using a Rao-Blackwellized estimator and a Bernoulli distributed simulator. We refer the reader to Rainforth et al. (2024) for further details on BED and approximating the EIG.

E ADDITIONAL DETAILS OF NUMERICAL AND BREAST CANCER EXPERIMENTS

E.1 Metrics Used for Experiments

The structured Hamming distance (SHD) measures how different two graphs are. This distance is given by counting the number of incorrect edges, meaning the number of changes (flips, additions or removals) of edges to move from one of the graphs to the other.

For (out-of-sample) classification accuracy and predictive performance in the synthetic case we generate a held-out dataset $\mathcal{D}_{ho} = \{\tilde{\mathbf{x}}_n \mid \tilde{\mathbf{x}}_n \in \mathbb{R}^{20}\}_{n=1}^{100}$ generated from a single ground truth BN in the homogeneous setting (Section 4.3) and, in the heterogeneous setting (Section 4.2), from two BNs with mixing probabilities $(0.5, 0.5)$. In the breast cancer labeling assignment (Section 4.4) we use 10-fold cross validation. To compute the classification accuracy and predictive performance we use the component ordering of the components that maximizes the (in-sample) MAP classification accuracy using the modified Jonker-Volgenant algorithm[3] described by Crouse (2016). The ESHD for heterogeneous experiments is the sum of the ESHDs for each component with respect to the ground truth graph indicated by the ordering given by maximizing the in-sample MAP classification accuracy.

We assess the predictive performance using the MAP classifications of out-of-sample observations to components (denoted $\tilde{k}(n) = \underset{k}{\operatorname{argmax}}\ q(\tilde{c}_{nk} = 1)$) and computing the *(MAP) negative log point-wise predictive density* (Gelman et al. 2014):

$$\text{MAP Neg. } \mathcal{LPPD}(\mathcal{D}_{ho}) = \sum_k \sum_n \mathbf{1}_{\{k=\tilde{k}(n)\}} \log \mathbb{E}_{p(G_k, \Theta_k|\mathcal{D})} \left[p(\tilde{\mathbf{x}}_n \mid G_k, \Theta_k)\right].$$

[3]implemented in Python SciPy (Virtanen et al. 2020).

E.2 Settings for Numerical Experiments

We instantiate VaMSL with Gaussian BNs parametrized by linear or non-linear mean functions and observational noise set to $\sigma^2 = 0.1$. Where non-linearity is instantiated using 2-layer neural networks (NN) with 5 neurons per hidden layer and ReLU activation functions. All the parameters for the NNs and linear models are randomly set with a standard normal distribution. For all the experiments we take a total 6000 gradient steps for SVGD. The step size η_t is set adaptively using RMSProp[4] with a learning rate of 0.005. In the mixture case we do 10 CAVI updates, iteratively updating first the SVGD particle distribution and, subsequently, the responsibilities and the mixing weight distribution. For the first 9 CAVI updates the SVGD particle optimization consist of 50 gradient steps while the last one consists of the final 5500 steps. The annealing schedules for the temperatures are set such that for step t we use $\beta_t = t$ as well as $\omega_t = 0.2t$ for linear BNs and $\omega_t = 0.02t$ for non-linear BNs. For the length-scales of the SVGD kernel, given in Equation (24), we use $\gamma_Z = 5$ as well as $\gamma_\Theta = 500$ for linear BNs and $\gamma_\Theta = 1000$ for non-linear BNs. When updating the responsibilities, for the first 9 CAVI updates, we approximate the expectation over $q(G_k, \Theta_k)$ in Equation (18) using latent graphs embeddings, as enabled by Proposition 2, and use all particles available. As explained in Section A.1, it might be necessary to randomly restart VaMSL for optimal performance, in our experiments we use a maximum of 5 random restarts.

For the querying strategy we use 60 particles for inference, for the mixture of graphs we use 60 particles per component for VaMSL and 120 particles for DiBS. In the cancer labeling task we use DiBS with 120 particles to infer the MAP graphs used as expert information and we use 15 particles per component when inferring the mixture model. For the observation noise For all experiments, we set $\ell = d$. To compute the approximate gradients for VaMSL, given in Equation (25) and Equation (26), we use four Monte Carlo samples. Otherwise we use values reflecting the default values of the DiBS implementation[5] of Lorch et al. (2021). We do not use their weighted particle mixture, termed DiBS+.

For the informative prior we set $\alpha_0 = \beta_0 = 10$ in the synthetic experiments $\alpha_0 = \beta_0 = 1000$ in the real-world experiments. When calculating the EIG of a query, through Equation (27), we use all particles available.

For the GMM we use default values and initialize the parameters from randomly selected data points. For the causal k-means we use the implementation[6] given by Markham et al. (2022), using the default value (0.1) for the significance level. Experiments were run on NVIDIA H100 GPUs.

Figure 1 features inference with 100 observations from a two component mixture, with mixing probabilities (0.5, 0.5), of linear Gaussian BNs with ER graphs over 4 variables and an expectation of 1 edge per node. VaMSL was initiated with one correct soft constraint per component about a random edge's existence. Otherwise all settings were the same as the default settings listed for synthetic data above.

E.3 Expert Oracle

To assess the effect of incorporating expert information we use an expert oracle. In the synthetic case, the oracle's answers are based on the ground truth graph that was used to generate the synthetic data, while in the real-world data setting the answers are based on the MAP graphs obtained when running DiBS on the homogeneous populations of a subset of the breast cancer data not used in the presented experiments. For our real world experiments we used half of the data to infer the graphs for the expert oracle and the other half for benchmarking our method.

The expert oracle returns probabilities given edge queries by sampling a value from a beta distribution parametrized by a mean and a variance parameter. The mean is determined by the reliability $r \in [0, 1]$ of the expert and the binary value indicating the edge's existence in the corresponding expert graph. It is calculated according to $\left|r - 1 + \mathbf{1}_{(i \to j)}\right|$. The variance is set to 0.05 and the default value for the reliability is $r = 0.9$.

[4] find an implementation in the Python JAX package (Bradbury et al. 2018) at http://github.com/jax-ml/jax.
[5] found at https://github.com/larslorch/dibs.
[6] found at https://causal.dev/code/fibroblast_clustering.py.

E.4 Additional Results of Erdős-Rényi Numerical Experiments

In this subsection we provide supplementary numerical results to the example shown in the main text. These results use an Erdős-Rényi graph-prior, like the one in the main text, which corresponds to $p(G_\omega(\mathbf{Z}_k)) \propto q^{\|G_\omega(\mathbf{Z}_k)\|_1}(1-q)^{\binom{d}{2}-\|G_\omega(\mathbf{Z}_k)\|_1}$, where q is the probability for any given edge to exist.

In Figure 7 we show the metrics presented in the main text and Appendix E.1 for the linear Gaussian Erdős Rényi graphs. These metrics are computed for out-of-sample data, trained using the same setting described in the main text in Section 4 with a linear parametrization. The results show how VaMSL + expert performs much better than the competing baselines in terms of structure learning as assessed by ESHD (left panels). In terms of predictive power (top-right) assessed by the log-predictive density it performs about as good as the gold-standard case of having the labels (DiBS+labels). Similarly for classification accuracy (bottom-right), both VaMSL with and without expert perform just as good as having the labels, and perform much better than both classical DiBS and the Gaussian mixture model.

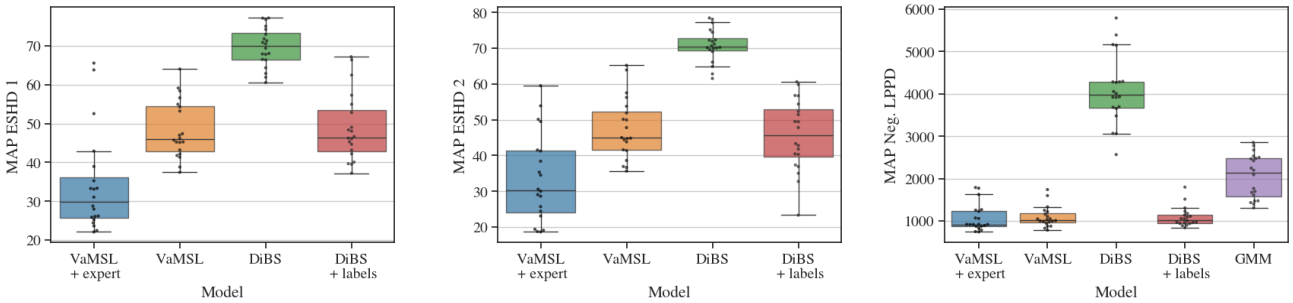

Figure 7: Boxplots and values of metrics in held-out observations, simulated with **linear** Gaussian ER BNs. *Left*: ESHD in the first component, *Center*: ESHD in the second component, *Right*: predictive power by the log point-wise predictive density. ESHD not available for Causal k-means and GMMs.

Figure 8 shows how in the non-linear case the same conclusion is reached, meaning that the type of parametrization is not affecting our conclusions. Structure learning is performing better, even than DiBS with labels—the gold standard case. Regarding the predictions, the log-predictive density shows a similar performance and even better than the GMM, with a better performance when the expert is included.

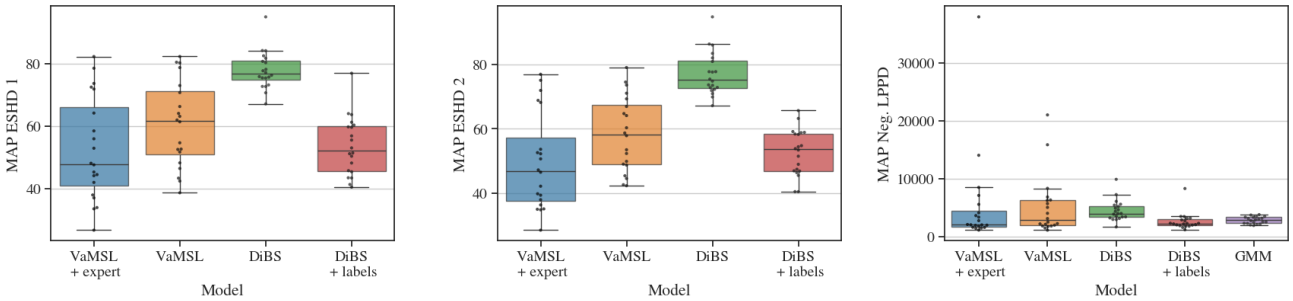

Figure 8: Boxplots of metrics of interest for simulations in **non-linear** Gaussian ER BNs. *Top-left*: ESHD in the first component, *Bottom-left*: ESHD in the second component, *Right-top*: predictive power by the log point-wise predictive density, *Bottom-right*: percentage of correctly labeled samples. ESHD and neg-lppd not available for Causal k-means and GMMs.

E.5 Additional Results in Scale-free Numerical Experiments

In this subsection we provide additional numerical results for the scale-free setting, where the density corresponds to $p(G_\omega(\mathbf{Z}_k)) \propto \prod_{i=1}^{d}(1 + \|G_\omega(\mathbf{Z}_i)\|_1)^{-3}$, where $G_\omega(\mathbf{Z})_i$ is the i-th row of $G_\omega(\mathbf{Z})$.

Figures 9 and 10 display how our proposed approach still performs better than the competing approaches. Our proposed method, VaMSL + expert outperforms all the other approaches in structure learning, and performs on par to the ideal scenario: classification when the labels are available at training time.

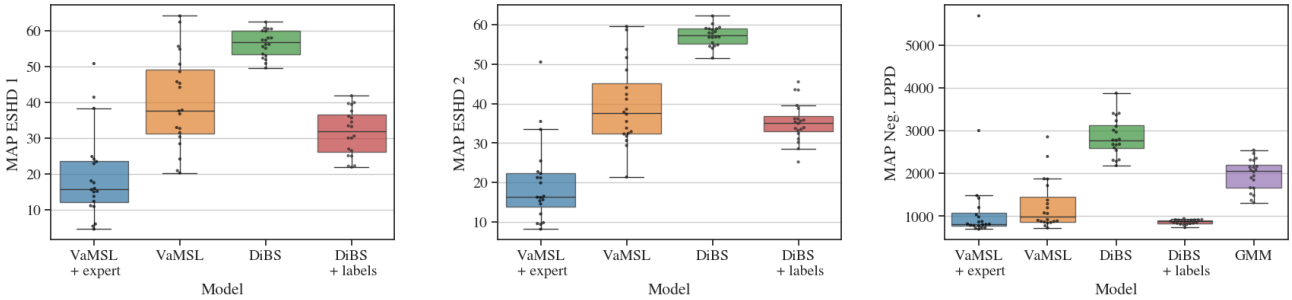

Figure 9: Boxplots of metrics of interest for simulations in **linear** Gaussian SF BNs. *Left*: ESHD in the first component, *Center*: ESHD in the second component, *Right*: predictive power by the neg. lppd. ESHD and neg-lppd not available for Causal k-means and GMMs. Neg. lppd not available for Causal k-means.

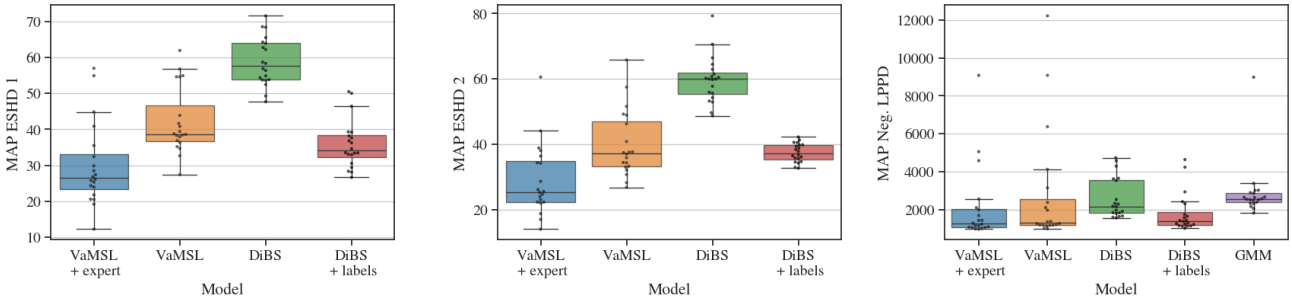

Figure 10: Boxplots of metrics of interest for simulations in **non-linear** Gaussian SF BNs. *Left*: ESHD in the first component, *Center*: ESHD in the second component, *Right*: predictive power by the neg. lppd. ESHD not available for Causal k-means and GMMs. Neg. lppd not available for Causal k-means.

In Figure 11 we display how our BED querying strategy has a marked improvement over the random approach, this applies to both possibilities considered, with varying number of queries (top) and with different levels of expert reliability (bottom), in both linear (left) and non-linear settings.

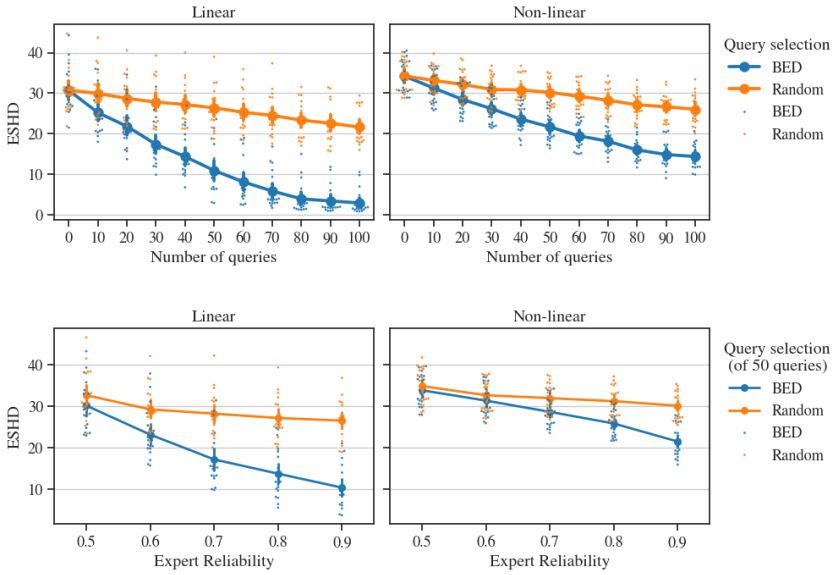

Figure 11: Comparison between BED and random querying of the expert with varying number of queries (*top*) and varying expert reliability (*bottom*) when inferring Gaussian SF BNs, displayed by average performance with 95% bootstrapped confidence interval, and the obtained ESHD. For both **linear** (*left*) and **non-linear** graphs (*right*)

E.6 Additional Results and Details of Breast Cancer Dataset

To complement the main result in Section 4.4, in Figure 12 we display other metrics of interest in the applied setting of the breast cancer dataset. Namely, the average MAP ESHD (left), the MAP neg. lppd (center), and the classification accuracy (right). These plots repeat the main message from the simulated experiments: VaMSL + expert performs much better in terms of structure learning and classification accuracy, while preserving a competitive predictive power.

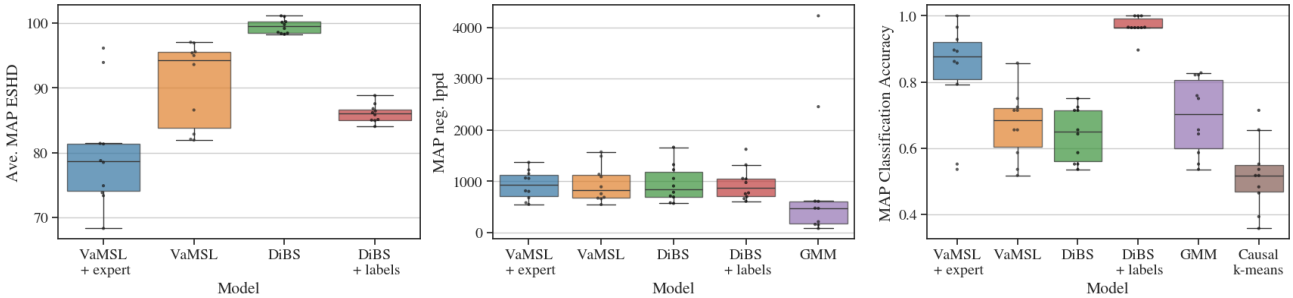

Figure 12: Boxplots and values for metrics of interest in out-of-sample data for breast cancer data set in methods of interest. *Left*: average ESHD, *Center*: predictive power assessed by the log point-wise predictive density, *Right*: percentage of correctly labeled samples. ESHD and neg. lppd not available for Causal k-means, and ESHD not available for GMMs.

E.6.1 Expert Queries in Breast Cancer Dataset

In Figure 13 we showcase the performance of VaMSL with an increasing number of queries in the breast cancer dataset. The left panel shows how the graph distance diminishes with the number of queries, as is to be expected. Specifically, it shows how the posterior graphs become closer to the assumed graph as the number of queries increases. The center panel shows how despite an increased performance in the graph distribution (left panel), the raw predictive power — as assessed by the MAP negative log-predictive density (see Appendix E.1) is not increasing. Nevertheless, the correct classification is still being obtained as shown in the right panel. We suspect that the vast dimensionality of the parameter space (since we are in non-linear models) and possible multi-modal topography of the loss function explains the lack of improved performance in the log-predictive density domain. However, our two main concerns do improve with an increased number of queries: obtaining the correct graph structure (ESHD), and obtaining an improved classification (classification accuracy).

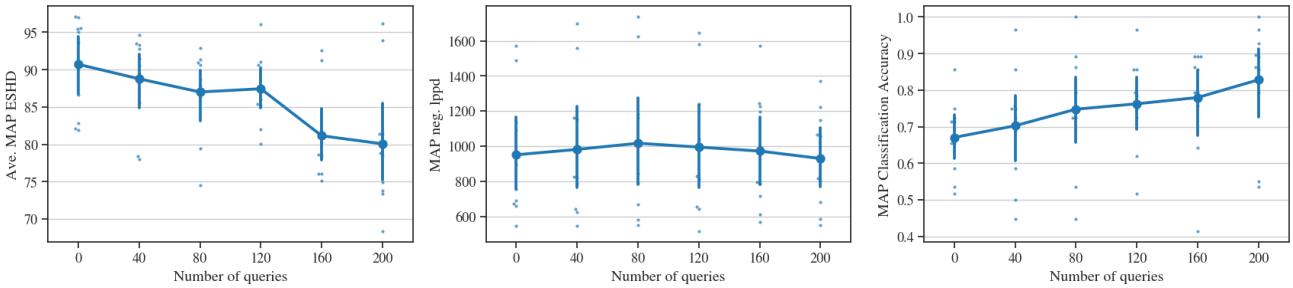

Figure 13: Mean and 95% (bootstrapped) confidence interval of metrics for out-of-sample data using VaMSL with the BED querying strategy in the breast cancer dataset, modeled with SF priors and non-linear models. Assessed by (*left*) average ESHD, (*center*) MAP negative log-pointwise predictive density, and (*right*) classification accuracy.

E.6.2 Breast Cancer Dataset Modeled with Erdő-Rényi Priors

To confirm the performance of the model in the applied setting with the ER priors, we perform the same routines as in Section 4.4. In Figure 14 we replicate Figure 12 using ER priors instead of the SF priors used in the main text. The same conclusion holds, albeit at a smaller magnitude. The expert-informed model (VaMSL + expert) outperforms the other models in terms of classification (right panel), Additionally, the structure learning (left panel) is vastly better, and the predictive performance (center) is on-par to the other methods.

In Figure 15 we show how additional queries to the expert do improve the method in the two main metrics of interest: labeling accuracy (right) and structure learning. As such, validating our query strategy.

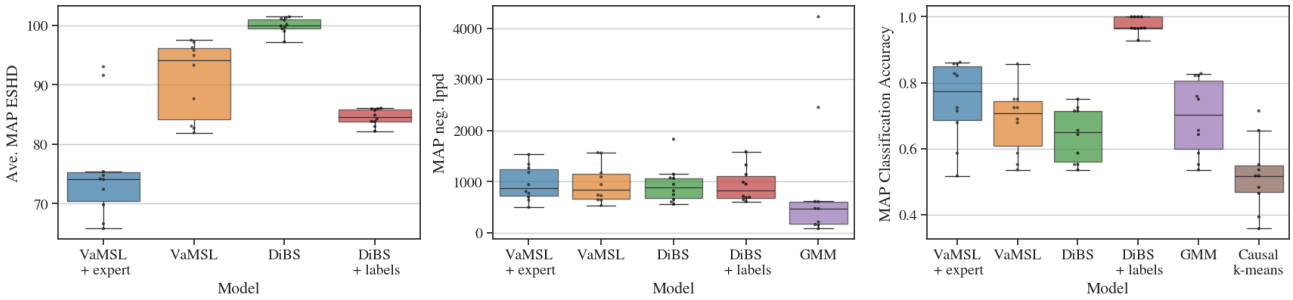

Figure 14: Boxplots and values for metrics of interest in out-of-sample data for breast cancer data set in methods of interest using ER priors. *Left*: average ESHD, when compared to a perfect expert model, *Center*: predictive power assessed by the log point-wise predictive density, *Right*: percentage of correctly labeled samples. ESHD and neg. lppd not available for Causal k-means, and ESHD not available for GMMs.

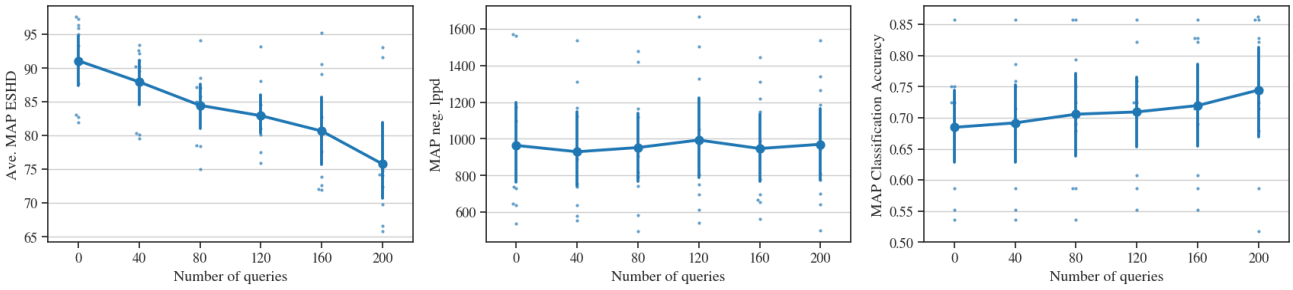

Figure 15: Mean and 95% (bootstrapped) confidence interval of metrics for out-of-sample data using VaMSL with the BED querying strategy in the breast cancer dataset, modeled with ER priors and non-linear models. Assessed by (*left*) average ESHD, (*center*) MAP negative log-pointwise predictive density, and (*right*) classification accuracy.

E.7 Ablation Studies on Hyperparameters

Below we demonstrate the effect of varying the informativeness of our proposed (latent) graph prior. The informativeness is controlled by the hyperparameters α_0, β_0 which, while these could vary edge-wise if the expert had opinions of different strength for given edges, we take to be equal for all edges and $\alpha_0 = \beta_0$. As can be seen from 16, the results of the inference are more dramatically impacted when using a higher informativeness. The experimental setup is otherwise identical to that used in Section 4.3.

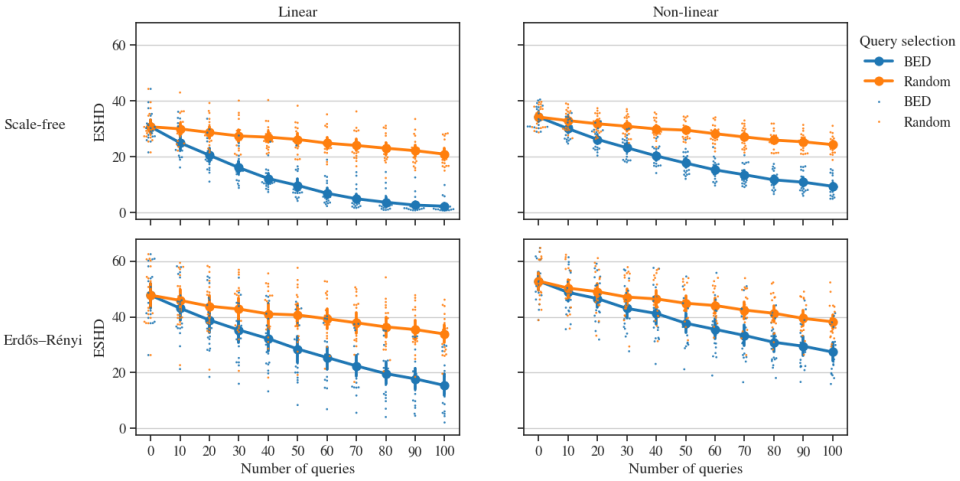

Figure 16: Mean and 95% (bootstrapped) confidence interval of ESHD for VaMSL with more informative prior using hyperparameters $\alpha_0 = \beta_0 = 50$. Top panels correspond to scale-free graphs and bottom panels to Erdős-Rényi. Left panels show linear parametrizations. Right panels show non-linear parametrizations.

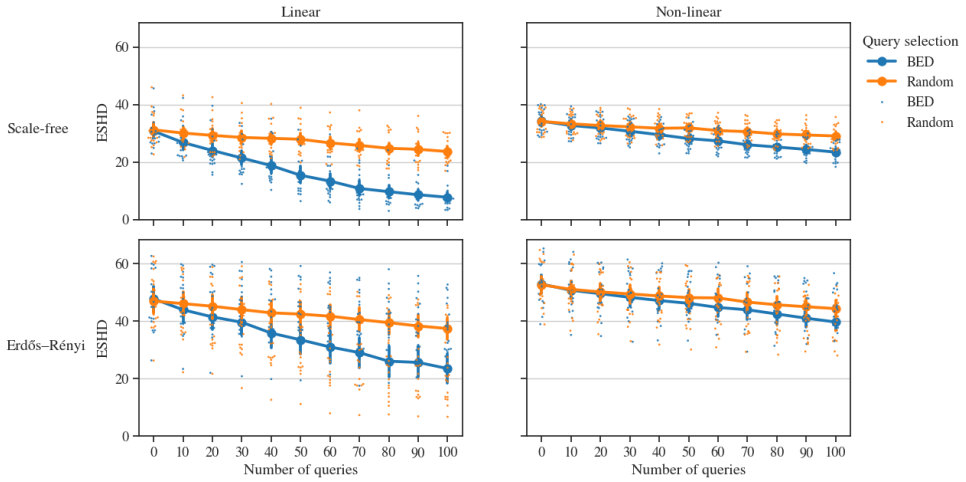

Figure 17: Mean and 95% (bootstrapped) confidence interval of ESHD for VaMSL with less informative prior using hyperparameters $\alpha_0 = \beta_0 = 2$. Top panels correspond to scale-free graphs and bottom panels to Erdős-Rényi. Left panels show linear parametrizations. Right panels show non-linear parametrizations.

E.8 Ablation Studies on α_0 and Reliability

Figures 18 and 19 display the performance (measured by ESHD) of different degrees of informativeness, as encoded by α_0, fare against different levels of reliability. When the expert is highly reliable, the informativeness of the prior does not seem to have a significant effect, and we obtain similar performance through out. On the other hand, for lower levels of expert-reliability, the informativeness of the parameter is key to improve the performance. Other than the expert reliability and α_0, the experimental settings are identical to those in in Section 4.4.

The decreased structure learning when the reliability is set to 0.5 and the informativeness is increased is due to the implementation of the expert oracle (as explained in Section E.3). When the oracle's mean is set to 0.5 the distribution for the responses will be equally probable to give slightly incorrect answers as correct ones, leading to the increased informativeness of the prior emphasizing incorrect beliefs about the ground truth graph.

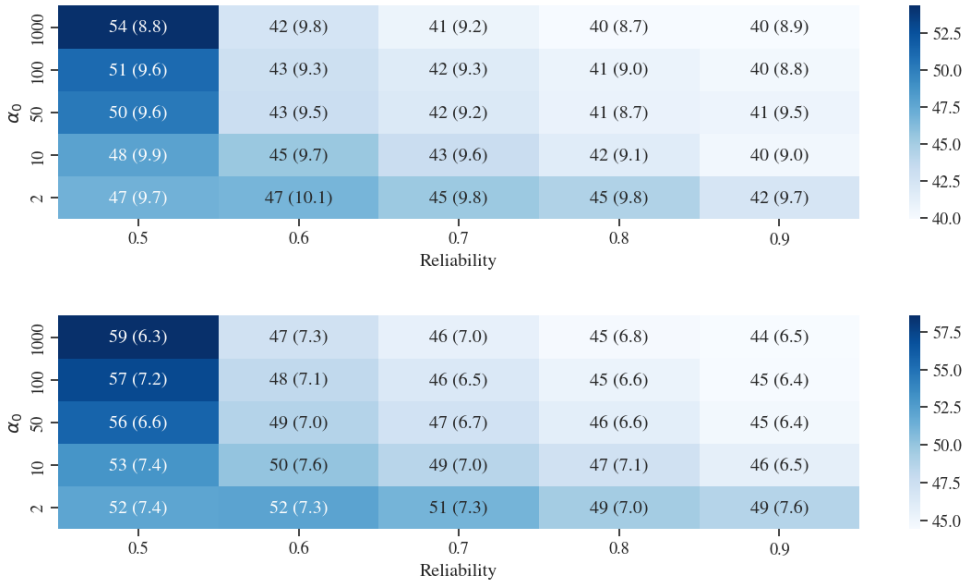

Figure 18: Average ESHD (SD) for varying levels of α_0 and expert reliability in ER graphs, over 20 independent runs. For **linear** (*top*) and **non-linear** graphs (*bottom*).

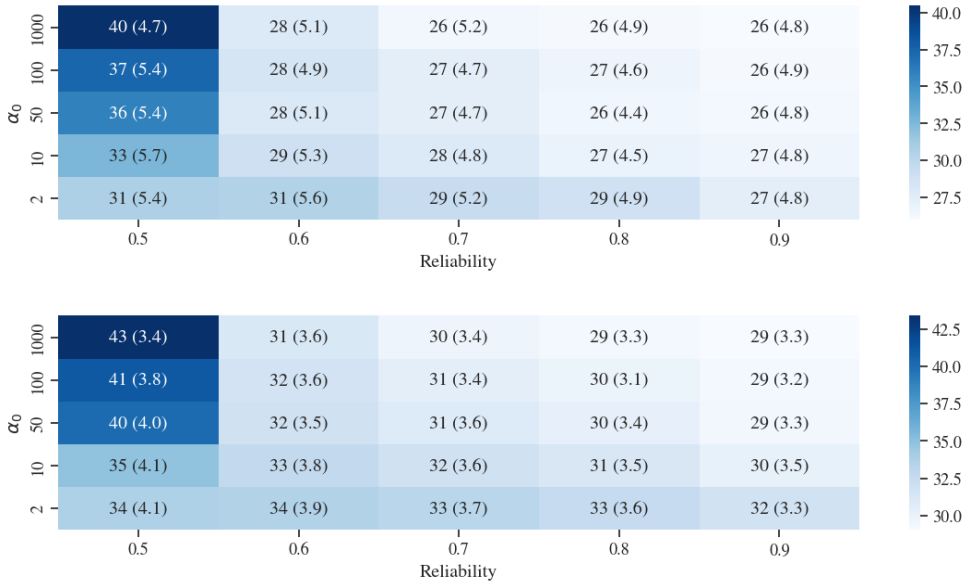

Figure 19: Average ESHD (SD) for varying levels of α_0 and expert reliability in SF graphs, over 20 independent runs. For **linear** (*top*) and **non-linear** graphs (*bottom*).

E.9 Ablation Study of the Expert Model

In Tables 1 and 2 we compare our proposed expert model using soft-constraints against NTS-NOTEARS (Sun et al. 2023), which needs binary responses. This showcases how binarized responses from an expert fail to correctly capture the nuances that continuous responses are able to capture. The approach we follow is to map the expert's responses, originally in $(0, 1)$, to $\{0, 1\}$ by using a rounding function. In principle, this should lead to an advantage to NTS-NOTEARS as the given responses are correct and hard constraints have a bigger impact than soft constraints. However, as displayed below, in this setting VaMSL outperforms NTS-NOTEARS irrespective of the number of queries.

Table 1: Average (SD) of (E)SHD for **nonlinear** Gaussian ER BNs with 20 variables across 10 replications with randomly selected queries. NTS-NOTEARS was run with the Additive Noise Model (ANM) and default settings. All methods using a single component.

Method	0 queries	20 queries	40 queries	60 queries	80 queries	100 queries
VaMSL + expert	51 (8.6)	47 (7.6)	45 (7.7)	44 (6.8)	41 (7.2)	38 (5.9)
NTS-NOTEARS	59 (6.2)	57 (4.4)	54 (7.0)	54 (5.7)	50 (5.9)	51 (9.5)

Table 2: Means and standard deviations of (E)SHD for **nonlinear** Gaussian SF BNs with 20 variables across 10 replications with randomly selected queries. NTS-NOTEARS was run with the Additive Noise Model (ANM) and default settings. All methods using a single component.

Method	0 queries	20 queries	40 queries	60 queries	80 queries	100 queries
VaMSL + expert	34 (4.2)	32 (3.5)	30 (3.4)	29 (3.2)	27 (3.4)	25 (2.9)
NTS-NOTEARS	83 (7.2)	77 (6.1)	73 (7.6)	68 (5.4)	64 (8.5)	55 (6.8)

E.10 Computational Scalability of VaMSL to Higher Dimensions

Tables 3-6 show the computational scalability of VaMSL on two components to higher number of variables by displaying the running times for the different combinations of nonlinear-linear and scale-free and Erdős-Rényi graphs. Overall, the results match the scale of running times stated in Appendix A.1.

Table 3: Means and standard deviations of computation times (sec.) for **linear** Gaussian ER mixtures with two components across 10 replications.

N particles	10 Variables	20 Variables	50 Variables
10	71 (2.1)	106 (2.2)	444 (328.4)
30	146 (2.8)	265 (4.3)	928 (6.5)

Table 4: Means and standard deviations of computation times (sec.) for **nonlinear** Gaussian ER mixtures with two components across 10 replications.

N particles	10 Variables	20 Variables	50 Variables
10	402 (147.5)	1113 (1189.4)	2960 (2061.3)
30	775 (269.3)	2042 (1210.3)	10269 (6277.1)

Table 5: Means and standard deviations of computation times (sec.) for **linear** Gaussian **SF** mixtures with two components across 10 replications.

N particles	10 Variables	20 Variables	50 Variables
10	70 (1.8)	107 (1.4)	338 (4.0)
30	173 (4.4)	306 (6.1)	1779 (428.4)

Table 6: Means and standard deviations of computation times (sec.) for **nonlinear** Gaussian **SF** mixtures with two components across 10 replications.

N particles	10 Variables	20 Variables	50 Variables
10	316 (93.1)	761 (534.4)	2219 (776.1)
30	813 (309.6)	2346 (1546.8)	7100 (3576.2)

E.11 Behavior of VaMSL against Misspecification of the Number of Components

In Tables 7-10 we show how VaMSL behaves when the number of components is misspecified. We consider two possible sources of model misspecification. First, there are two components in the true data generating process and the model incorrectly assumes there are three mixture components. Second, there are three components in the data generating process, and the model assumes there are two mixture components. In each experiment, the components in the models have an equal share of a total of 60 particles. For these experiments, we report the weighted ESHD, defined as:

$$\text{wESHD} = \sum_n \sum_i \sum_k c_{ni} q(c_{nk} = 1) \text{ESHD}(p(G_k \mid \mathcal{D}), G_i^*),$$

where G_i^* refers to the ground truth graph of the ith DGP and $c_{ni} = 1$ when the observation $\mathbf{x}_n$ was generated from the DGP associated with G_i^*, while otherwise $c_{ni} = 0$.

The results show how, although choosing to few components will adversely affect the inference, when there are more components than DGPs the inference does not suffer. This is due to the mixture model either degenerating the responsibilities of the additional component, rendering the mixture correctly specified, or using it to infer the same graph as one of the other components thereby not negatively impacting the inference.

Table 7: Means and standard deviations of weighted ESHD for **linear** Gaussian ER mixtures across 10 replications. All mixture components have the same probability (1/number of components).

Number of DGPs	1 Component	2 Components	3 Components
2	72 (3.8)	44 (6.3)	45 (8.5)
3	71 (2.6)	68 (7.0)	55 (10.7)

Table 8: Means and standard deviations of weighted ESHD for **nonlinear** Gaussian ER mixtures across 10 replications. All mixture components have the same probability (1/number of components).

Number of DGPs	1 Component	2 Components	3 Components
2	77 (7.5)	62 (11.3)	57 (10.1)
3	84 (5.1)	72 (11.2)	60 (11.8)

Table 9: Means and standard deviations of weighted ESHD for **linear** Gaussian SF mixtures across 10 replications. All mixture components have the same probability (1/number of components).

Number of DGPs	1 Component	2 Components	3 Components
2	58 (3.2)	45 (10.8)	42 (9.4)
3	64 (2.2)	56 (4.2)	47 (6.7)

Table 10: Means and standard deviations of weighted ESHD for **nonlinear** Gaussian SF mixtures across 10 replications. All mixture components have the same probability (1/number of components).

Number of DGPs	1 Component	2 Components	3 Components
2	60 (6.7)	44 (12.6)	41 (9.2)
3	78 (10.4)	62 (12.6)	45 (9.1)