

Appendices

Table of Contents

A Further Discussion of Related Work	18
B Background on Gaussian processes	19
C Derivation of the Information Gain Utility Functions	20
C.1 Information Gain for General Queries	20
C.2 Derivation of the Causal Discovery Utility Function	20
C.3 Derivation of the Causal Model Learning Utility Function	21
C.4 Derivation of the Causal Reasoning Utility Function	23
D Approximate Inference and Experimental Details	24
D.1 Estimating Posterior Marginal Interventional Likelihoods	24
D.2 Sampling Ground Truth Graphs	24
D.3 Normal-Inverse-Gamma Prior for Root Nodes	25
D.4 Gamma Priors for GP Hyperparameters of Non-Root Nodes	25
D.5 DiBS for Approximate Posterior Graph Inference	25
D.6 Estimation of the Information Gain Utility Functions	25
D.7 Bayesian Optimisation for Experimental Design	26
E Implementation Details	27
E.1 Particle Resampling	27
E.2 Shared Priors and Caching of Marginal Likelihoods	27
E.3 Computational Complexity	28
E.4 Implementation and Computing Resources	29
F Evaluation Metrics	30
G Extended Experimental Results	32

A Further Discussion of Related Work

In this section, we further discuss the most closely related prior works, which also consider a Bayesian active learning approach for causal discovery. These methods are summarised and contrasted with ABCI in Tab. 1. Similar to our approach, they also all assume acyclicity and causal sufficiency.

Table 1: Comparison of ABCI with closely related active Bayesian causal discovery methods in terms of the learning objective, that is, the causal target query, and the considered model class.

Work	Target Query	Model Class
Tong and Koller [79], Murphy [50]	causal graph G	Conjugate Dirichlet-Multinomial
Cho et al. [11]	causal graph G	Conjugate linear Gaussian-inverse-Gamma
Agrawal et al. [1]	some function $\phi(G)$ of the causal graph G	Linear Gaussian
Tigas et al. [78]	causal graph G and param- eters of f_i	Additive Gaussian noise with parametric neural network functions f_i
GP-DiBS-ABCI (ours)	some function $q(\mathcal{M})$ of the full SCM \mathcal{M}	Additive Gaussian noise with nonpara- metric functions f_i modeled by GPs

The early experimental design work by Tong and Koller [79] and Murphy [50] already investigated active causal discovery from a Bayesian perspective. They focused on the case in which all variables are multinomial to allow for tractable, closed-form posterior inference with a conjugate Dirichlet prior.

The setting with continuous variables was not explored from an active Bayesian causal discovery perspective until the work of Cho et al. [11], who consider the linear Gaussian case in the context of biological networks. Cho et al. [11] similarly use an inverse-Gamma prior to enable closed-form posterior inference. In these approaches, experiment selection targets the full causal graph. Agrawal et al. [1] extend the work of Cho et al. [11] by enabling the active learning of some function of the causal graph and handling interventional budget constraints.

Similarly to our approach, the concurrent work by Tigas et al. [78] models nonlinear causal relationships with additive Gaussian noise in the active learning setting. However, they are limited to targeting the full SCM for experiment design, which corresponds to our q_{CML} objective. In addition, their approach does not quantify the uncertainty in the functions conditional on a causal graph sampled from the graph posterior. In contrast, our nonparametric approach both directly models the epistemic uncertainty in the functions and mitigates the risk of model misspecification by jointly learning the kernel hyperparameters. Moreover, our method is Bayesian over the unknown noise variances, which are usually unknown in practice. It is unclear whether Tigas et al. [78] hand-specify a constant noise variance a priori, or whether they infer it jointly with the function parameters [78, cf. § 5.4.1].

Other related work by Shanmugam et al. [70] considers the problem of finding the minimal number of perfectly informative (w.r.t. conditional independences induced by the true underlying graph) multi-target interventions to fully identify the true causal graph. In contrast, we assume that only finitely many data points are available per experiment/intervention. Thus, we try to perform at each time step (possibly repeating) interventions to maximally reduce our uncertainty in the target causal query (which may be the causal graph). As another point of difference, we also optimise for the actual intervention value, whereas Shanmugam et al. [70] only optimise for the intervention targets.

B Background on Gaussian processes

We use Gaussian Processes (GPs) to model mechanisms of non-root nodes X_i , i.e., we place a GP prior on $p(f_i | G)$. In the following, we give some background on GPs and how to compute probabilistic quantities thereof relevant to this work. For further information on GPs we refer the reader to Williams and Rasmussen [82].

A $\mathcal{GP}(m_i(\cdot), k_i^G(\cdot, \cdot))$ is a collection of random variables, any finite number of which have a joint Gaussian distribution, and is fully determined by its mean function $m_i(\cdot)$ and covariance function (or kernel) $k_i^G(\cdot, \cdot)$, where

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})], \quad \text{and} \quad k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]. \quad (\text{B.1})$$

In our experiments, we choose the mean function $m_i(x) \equiv 0$ to be zero and a rational quadratic kernel

$$k_{RQ}(\mathbf{x}, \mathbf{x}') = \kappa_i^o \cdot \left(1 + \frac{1}{2\alpha} (\mathbf{x} - \mathbf{x}')^\top \kappa_i^l (\mathbf{x} - \mathbf{x}')\right)^{-\alpha} \quad (\text{B.2})$$

as our covariance function. Here, α denotes a weighting parameter, κ_i^o denotes an output scale parameter and κ_i^l denotes a length scale parameter. For the weighting parameter, we use a default value of $\alpha = \log 2 \approx 0.693$. For κ_i^l and κ_i^o we choose priors according to Appx. D.4. In Section 4.1 we summarise both parameters as $\boldsymbol{\kappa}_i = (\kappa_i^o, \kappa_i^l)$.

In this work, we consider Gaussian additive noise models (see Eq. (4.1)). Hence, for a given non-root node X_i in some graph G , we have

$$p(X_i | \mathbf{pa}_i^G, f_i, \sigma_i^2, G) = \mathcal{N}(X_i | f_i(\mathbf{pa}_i^G), \sigma_i^2) \quad (\text{B.3})$$

where \mathbf{pa}_i^G denotes the parents of X_i in G . For some batch of collected data $\mathbf{x} = \{\mathbf{x}^n\}_{n=1}^N$, let $\mathbf{x}_i = (x_i^1, \dots, x_i^N)^T$, $\mathbf{pa}_i^G = (\mathbf{pa}_i^{G,1}, \dots, \mathbf{pa}_i^{G,N})$, and \mathbf{K} the Gram matrix with entries $K_{m,n} = k_{RQ}(\mathbf{pa}_i^{G,m}, \mathbf{pa}_i^{G,n})$. Then, we can compute the prior marginal log-likelihood, which is needed to compute $p(\mathbf{x}^{1:t} | G)$, in closed form as

$$\log p(\mathbf{x}_i | \mathbf{pa}_i^G, \sigma_i^2, G) = \log \mathbb{E}_{f_i | G} [p(\mathbf{x}_i | \mathbf{pa}_i^G, f_i, \sigma_i^2, G)] \quad (\text{B.4})$$

$$= -\frac{1}{2} \mathbf{x}_i^T (K + \sigma^2 I)^{-1} \mathbf{x}_i - \frac{1}{2} \log |K + \sigma^2 I| - \frac{N}{2} \log 2\pi. \quad (\text{B.5})$$

To predict the function values $f_i(\widetilde{\mathbf{pa}}_i^G)$ at unseen test locations $\widetilde{\mathbf{pa}}_i^G = (\widetilde{\mathbf{pa}}_i^{G,1}, \dots, \widetilde{\mathbf{pa}}_i^{G,\tilde{N}})$ given previously observed data \mathbf{x} , let \mathbf{K}^\dagger be the $(\tilde{N} \times N)$ covariance matrix with entries $K_{m,n}^\dagger = k_{RQ}(\widetilde{\mathbf{pa}}_i^{G,m}, \mathbf{pa}_i^{G,n})$ and $\tilde{\mathbf{K}}$ be the $(\tilde{N} \times \tilde{N})$ covariance matrix with entries $\tilde{K}_{m,n} = k_{RQ}(\widetilde{\mathbf{pa}}_i^{G,m}, \widetilde{\mathbf{pa}}_i^{G,n})$. Then, the predictive posterior is multivariate Gaussian

$$p(f_i(\widetilde{\mathbf{pa}}_i^G) | \widetilde{\mathbf{pa}}_i^G, \mathbf{x}, \sigma_i^2, G) = \mathcal{N}(\boldsymbol{\mu}_f, \boldsymbol{\Sigma}_f) \quad (\text{B.6})$$

with mean

$$\boldsymbol{\mu}_f = \mathbf{K}^\dagger [\mathbf{K} + \sigma_i^2 I]^{-1} \mathbf{x}_i \quad (\text{B.7})$$

and covariance

$$\boldsymbol{\Sigma}_f = \tilde{\mathbf{K}} - \mathbf{K}^\dagger [\mathbf{K} + \sigma_i^2 I]^{-1} \mathbf{K}^\dagger. \quad (\text{B.8})$$

Finally, the marginal posterior over observations \tilde{X}_i , which is needed to sample and evaluate candidate experiments in the experimental design process, is given by

$$p(\tilde{X}_i | \widetilde{\mathbf{pa}}_i^G, \mathbf{x}, \sigma_i^2, G) = \mathcal{N}(\boldsymbol{\mu}_{X_i}, \boldsymbol{\Sigma}_{X_i}) \quad (\text{B.9})$$

with mean

$$\boldsymbol{\mu}_{X_i} = \boldsymbol{\mu}_f \quad (\text{B.10})$$

and covariance

$$\boldsymbol{\Sigma}_{X_i} = \boldsymbol{\Sigma}_f + \sigma_i^2 I. \quad (\text{B.11})$$

C Derivation of the Information Gain Utility Functions

In the following, we provide the derivations for the expressions presented in Section 4.2.

C.1 Information Gain for General Queries

We show that

$$\arg \max_{a_t} I(Y; \mathbf{X}^t | \mathbf{x}^{1:t-1}) = \arg \max_{a_t} U(a_t) \quad (\text{C.1})$$

for $U(a_t)$ given in Eq. (4.11).

Proof. We write the mutual information in the following form

$$I(Y; \mathbf{X}^t | \mathbf{x}^{1:t-1}) = H(Y | \mathbf{x}^{1:t-1}) + H(\mathbf{X}^t | \mathbf{x}^{1:t-1}) - H(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}). \quad (\text{C.2})$$

In the above, we expand the joint entropy of experiment outcome and query as

$$H(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}) = -\mathbb{E}_{Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1})] \quad (\text{C.3})$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{Y, \mathbf{X}^t | \mathcal{M}} [\log p(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1})]] \quad (\text{C.4})$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{Y, \mathbf{X}^t | \mathcal{M}} [\log \mathbb{E}_{\mathcal{M}' | \mathbf{x}^{1:t-1}} [p(Y | \mathcal{M}') \cdot p(\mathbf{X}^t | \mathcal{M}')]]] \quad (\text{C.5})$$

for any query such that query and experiment outcome are conditionally independent given an SCM. This holds true, e.g., whenever Y is a deterministic function of \mathcal{M} such as $Y = q_{\text{cd}}(\mathcal{M}) = G$.

The marginal entropy of the experiment outcome given previously observed data is

$$H(\mathbf{X}^t | \mathbf{x}^{1:t-1}) = -\mathbb{E}_{\mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | \mathbf{x}^{1:t-1})] \quad (\text{C.6})$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log p(\mathbf{X}^t | \mathbf{x}^{1:t-1})]] \quad (\text{C.7})$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log \mathbb{E}_{\mathcal{M}' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | \mathcal{M}')]]] \quad (\text{C.8})$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log \mathbb{E}_{f', \sigma^{2'}, G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | f', \sigma^{2'}, G')]]] \quad (\text{C.9})$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]] \quad (\text{C.10})$$

$$= -\mathbb{E}_{f, \sigma^2, G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | f, \sigma^2, G} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]] \quad (\text{C.11})$$

$$= -\mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]] \quad (\text{C.12})$$

Finally, since the query posterior entropy $H(Y | \mathbf{x}^{1:t-1})$ does not depend on the candidate experiment a_t , we obtain

$$\begin{aligned} & \arg \max_{a_t} I(Y; \mathbf{X}^t | \mathbf{x}^{1:t-1}) \\ &= \arg \max_{a_t} H(Y | \mathbf{x}^{1:t-1}) + H(\mathbf{X}^t | \mathbf{x}^{1:t-1}) - H(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}) \\ &= \arg \max_{a_t} H(\mathbf{X}^t | \mathbf{x}^{1:t-1}) - H(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}) \end{aligned} \quad (\text{C.13})$$

which, together with Eqs. (C.5) and (C.8), completes the proof. \square

C.2 Derivation of the Causal Discovery Utility Function

To derive $U_{\text{cd}}(a)$, we note that $Y = q_{\text{cd}}(\mathcal{M}) = G$, and hence the joint entropy of experiment outcome and query in Eq. (C.3) becomes

$$H(G, \mathbf{X}^t | \mathbf{x}^{1:t-1}) = -\mathbb{E}_{G, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(G, \mathbf{X}^t | \mathbf{x}^{1:t-1})] \quad (\text{C.14})$$

$$= -\mathbb{E}_{G, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | G, \mathbf{x}^{1:t-1}) + \log p(G | \mathbf{x}^{1:t-1})] \quad (\text{C.15})$$

$$= -\mathbb{E}_{G, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | G, \mathbf{x}^{1:t-1})] + H(G | \mathbf{x}^{1:t-1}) \quad (\text{C.16})$$

$$= -\mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | G, \mathbf{x}^{1:t-1})]] + H(G | \mathbf{x}^{1:t-1}). \quad (\text{C.17})$$

Substituting this into Eq. (C.2) yields

$$I(G; \mathbf{X}^t | \mathbf{x}^{1:t-1}) \tag{C.18}$$

$$= H(\mathbf{X}^t | \mathbf{x}^{1:t-1}) + \mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | G, \mathbf{x}^{1:t-1})]] . \tag{C.19}$$

By Eq. (C.12), we have

$$= \mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | G, \mathbf{x}^{1:t-1}) - \log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]] \tag{C.20}$$

which recovers the utility function $U_{\text{CD}}(a)$ from Eq. (4.13).

C.3 Derivation of the Causal Model Learning Utility Function

To derive $U_{\text{CML}}(a)$ given $Y = q_{\text{CML}}(\mathcal{M}) = \mathcal{M}$, the joint entropy of experiment outcome and query in Eq. (C.3) are given by

$$H(\mathcal{M}, \mathbf{X}^t | \mathbf{x}^{1:t-1}) = -\mathbb{E}_{\mathcal{M}, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(\mathcal{M}, \mathbf{X}^t | \mathbf{x}^{1:t-1})] \tag{C.21}$$

$$= -\mathbb{E}_{\mathcal{M}, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | \mathcal{M}, \mathbf{x}^{1:t-1}) + \log p(\mathcal{M} | \mathbf{x}^{1:t-1})] \tag{C.22}$$

$$= -\mathbb{E}_{\mathcal{M}, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(\mathbf{X}^t | \mathcal{M})] + H(\mathcal{M} | \mathbf{x}^{1:t-1}) \tag{C.23}$$

$$= -\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log p(\mathbf{X}^t | \mathcal{M})]] + H(\mathcal{M} | \mathbf{x}^{1:t-1}). \tag{C.24}$$

As previously, substituting this into Eq. (C.2) yields

$$I(G; \mathbf{X}^t | \mathbf{x}^{1:t-1}) = H(\mathbf{X}^t | \mathbf{x}^{1:t-1}) + \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log p(\mathbf{X}^t | \mathcal{M})]] \tag{C.25}$$

and by Eq. (C.10), we have

$$= \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log p(\mathbf{X}^t | \mathcal{M}) - \log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]] \tag{C.26}$$

which recovers the utility $U_{\text{CML}}(a)$ from Eq. (4.14).

For our concrete modeling choices we can further simplify this utility. Let $\mathbf{Anc}_i^{\mathcal{M}}$ and $\mathbf{Pa}_i^{\mathcal{M}}$ denote the ancestor and parent sets of node X_i in \mathcal{M} . Then,

$$\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log p(\mathbf{X}^t | \mathcal{M})]] \tag{C.27}$$

$$= \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} \left[\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} \left[\log \prod_{i \notin \mathcal{I}^t} p^{\text{do}(a_t)}(\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \mathcal{M}) \right] \right] \tag{C.28}$$

$$= \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} \left[\mathbb{E}_{\mathbf{X}^t | \mathcal{M}} \left[\sum_{i \notin \mathcal{I}^t} \log p^{\text{do}(a_t)}(\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \mathcal{M}) \right] \right] \tag{C.29}$$

$$= \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} \left[\sum_{i \notin \mathcal{I}^t} \mathbb{E}_{\mathbf{X}^t | \mathcal{M}} [\log p^{\text{do}(a_t)}(\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \mathcal{M})] \right] \tag{C.30}$$

$$= \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} \left[\sum_{i \notin \mathcal{I}^t} \mathbb{E}_{\mathbf{Anc}_i^{\mathcal{M}} | \text{do}(a_t), \mathcal{M}} \left[\mathbb{E}_{\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \text{do}(a_t), \mathcal{M}} [\log p^{\text{do}(a_t)}(\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \mathcal{M})] \right] \right] . \tag{C.31}$$

Since our root nodes and GPs assume an additive Gaussian noise model, the innermost expectation amounts to the negative entropy the Gaussian noise variable, i.e.,

$$\mathbb{E}_{\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \text{do}(a_t), \mathcal{M}} [\log p^{\text{do}(a_t)}(\mathbf{X}_i^t | \mathbf{pa}_i^{\mathcal{M}}, \mathcal{M})] = -\frac{N_t}{2} \log(2\pi\sigma_i^2 e). \tag{C.32}$$

As we further assume a homoscedastic noise model for our GPs, Eq. (C.31) reduces to

$$\mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t-1}} \left[\sum_{i \notin \mathcal{I}^t} -\frac{N_t}{2} \log(2\pi\sigma_i^2 e) \right] \quad (\text{C.33})$$

$$= -\mathbb{E}_{\mathbf{f}, \sigma^2, G | \mathbf{x}^{1:t-1}} \left[\sum_{i \notin \mathcal{I}^t} \frac{N_t}{2} \log(2\pi\sigma_i^2 e) \right] \quad (\text{C.34})$$

$$= -\mathbb{E}_G | \mathbf{x}^{1:t-1} \left[\mathbb{E}_{\sigma^2 | G, \mathbf{x}^{1:t-1}} \left[\sum_{i \notin \mathcal{I}^t} \frac{N_t}{2} \log(2\pi\sigma_i^2 e) \right] \right] \quad (\text{C.35})$$

$$= -\mathbb{E}_G | \mathbf{x}^{1:t-1} \left[\sum_{i \notin \mathcal{I}^t} \mathbb{E}_{\sigma_i^2 | G, \mathbf{x}^{1:t-1}} \left[\frac{N_t}{2} \log(2\pi\sigma_i^2 e) \right] \right], \quad (\text{C.36})$$

which can be approximated by nested Monte Carlo estimation. For non-root nodes we approximate the inner expectation with a single point estimate (cf. Section 4.1). For root nodes we can compute the inner expectation in closed form as

$$\mathbb{E}_{\sigma_i^2 | G, \mathbf{x}^{1:t-1}} \left[\frac{N_t}{2} \log(2\pi\sigma_i^2 e) \right] = \frac{N_t}{2} (\log(2\pi e) - \psi(\alpha_i^t) + \log \beta_i^t) \quad (\text{C.37})$$

where α_i^t, β_i^t are the parameters of the inverse-gamma noise posterior $\sigma_i^2 \sim \Gamma^{-1}(\sigma_i^2 | \alpha_i^t, \beta_i^t)$ (see Appx. D.3) and $\psi(\cdot)$ is the digamma function.

Proof (adapted from [74]). We need to show that

$$\mathbb{E}_{\sigma^2} [\log(\sigma^2)] = -\psi(\alpha) + \log \beta \quad (\text{C.38})$$

where the noise variance σ^2 follows an inverse-gamma density

$$\sigma^2 \sim \Gamma^{-1}(\sigma^2 | \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot (\sigma^2)^{-\alpha-1} \cdot e^{-\frac{\beta}{\sigma^2}}. \quad (\text{C.39})$$

By substituting $y = \log \sigma^2$ we get

$$y \sim p(y | \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \cdot e^{-\alpha y} \cdot e^{-\beta e^{-y}}. \quad (\text{C.40})$$

Now note that

$$\int_{-\infty}^{\infty} p(y | \alpha, \beta) dy = 1 \quad (\text{C.41})$$

and hence

$$\frac{\Gamma(\alpha)}{\beta^\alpha} = \int_{-\infty}^{\infty} e^{-\alpha y} \cdot e^{-\beta e^{-y}} dy. \quad (\text{C.42})$$

By differentiating the latter integrand w.r.t. α we get

$$\frac{d}{d\alpha} \left(e^{-\alpha y} \cdot e^{-\beta e^{-y}} \right) = (-y) e^{-\alpha y} \cdot e^{-\beta e^{-y}} = (-y) \cdot p(y | \alpha, \beta) \cdot \frac{\Gamma(\alpha)}{\beta^\alpha}. \quad (\text{C.43})$$

Bringing the parts together we obtain

$$\mathbb{E}_{\sigma^2} [\log(\sigma^2)] = \mathbb{E}_y [y] \quad (\text{C.44})$$

$$= \int_{-\infty}^{\infty} y \cdot p(y | \alpha, \beta) dy \quad (\text{C.45})$$

$$\stackrel{\text{Eq. (C.43)}}{=} -\frac{\beta^\alpha}{\Gamma(\alpha)} \int_{-\infty}^{\infty} \frac{d}{d\alpha} \left(e^{-\alpha y} \cdot e^{-\beta e^{-y}} \right) dy \quad (\text{C.46})$$

$$= -\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{d}{d\alpha} \left(\int_{-\infty}^{\infty} e^{-\alpha y} \cdot e^{-\beta e^{-y}} dy \right) \quad (\text{C.47})$$

$$\stackrel{\text{Eq. (C.42)}}{=} -\frac{\beta^\alpha}{\Gamma(\alpha)} \frac{d}{d\alpha} \left(\frac{\Gamma(\alpha)}{\beta^\alpha} \right) \quad (\text{C.48})$$

$$= -\frac{\beta^\alpha}{\Gamma(\alpha)} \left(\beta^{-\alpha} \cdot \frac{d}{d\alpha} \Gamma(\alpha) - \Gamma(\alpha) \cdot \beta^{-\alpha} \cdot \log \beta \right) \quad (\text{C.49})$$

$$= -\psi(\alpha) + \log \beta, \quad (\text{C.50})$$

which completes the proof. \square

In summary, in our instance of GP-DIBS-ABCI we estimate the causal model learning utility as

$$U_{\text{CML}}(a_t) = -\mathbb{E}_{G | \mathbf{x}^{1:t-1}} \left[\sum_{i \in \mathbf{R}(G) \setminus \mathcal{I}^t} \frac{N_t}{2} (\log(2\pi e) - \psi(\alpha_i^t) + \log \beta_i^t) + \sum_{i \in \mathbf{NR}(G) \setminus \mathcal{I}^t} \mathbb{E}_{\sigma_i^2 | G, \mathbf{x}^{1:t-1}} \left[\frac{N_t}{2} \log(2\pi \sigma_i^2 e) \right] + \mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]] \right] \quad (\text{C.51})$$

C.4 Derivation of the Causal Reasoning Utility Function

We derive the utility function $U_{\text{CR}}(a)$ in Eq. (4.12) for the query $Y = X_j^{\text{do}(X_i=\psi)}$ with $\psi \sim p(\psi)$ a distribution over intervention values. Starting with the joint entropy in Eq. (C.3) we marginalise over graphs (instead of SCMs) to exploit that we can sample from and evaluate $p(\mathbf{X} | G, \mathbf{x}^{1:t-1})$ in closed form by using GPs:

$$\begin{aligned} -H(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}) &= \mathbb{E}_{Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}} [\log p(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1})] \quad (\text{C.52}) \\ &= \mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{Y, \mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(Y, \mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]] \quad (\text{C.53}) \end{aligned}$$

$$= \mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\mathbb{E}_Y | \mathbf{X}^t, G, \mathbf{x}^{1:t-1} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p(Y | \mathbf{X}^t, G', \mathbf{x}^{1:t-1}) \cdot p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]]]] \quad (\text{C.54})$$

To estimate $\mathbb{E}_Y | \mathbf{X}^t, G, \mathbf{x}^{1:t-1} [\cdot]$ we first sample intervention values $\psi \sim p(\psi)$ and then sample from the respective interventional densities $p^{\text{do}(X_i=\psi)}(X_j | \mathbf{X}^t, G, \mathbf{x}^{1:t-1})$ induced by candidate SCMs with graph G . Thus, the expectation becomes $\mathbb{E}_\psi [\mathbb{E}_{X_j | \mathbf{X}^t, G, \mathbf{x}^{1:t-1}}^{\text{do}(X_i=\psi)} [\cdot]]$. To evaluate $p(Y | \mathbf{X}^t, G', \mathbf{x}^{1:t-1})$ we estimate $p^{\text{do}(X_i=\psi)}(X_j | \mathbf{X}^t, G', \mathbf{x}^{1:t-1})$ as described in Appx. D.1. The joint entropy therefore becomes

$$\begin{aligned} -H(Y, \mathbf{X}^t | \mathbf{x}^{1:t-1}) &= \mathbb{E}_{G | \mathbf{x}^{1:t-1}} [\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t-1}} [\mathbb{E}_\psi [\mathbb{E}_{X_j | \mathbf{X}^t, G, \mathbf{x}^{1:t-1}}^{\text{do}(X_i=\psi)} [\log \mathbb{E}_{G' | \mathbf{x}^{1:t-1}} [p^{\text{do}(X_i=\psi)}(X_j | \mathbf{X}^t, G', \mathbf{x}^{1:t-1}) \cdot p(\mathbf{X}^t | G', \mathbf{x}^{1:t-1})]]]]]] \quad (\text{C.55}) \end{aligned}$$

By substituting Eqs. (C.13) and (D.1) into Eq. (C.12) we obtain the causal reasoning utility in Eq. (4.12).

D Approximate Inference and Experimental Details

In this section, we provide details about our approximate inference and estimation procedures, including the estimation of the marginal interventional likelihoods in Section D.1 and prior choices in Sections D.2 — D.4. We also provide details on DiBS for approximate graph posterior inference in Section D.5, the estimation of the information gain utilities in Section D.6, and our use of Bayesian Optimisation for experimental design in Section D.7.

D.1 Estimating Posterior Marginal Interventional Likelihoods

In the following, we show how we estimate (posterior) marginal interventional likelihoods $p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{x}^{1:t})$. Let \mathbf{Anc}_i^G and \mathbf{Pa}_i^G denote the ancestor and parent sets of node X_i in G . Then, the marginal interventional likelihood is given by

$$\begin{aligned} & p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{x}^{1:t}) \\ &= \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t}} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathcal{M}) \right] \end{aligned} \quad (\text{D.1})$$

$$= \mathbb{E}_{\mathbf{f}, \boldsymbol{\sigma}^2, G | \mathbf{x}^{1:t}} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{f}, \boldsymbol{\sigma}^2, G) \right] \quad (\text{D.2})$$

$$= \mathbb{E}_{\mathbf{f}, \boldsymbol{\sigma}^2, G | \mathbf{x}^{1:t}} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), \mathbf{f}, \boldsymbol{\sigma}^2, G} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{anc}_i^G, \mathbf{f}, \boldsymbol{\sigma}^2, G) \right] \right]. \quad (\text{D.3})$$

Given that X_i is independent of its non-descendants given its parents, we obtain

$$= \mathbb{E}_{\mathbf{f}, \boldsymbol{\sigma}^2, G | \mathbf{x}^{1:t}} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), \mathbf{f}, \boldsymbol{\sigma}^2, G} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{pa}_i^G, f_i, \sigma_i^2, G) \right] \right] \quad (\text{D.4})$$

$$= \mathbb{E}_G | \mathbf{x}^{1:t} \left[\mathbb{E}_{\mathbf{f}, \boldsymbol{\sigma}^2 | G, \mathbf{x}^{1:t}} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), \mathbf{f}, \boldsymbol{\sigma}^2, G} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{pa}_i^G, f_i, \sigma_i^2, G) \right] \right] \right]. \quad (\text{D.5})$$

Given that $p(\mathbf{f}, \boldsymbol{\sigma}^2 | G, \mathbf{x}^{1:t})$ factorises and \mathbf{Anc}_i^G are independent of mechanisms and noise variances $\mathbf{f}, \boldsymbol{\sigma}^2$ of the non-ancestors of X_i , we have

$$\begin{aligned} &= \mathbb{E}_G | \mathbf{x}^{1:t} \left[\mathbb{E}_{\mathbf{f}_{\mathbf{Anc}_i^G}, \boldsymbol{\sigma}_{\mathbf{Anc}_i^G}^2 | G, \mathbf{x}^{1:t}} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), \mathbf{f}_{\mathbf{Anc}_i^G}, \boldsymbol{\sigma}_{\mathbf{Anc}_i^G}^2, G} \left[\right. \right. \right. \\ & \quad \left. \left. \left. \mathbb{E}_{f_i, \sigma_i^2 | G, \mathbf{x}^{1:t}} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{pa}_i^G, f_i, \sigma_i^2, G) \right] \right] \right] \right]. \end{aligned} \quad (\text{D.6})$$

Finally, marginalising out the functions and noise variances, we obtain

$$= \mathbb{E}_G | \mathbf{x}^{1:t} \left[\mathbb{E}_{\mathbf{f}_{\mathbf{Anc}_i^G}, \boldsymbol{\sigma}_{\mathbf{Anc}_i^G}^2 | G, \mathbf{x}^{1:t}} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), \mathbf{f}_{\mathbf{Anc}_i^G}, \boldsymbol{\sigma}_{\mathbf{Anc}_i^G}^2, G} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{pa}_i^G, G) \right] \right] \right] \right] \quad (\text{D.7})$$

$$= \mathbb{E}_G | \mathbf{x}^{1:t} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), G} \left[p^{\text{do}(x_j)}(\mathbf{x}_i | \mathbf{pa}_i^G, G) \right] \right] \quad (\text{D.8})$$

$$= \mathbb{E}_G | \mathbf{x}^{1:t} \left[\mathbb{E}_{\mathbf{Anc}_i^G | \text{do}(x_j), G} \left[p(\mathbf{x}_i | \mathbf{pa}_i^G, G) \Big|_{X_j=x_j} \right] \right]. \quad (\text{D.9})$$

We use Monte Carlo estimation to approximate the outer expectation of this quantity according to Eq. (4.10). To approximate the inner expectation by performing ancestral sampling from the interventional density $p^{\text{do}(x_j)}(\mathbf{X} | G)$, where we use 50 samples when estimating the U_{CR} utility in Equation Eq. (4.12) and 200 samples when estimating the metrics described in Appx. F.

D.2 Sampling Ground Truth Graphs

When generating ground truth SCMs for evaluation, we sample causal graphs according to two random graph models. First, we sample scale-free graphs using the preferential attachment process presented by Barabási and Albert [6]. We use the `networkx.generators.barabasi_albert_graph` implementation provided in the NetworkX [29] Python package and interpret the returned, undirected graph as a DAG by only considering the upper-triangular part of its adjacency matrix. Before permuting the node labels, we generate graphs with in-degree 2 for nodes $\{X_i\}_{i=3}^d$ whereas X_1 and X_2 are always

root nodes. In addition, we consider Erdős-Renyi random graphs [20], where edges are sampled independently with probability $p = \frac{4}{d-1}$. After sampling edges, we choose a random ordering and discard any edges that disobey this ordering to obtain a DAG. Our choice of p yields an expected degree of 2. Unlike Lorch et al. [45], we do not provide our model with any kind of prior information on the graph structure.

D.3 Normal-Inverse-Gamma Prior for Root Nodes

We use a conjugate normal-inverse-gamma ($N\text{-}\Gamma^{-1}$) prior

$$p(f_i, \sigma_i^2 | G) = N\text{-}\Gamma^{-1}(\mu_i, \lambda_i, \alpha_i^R, \beta_i^R) \quad (\text{D.10})$$

as the joint prior over functions and noise parameters for root nodes in G (see Section 4 and Fig. 2). In our experiments, we use $\mu_i = 0$, $\lambda_i = 0.1$, $\alpha_i^R = 50$ and $\beta_i^R = 25$. When generating ground truth SCMs, we draw one sample for $(f_i^*, \sigma_i^{2,*})$ from this prior for all i and leave it fixed thereafter. Closed-form expressions for the (posterior) marginal likelihood can be found, e.g., in [51].

D.4 Gamma Priors for GP Hyperparameters of Non-Root Nodes

We model non-root node mechanisms with GPs (see Section 4.1), where each GP has a set of hyperparameters (κ_i, σ_i^2) where $\kappa_i = (\kappa_i^l, \kappa_i^o)$ includes a length scale and output scale parameter, respectively, and where σ_i^2 denotes the variance of the Gaussian noise variable U_i . In our experiments, we use $p(\sigma_i^2 | G) = \text{Gamma}(\alpha = 50, \beta = 500)$, $p(\kappa_i^o | G) = \text{Gamma}(\alpha = 100, \beta = 10)$ and $p(\kappa_i^l | G) = \text{Gamma}(\alpha = 30 \cdot |\mathbf{Pa}_i^G|, \beta = 30)$, where $|\mathbf{Pa}_i^G|$ denotes the size of the parent set of X_i in G .

D.5 DiBS for Approximate Posterior Graph Inference

DiBS [45] introduces a probabilistic latent space representation for DAGs to allow for efficient posterior inference in continuous space. Specifically, given some latent particle $\mathbf{z} \in \mathbb{R}^{d \times d \times 2}$ we can define an edge-wise generative model

$$p(G | \mathbf{z}) = \prod_{i=1}^d \prod_{\substack{j=1 \\ j \neq i}}^d p(G_{i,j} | \mathbf{z}) \quad (\text{D.11})$$

where $G_{i,j} \in \{0, 1\}$ indicates the absence/presence of an edge from X_i to X_j in G , and a prior distribution

$$p(\mathbf{Z}) \propto \exp(-\beta \mathbb{E}_{G | \mathbf{Z}} [h(G)]) \prod_{i,j,k} \mathcal{N}(z_{i,j,k} | 0, 1) \quad (\text{D.12})$$

where $h(G)$ is a scoring function quantifying the ‘‘degree of cyclicity’’ of G . β is a temperature parameter weighting the influence of the expected cyclicity in the prior. Lorch et al. [45] propose to use Stein Variational Gradient Descent [44] for approximate inference of $p(\mathbf{Z} | \mathbf{x}^{1:t})$. SVGD maintains a fixed set of particles $\mathbf{z} = \{\mathbf{z}_m\}_{m=1}^M$ and updates them using the posterior score $\nabla \log p(\mathbf{z} | \mathbf{x}^{1:t}) = \nabla \log p(\mathbf{z}) + \nabla \log p(\mathbf{x}^{1:t} | \mathbf{z})$. In our experiments, we use $K = 5$ latent particles. For the estimation of expectations as in Eq. (4.10), we use $K = 40$ MC graph samples unless otherwise stated, hence, a total of $M \cdot K = 200$ graphs, and we use the DiBS+ particle weighting. In contrast to the original DiBS version, we do not use the annealing parameter α to force the mass of $p(G | \mathbf{z})$ onto a single graph during training. For further details on the method and its implementation, we refer to the original publication [45] and the provided code.

D.6 Estimation of the Information Gain Utility Functions

When estimating the information gain utilities (see § 4.2 and Appx. C), we keep the set of Monte Carlo samples from the SCM posterior $p(\mathcal{M} | \mathbf{x}^{1:t})$ fixed for all evaluations of the chosen utility during a given experiment design phase at time t , i.e., during the optimisation for all candidate intervention sets and intervention targets. In our experiments, for the U_{CD} and U_{CML} utilities we sample 5 and 30 graphs to approximate the outer and inner expectations w.r.t. the posterior graphs, respectively.

We sample 100 hypothetical experiment outcomes with given batch size from $p(\mathbf{X}^t | G, \mathbf{x}^{1:t})$ to approximate the expectation $\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t}} [\cdot]$.

For the U_{CR} utility we sample 3 and 9 graphs to approximate the outer and inner expectations w.r.t. the posterior graphs, respectively. We sample 50 hypothetical experiment outcomes with given batch size from $p(\mathbf{X}^t | G, \mathbf{x}^{1:t})$ to approximate expectations of the form $\mathbb{E}_{\mathbf{X}^t | G, \mathbf{x}^{1:t}} [\cdot]$. To approximate the expectations $\mathbb{E}_{\psi} [\mathbb{E}_{\mathbf{X}_j | \mathbf{X}^t, G, \mathcal{D}}^{\text{do}(X_i=\psi)} [\cdot]]$ we sample 5 intervention values from $p(\psi)$ and draw 3 samples from $p^{\text{do}(X_i=\psi)}(X_j | \mathbf{X}^t, G, \mathcal{D})$ for each intervention value.

D.7 Bayesian Optimisation for Experimental Design

In order to find the optimal experiment $a_t^* = (\mathcal{I}^*, \mathbf{x}_{\mathcal{I}^*}^*)$ at time t , we compute the optimal intervention value $\mathbf{x}_{\mathcal{I}}^* \in \arg \max_{\mathbf{x}} U(\mathcal{I}, \mathbf{x})$ for each candidate intervention target set \mathcal{I} (see Eq. (4.15)). As the evaluation of our proposed utility functions $U(a)$ is expensive, we require an efficient approach for finding optimal intervention values using as few function evaluations as possible. Following von Kügelgen et al. [80], we employ *Bayesian optimisation* (BO) [46, 47] for this task and model our uncertainty in $U(\mathcal{I}, \mathbf{x})$ given previous evaluations $\mathcal{D}_{\text{BO}} = \{(\mathbf{x}_l, U(\mathcal{I}, \mathbf{x}_l))\}_{l=1}^k$ with a GP. We select a new candidate solution according to the GP-UCB acquisition function [76],

$$\mathbf{x}_{k+1} = \arg \max_{\mathbf{x}} \mu_k(\mathbf{x}) + \gamma \sigma_k(\mathbf{x}), \quad (\text{D.13})$$

where $\mu_k(\mathbf{x})$ and $\sigma_k(\mathbf{x})$ correspond to the mean and standard deviation of the GP predictive distribution $p(U(\mathcal{I}, \mathbf{x}) | \mathcal{D}_{\text{BO}})$ (see Appx. B). We then evaluate $U(\mathcal{I}, \mathbf{x}_{k+1})$ at the selected \mathbf{x}_{k+1} and repeat. The scalar factor γ trades off exploitation with exploration. In our experiments, we set $\gamma = 1$ and run the GP-UCB algorithm 8 times for each candidate set of intervention targets.

Algorithm 2: Particle Resampling

Input: set of latent particles $\mathbf{z} = \{z_k\}_{k=1}^K$
Output: set of resampled latent particles $\tilde{\mathbf{z}} = \{\tilde{z}_k\}_{k=1}^K$
 $\tilde{\mathbf{z}} \leftarrow \emptyset$ ▷ initialise set of resampled particles
 $N_{max} \leftarrow \lceil \frac{K}{4} \rceil$ ▷ max. number of particles to keep
 $\{w_k\}_{k=1}^K \leftarrow \left\{ \frac{p(z_k | \mathbf{x}^{1:t}) \bar{p}(z_k)}{\sum_k p(z_k | \mathbf{x}^{1:t}) \bar{p}(z_k)} \right\}$ ▷ compute particle weights
 $n_{kept} \leftarrow 0$
for w_k **in** $\text{sort_descending}(\{w_k\}_{k=1}^K)$ **do**
 if $n_{kept} < N_{max}$ **and** $w_k > 0.01$ **then**
 $\tilde{\mathbf{z}} \leftarrow \tilde{\mathbf{z}} \cup \{z_k\}$ $n_{kept} \leftarrow n_{kept} + 1$
 end
 else
 $z_{new} \sim p(\mathbf{Z})$
 $\tilde{\mathbf{z}} \leftarrow \tilde{\mathbf{z}} \cup \{z_{new}\}$
 end
end

E Implementation Details

In this section, we give details about our implementation, including our particle resampling procedure in Section E.1, the sharing and caching of priors in Section E.2, a discussion of the computational complexity of our implementation in Section E.3, and finally some information on our code framework and computing resources in Section E.4. Our implementation is available at <https://www.github.com/chritoth/active-bayesian-causal-inference>.

E.1 Particle Resampling

As described in Alg. 1, we resample latent particles $\mathbf{z} = \{z_k\}_{k=1}^K$ according to a predefined schedule instead of sampling new particles from the particle prior $p(\mathbf{Z})$ after each epoch. Although sampling new particles would allow for higher diversity in the graph Monte Carlo samples and their respective mechanisms, it also entails a higher computational burden as the caching of mechanism marginal log-likelihoods is not as effective anymore. On the other hand, keeping a subset of the inferred particles is efficient, because once we have inferred a “good” particle z_k that supposedly has a high posterior density $p(z_k | \mathbf{x}^{1:t})$ it would be wasteful to discard the particle only to infer a similar particle again. Empirically, we found that keeping particles depending on their unnormalized posterior densities according to Alg. 2 does not diminish inference quality while increasing computational efficiency. In our experiments, we chose the following resampling schedule:

$$r_t = \begin{cases} 1 & \text{if } t \in \{1, 2, 3, 4, 5, 6, 9\} \\ 1 & \text{if } t \bmod 5 = 0 \\ 0 & \text{otherwise.} \end{cases}$$

E.2 Shared Priors and Caching of Marginal Likelihoods

We share priors for mechanisms and noise $p(f_i, \sigma_i | G)$, as well as for GP hyperparameters $p(\kappa_i | G)$, across all graphs G that induce the same parent set \mathbf{Pa}_i^G . Consequently, not only the posteriors $p(f_i, \sigma_i | G, \mathbf{x}^{1:t})$ and $p(\kappa_i | G, \mathbf{x}^{1:t})$, but also the GP marginal likelihoods $p(\mathbf{x}_i^{1:t} | G)$ and GP predictive marginal likelihoods $p(\mathbf{x}_i^{t+1} | G, \mathbf{x}^{1:t})$ can be shared across graphs with identical parent sets for node X_i . By caching the values of the computed GP (posterior) marginal likelihoods, we substantially save on computational cost when computing expectations of the form $\mathbb{E}_{G|\mathbf{z}} [p(\mathbf{x}^{1:t} | G) \phi(G)]$ and $\mathbb{E}_{G|\mathbf{z}} [p(\mathbf{x}^{t+1} | G, \mathbf{x}^{1:t}) \phi(G)]$ where $\phi(G)$ is some quantity depending the graph.

Specifically, consider that $p(\mathbf{x}^{1:t} | G) = \prod_i p(\mathbf{x}_i^{1:t} | G)$ factorizes into the GP marginal likelihoods of the individual mechanisms, so for d nodes in the graph and N samples in $\mathbf{x}^{1:t}$ (counted over all time steps) the complexity of computing $p(\mathbf{x}^{1:t} | G)$ is $O(d \cdot N^3)$ for a fixed set of GP hyperparameters (for

simplicity we ignore that not all d mechanisms are modelled by GPs as some are root nodes, so this is not a tight bound). Thus, in the worst case, estimating $p(\mathbf{x}^{1:t}|z) = \mathbb{E}_{G|z}[p(\mathbf{x}^{1:t}|G)]$ with K graph samples would yield a complexity of $O(K \cdot d \cdot N^3)$. By caching the marginal likelihoods as outlined above we can rewrite the complexity $O(K \cdot d \cdot N^3)$ as $O(L \cdot N^3)$ where $L \leq K \cdot d$ denotes the number of unique mechanisms entailed by the set of K graph samples. Although this does not reduce the worst case complexity it nevertheless greatly alleviates the computational demand in practice.

The benefit of caching becomes even more pronounced as $p(G|z)$ concentrates its mass on a small set of similar graphs as a result of the inference process. In particular, when updating the latent particles using SVGD we do not need to recompute $p(\mathbf{x}^{1:t}|G)$ after we have once before sampled G , which greatly speeds up the gradient estimation of the particle posterior.

E.3 Computational Complexity

There are two main phases in our algorithm (disregarding the computation of metrics for evaluation), (i) the inference phase where we (approximately) infer the posterior over SCMs $p(\mathcal{M}|\mathbf{x}^{1:t})$ after collecting new experimental data, and (ii) the experimental design phase.

The inference phase has worst-case complexity in $O(T_{SVGD} \cdot (T_{HP} \cdot M \cdot K \cdot d \cdot N^3 + M^2 \cdot d^2))$ where T_{SVGD} is the number of SVGD update steps, T_{HP} is the number of GP hyperparameter update steps, M is the number of latent z particles, K is the number of graph samples per latent z particle, d is the number of nodes in the network, and N is the number of collected experimental samples in $\mathbf{x}^{1:t}$. The computation of the GP marginal likelihood dominates the complexity of the inference phase. To improve scalability we make use of shared priors and caching. Additionally, we update the GP hyperparameters according to a predefined schedule instead of doing so after each performed experiment. In our experiments, both measures reduce the factor $T_{HP} \cdot M \cdot K \cdot d$ significantly. For example, running inference with 5 freshly initialized z particles with 40 graph samples each on a scale-free SCM with 20 nodes updates the hyperparameters of (2970, 964, 177) GPs during SVGD update steps (1, 5, 10), and of less than 15 GPs after 20 SVGD update steps. Compared to $M \cdot K \cdot d = 4000$ in this example, the benefit is evident.

In the experimental design phase we parallelize finding the optimal intervention value for each candidate target node, so the complexity is basically the number of Bayesian optimization (BO) steps times the complexity of the utility we want to optimize for. For a general query (cf. Eq. (4.11)) we have complexity in $O(T_{BO} \cdot M \cdot K_{outer} \cdot S \cdot Q \cdot M \cdot K_{inner} \cdot (O(p(y|\mathcal{M}) + d \cdot N^3)))$ where T_{BO} is the number of Bayesian optimization iterations, M is the number of latent z particles, K_{outer} is the number of graph samples in the outer SCM expectation in, S is the number of simulated experiments per SCM, Q is the number of simulated queries, K_{inner} is the number of graph samples in the inner SCM expectation, $O(p(y|\mathcal{M}))$ is the complexity of evaluating the query likelihood and $d \cdot N^3$ is the complexity of evaluating the GP predictive posteriors for the simulated experiments. For the causal discovery and model learning utilities the complexity reduces to $O(T_{BO} \cdot M \cdot K_{outer} \cdot S \cdot M \cdot K_{inner} \cdot d \cdot N^3)$.

In summary, the complexity of our ABCI implementation is dominated the experimental design phase from a high-level perspective. On a lower level, the cubic scaling of GP inference is the major computational issue that we alleviate by caching the (posterior) marginal log-likelihoods (see Appx. E.2 for details). However, in a small data regime where experimental data is costly to obtain, GPs are not a prohibitive element in our inference chain. Furthermore, GP scaling issues could be alleviated, e.g., by using sparse GP approximation or any other kind of scalable Bayesian mechanism model. Disregarding issues of GP scaling, the estimation of the information gain utilities is still costly, simply because it requires many levels of nested sampling and too few Monte-Carlo samples will yield too noisy, in the worst case unusable utility estimates. We believe that in follow-up work much can be gained in terms of scalability as well as performance by incorporating recent advances in nested Monte-Carlo/information gain estimation techniques(e.g., [8, 28, 63]).

Finally, consider that a single estimation of the causal discovery utility for an SCM with 20 nodes with $N = 500$ previously collected experimental samples takes approximately 2 minutes on an off-the-shelf laptop. Thus, for 10 BO iterations we can do the experimental design phase in 20 minutes (assuming we parallelize the utility optimization for each node). In a practical application scenario one might be very willing to invest hours or days for the design phase before conducting a costly experiment.

E.4 Implementation and Computing Resources

Our Python implementation uses the PyTorch [54], GPyTorch [24], CDT [41], SKLearn [58], NetworkX [29] and BoTorch [5] packages, which greatly eased our implementation efforts. All of our experiments were run on CPUs. We parallelise the experiment design by running the optimisation process for each candidate intervention set on a separate core.

F Evaluation Metrics

In this section, we provide details on the metrics used to evaluate our method in Section 5 and Appx. G. In our experiments, we use (nested) Monte Carlo estimators to approximate intractable expectations.

Kullback-Leibler Divergence. We evaluate the inferred posterior over queries given observed data, $p(Y | \mathbf{x}^{1:t})$, to the true query distribution $p(Y | \mathcal{M}^*)$ using the Kullback-Leibler Divergence (KLD), i.e.,

$$\text{KL}(p(Y | \mathcal{M}^*) || p(Y | \mathbf{x}^{1:t})) = \mathbb{E}_{Y | \mathcal{M}^*} [\log p(Y | \mathcal{M}^*) - \log p(Y | \mathbf{x}^{1:t})] \quad (\text{F.1})$$

$$= \mathbb{E}_{Y | \mathcal{M}^*} [\log p(Y | \mathcal{M}^*) - \log \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t}} [p(Y | \mathcal{M})]]. \quad (\text{F.2})$$

Query KLD. For $Y = X_5^{\text{do}(X_3=\psi)}$ with $\psi \sim p(\psi)$ we have

$$\text{Query KLD} = \mathbb{E}_{\psi} \left[\text{KL}(p^{\text{do}(X_3=\psi)}(X_5 | \mathcal{M}^*) || p^{\text{do}(X_3=\psi)}(X_5 | \mathbf{x}^{1:t})) \right] \quad (\text{F.3})$$

$$= \mathbb{E}_{\psi} \left[\mathbb{E}_{X_5 | \text{do}(X_3=\psi), \mathcal{M}^*} \left[\log p^{\text{do}(X_3=\psi)}(X_5 | \mathcal{M}^*) - \log p^{\text{do}(X_3=\psi)}(X_5 | \mathbf{x}^{1:t}) \right] \right]. \quad (\text{F.4})$$

To approximate the outer two expectations, we keep a fixed set of samples for each ground truth SCM to enhance comparability between different ABCI runs. For $p^{\text{do}(X_3=\psi)}(X_5 | \mathbf{x}^{1:t})$, we use the estimator described in Section D.1.

SCM KLD. For $Y = q_{\text{CML}}(\mathcal{M}) = \mathcal{M}$, we have

$$\text{SCM KLD} = \text{KL}(p(\mathcal{M} | \mathcal{M}^*) || p(\mathcal{M} | \mathbf{x}^{1:t})) \quad (\text{F.5})$$

$$= \mathbb{E}_{\mathcal{M} | \mathcal{M}^*} [\log p(\mathcal{M} | \mathcal{M}^*) - \log p(\mathcal{M} | \mathbf{x}^{1:t})] \quad (\text{F.6})$$

$$= 0 - \log p(\mathcal{M}^* | \mathbf{x}^{1:t}) \quad (\text{F.7})$$

$$= -\log \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t}} [p(\mathcal{M}^* | \mathcal{M})] \quad (\text{F.8})$$

$$= -\log \mathbb{E}_{G, \mathbf{f}, \sigma^2 | \mathbf{x}^{1:t}} [p(G^*, \mathbf{f}^*, \sigma^{2,*} | G, \mathbf{f}, \sigma^2)] \quad (\text{F.9})$$

$$= -\log \mathbb{E}_{G, \mathbf{f}, \sigma^2 | \mathbf{x}^{1:t}} [p(G^* | G) p(\mathbf{f}^* | \mathbf{f}) p(\sigma^{2,*} | \sigma^2)]. \quad (\text{F.10})$$

Now note that $p(G^* | G) = 1$ if $G = G^*$ and 0 otherwise, $p(\mathbf{f}^* | \mathbf{f}) = \delta(\mathbf{f}^* - \mathbf{f})$, and $p(\sigma^{2,*} | \sigma^2) = \delta(\sigma^{2,*} - \sigma^2)$. Hence, the SCM KLD vanishes iff the SCM posterior $p(\mathcal{M} | \mathbf{x}^{1:t})$ collapses onto the true SCM \mathcal{M}^* , and is infinite otherwise.

Average Interventional KLD. Computing the KLD for $Y = q_{\text{CML}}(\mathcal{M}) = \mathcal{M}$ is not useful for evaluation, since it vanishes when the SCM posterior $p(\mathcal{M} | \mathbf{x}^{1:t})$ collapses onto the true SCM \mathcal{M}^* and is infinite otherwise. For this reason, we report the *average interventional KLD* as a proxy metric, which we define as

$$\text{Avg. I-KLD} = \frac{1}{d} \sum_{i=1}^d \mathbb{E}_{\psi} \left[\text{KL}(p^{\text{do}(X_i=\psi)}(X | \mathcal{M}^*) || p^{\text{do}(X_i=\psi)}(X | \mathbf{x}^{1:t})) \right] \quad (\text{F.11})$$

$$= \frac{1}{d} \sum_{i=1}^d \mathbb{E}_{\psi} \left[\mathbb{E}_{X | \text{do}(X_i=\psi), \mathcal{M}^*} \left[\log p^{\text{do}(X_i=\psi)}(X | \mathcal{M}^*) - \log p^{\text{do}(X_i=\psi)}(X | \mathbf{x}^{1:t}) \right] \right] \quad (\text{F.12})$$

$$= \frac{1}{d} \sum_{i=1}^d \mathbb{E}_{\psi} \left[\mathbb{E}_{X | \text{do}(X_i=\psi), \mathcal{M}^*} \left[\log p^{\text{do}(X_i=\psi)}(X | \mathcal{M}^*) - \log \mathbb{E}_{\mathcal{M} | \mathbf{x}^{1:t}} [p^{\text{do}(X_i=\psi)}(X | \mathcal{M})] \right] \right]. \quad (\text{F.13})$$

As with the Query KLD, we keep a fixed set of MC samples per ground truth SCM to approximate the two outer expectations to enhance comparability between different ABCI runs.

Expected Structural Hamming Distance. The Structural Hamming Distance (SHD)

$$\text{SHD}(G, G^*) = |\{(i, j) \in G : (i, j) \notin G^*\}| + |\{(i, j) \in G^* : (i, j) \notin G\}| \quad (\text{F.14})$$

denotes the simple graph edit distance, i.e., it counts the number of edges (i, j) that are present in the prediction graph G and not present in the reference graph G^* and vice versa. We report the expected SHD w.r.t. our posterior over graphs as

$$\text{ESHD}(G, G^*) = \mathbb{E}_{G | \mathbf{x}^{1:t}} [\text{SHD}(G, G^*)] \quad (\text{F.15})$$

AUPRC. Following previous work [19, 21, 45, 78], we report the *area under the precision recall curve* (AUPRC) by casting graph learning as a binary edge prediction problem given our inferred posterior edge probabilities $p(G_{i,j} | \mathbf{x}^{1:t})$. Refer to e.g. Murphy [52] for further information on this quantity.

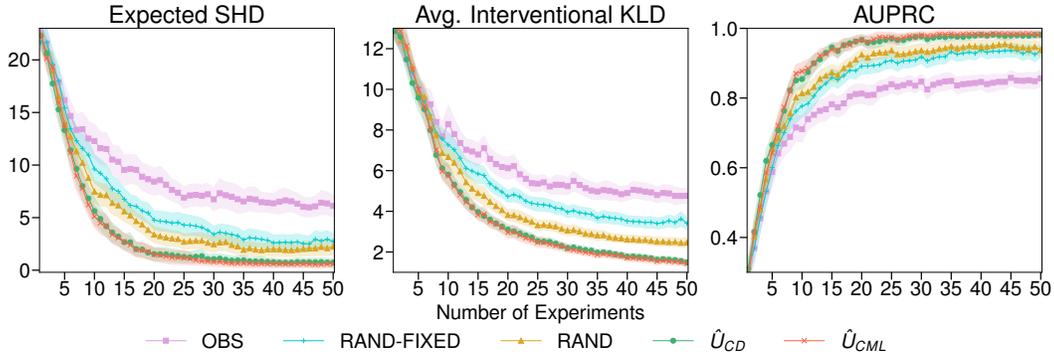


Figure 5: **Causal Discovery and SCM Learning on Scale-free Graphs with 10 Variables.** Comparison of the experimental design strategies with random and observational baselines on simulated ground truth models with 10 nodes. Lines and shaded areas show means and 95% confidence intervals (CIs) across 50 runs (10 randomly sampled ground-truth SCMs with 5 restarts per SCM). The U_{CD} and U_{CML} objectives perform on par with each other. Both clearly outperform the observational and random baselines on all metrics.

G Extended Experimental Results

Causal Discovery and SCM Learning for SCMs with $d = 10$ Variables. We report results on ground truth SCMs with $d = 10$ variables and scale-free graphs in Fig. 5. We initialise all methods with 5 observational samples and perform experiments with a batch size of 3. All other parameters are chosen as described in Appx. D.

Causal Discovery and SCM Learning for SCMs with $d = 20$ Variables. To demonstrate the scalability of our framework, we report results on ground truth SCMs with $d = 20$ variables and scale-free or Erdős-Renyi graphs in Fig. 6 and Fig. 7, respectively. We initialise all methods with 50 observational samples and perform experiments with a batch size of 5. All other parameters are chosen as described in Appx. D.

While ABCI shows clear benefits when scale-free causal graphs underlie the SCMs, we find that the advantage of ABCI diminishes on SCMs with unstructured Erdős-Renyi graphs, which appear to pose a harder graph identification problem. Moreover, we expect performance of our inference machinery, especially together with the informed action selection, to increase when investing more computational power to improve the quality of our estimates, e.g., by increasing the number of Monte Carlo samples used in our estimators and increasing the number of evaluations during the Bayesian optimisation phase.

Finally, in Fig. 8 we show that using a simple linear model (GP model with a linear kernel) is not able to reasonably capture the characteristics of the ground truth model (non-linear GP model) due to the model mismatch.

Learning Interventional Distributions vs. Causal Discovery and SCM Learning. We report additional metrics for our causal reasoning experiment as described in § 5 in Figs. 9 and 10. The key result here is that U_{CR} yields a significantly lower Query KLD while exhibiting a worse ESHD and Average I-KLD scores, which indicates that, indeed, the U_{CR} learns only those parts of the model that are relevant to reducing the uncertainty in our target query. This is more data efficient than trying to learn the entire model first and then answering the causal query of interest.

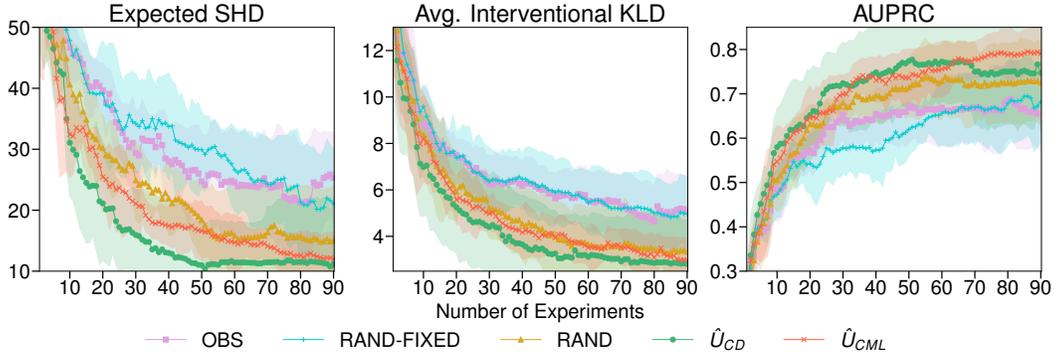


Figure 6: **Causal Discovery and SCM Learning on Scale-free Graphs with 20 Variables.** (Same figure as in Fig. 3 with additional confidence intervals for OBS and RAND FIXED.) Comparison of the experimental design strategy for causal discovery (U_{CD}) with random and observational baselines on simulated ground truth models with 20 nodes. Lines and shaded areas show means and 95% confidence intervals (CIs) across 15 runs (5 randomly sampled ground-truth SCMs with 3 restarts per SCM). The U_{CD} objective significantly outperforms the observational and random baselines on all metrics.

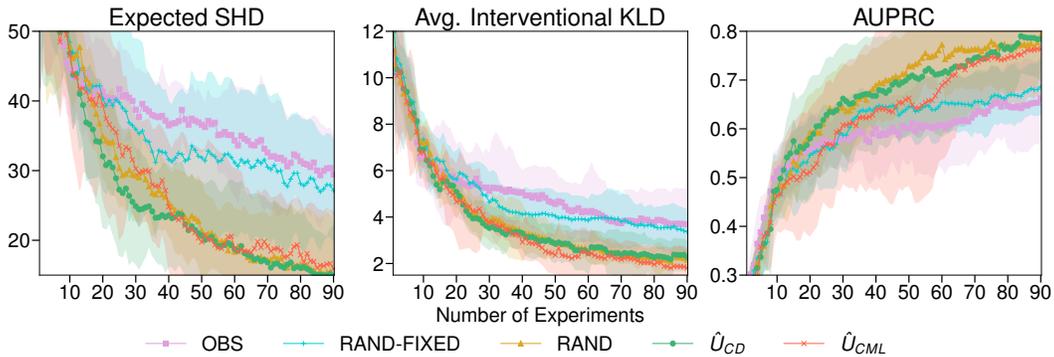


Figure 7: **Causal Discovery and SCM Learning on Erdős-Renyi Graphs with 20 Variables.** Comparison of experimental design strategies for causal discovery (U_{CD}) and causal model learning (U_{CML}) with random and observational baselines on simulated ground truth models with 20 nodes. Lines and shaded areas show means and 95% confidence intervals (CIs) across 15 runs (5 randomly sampled ground-truth SCMs with 3 restarts per SCM). The U_{CD} and U_{CML} strategies perform approx. equal to the strong random baseline (RAND) on all metrics, however, all three are significantly better than the weak random (RAND FIXED) and observational baselines. We expect that improving the quality of the U_{CD} and U_{CML} estimates (e.g., by scaling up computational resources invested in the MC estimates) yield similar benefits of the experimental design utilities as apparent in Fig. 6.

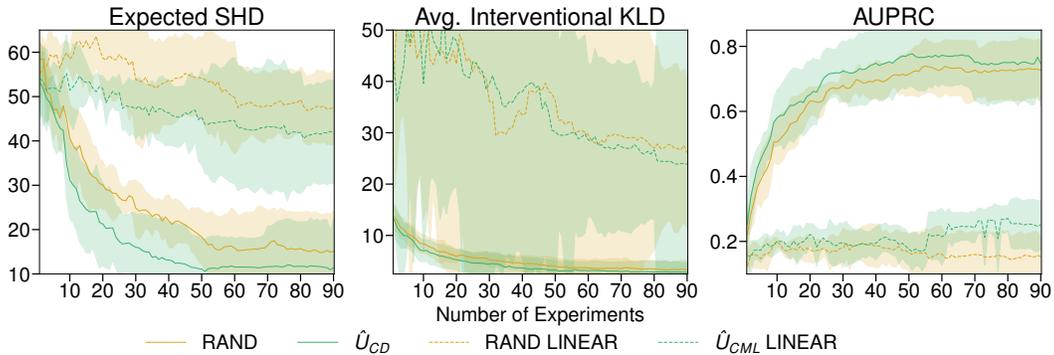


Figure 8: **Causal Discovery and SCM Learning on Scale-free Graphs with 20 Variables.** Comparison of non-linear GP model with a linear model (linear GP kernel) for U_{CD} and RAND on simulated ground truth models with 20 nodes. Lines and shaded areas show means and 95% confidence intervals (CIs) across 15 runs (5 randomly sampled ground-truth SCMs with 3 restarts per SCM). Clearly, the model mismatch in the linear model prohibits the identification of the ground-truth graph.

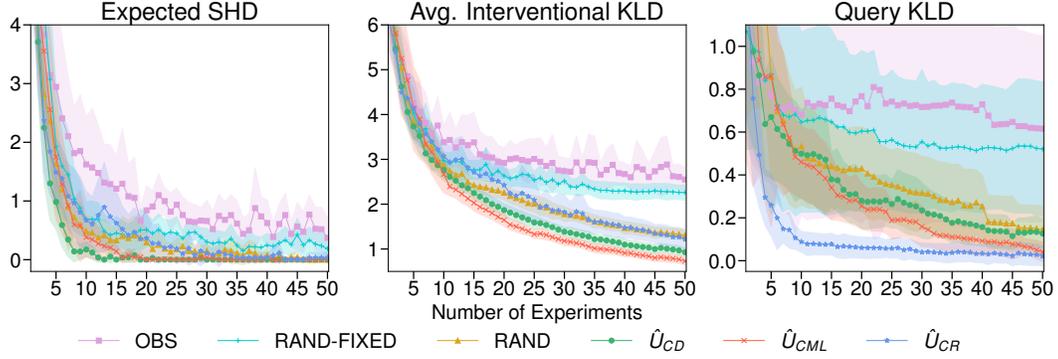


Figure 9: **Learning Interventional Distributions.** Comparison of the experimental design strategies with random and observational baselines. Lines and shaded areas show means and 95% confidence intervals (CIs) across 30 runs (10 randomly sampled ground-truth SCMs with 3 restarts per SCM). U_{CD} , U_{CML} and U_{CR} perform best w.r.t. the ESHD, Avg. I-KLD and Query KLD metrics respectively, which is expected.

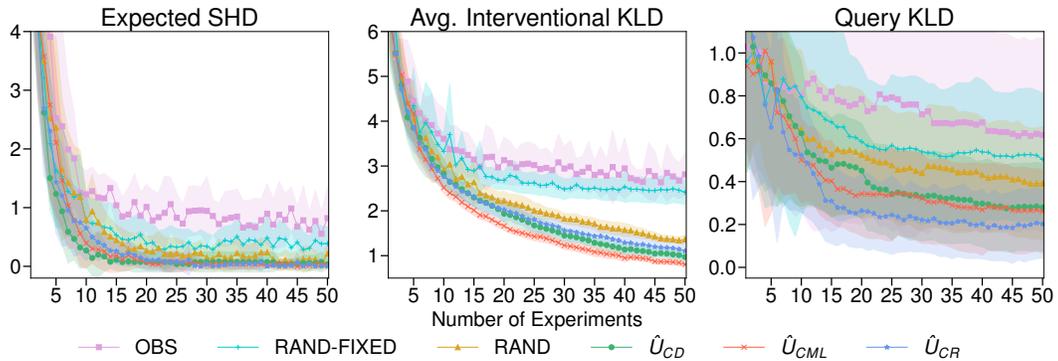


Figure 10: **Learning Interventional Distributions.** Comparison of the experimental design strategies with random and observational baselines. Lines and shaded areas show means and 95% confidence intervals (CIs) across 30 runs (10 randomly sampled ground-truth SCMs with 3 restarts per SCM). U_{CD} , U_{CML} and U_{CR} perform best w.r.t. the ESHD, Avg. I-KLD and Query KLD metrics respectively, which is expected.