MetaCaDI: A Meta-Learning Framework for Causal Discovery from Multiple Environments with Unknown Interventions

Hans Jarett Ong[1] **Yoichi Chikahara**[2] **Tomoharu Iwata**[2]

[1]Nara Institute of Science and Technology, Nara, Japan
[2]NTT Communication Science Laboratories, Kyoto, Japan,

Abstract

Uncovering the causal mechanisms of complex real-world systems remains a significant challenge, as these systems often entail high data collection costs and involve unknown interventions. We introduce MetaCaDI, the first framework to cast the identification of unknown interventions as a meta-learning problem, explicitly leveraging a jointly learned causal graph. MetaCaDI is a Bayesian framework that learns a shared causal structure across multiple environments and is optimized to rapidly adapt to new, few-shot intervention target identification tasks. A key innovation is our model's analytical adaptation, which uses a closed-form solution to bypass expensive and potentially unstable gradient-based bilevel optimization. Extensive experiments on synthetic and complex gene expression data demonstrate that MetaCaDI significantly outperforms state-of-the-art methods. It excels at identifying intervention targets from as few as 3 samples—where existing methods collapse to random chance—while robustly recovering the shared causal graph, proving its effectiveness in data-scarce scenarios.

1 INTRODUCTION

This paper addresses the problem of causal discovery from multiple environments, which has emerged as a promising avenue for unraveling complex real-world systems [Mooij et al., 2020, Li et al., 2023, Jalaldoust et al., 2025]. In this problem, we consider the setting where we have access to multiple datasets generated by a shared causal graph structure, but each dataset is produced from a different *environment*, where distinct variables are intervened upon. The goal of this problem is to jointly infer the shared causal graph structure and distinct, unknown intervention targets.

Although this problem is pervasive in real-world applications, achieving high performance remains challenging, as the data sample size is often limited in practice. To illustrate this, we present two motivating examples in industrial and medical applications:

Example 1 (Large computing system): Consider a large-scale cloud computing system whose components (e.g., physical servers and software applications) occasionally experience localized faults such as hardware malfunctions or software bugs [Varici et al., 2022]. We can represent such a system as a causal graph whose nodes represent the components and whose edges express the dependencies among them. Due to the complex interdependencies, a system engineer wants to identify the graph structure and the causes of the faults (i.e., the intervention targets) [Mariani et al., 2018]. The engineer has access to multiple datasets collected under different system conditions. However, such datasets are often scarce, as one cannot intentionally cause a system fault. Nevertheless, the engineer needs to diagnose failures **in real-time**, using only a few observations of the failure.

Example 2 (Pharmacological neuroscience): In neuroscience, it is common to model the human brain as a causal graph where nodes represent brain regions and edges express the effective connectivity among them [Friston et al., 2003, Valdes-Sosa et al., 2011]. By collecting functional magnetic resonance imaging (fMRI) datasets from patient cohorts administered different pharmacological treatments, neurologists seek to infer this graph structure shared across datasets, as well as the specific brain regions modulated by each drug, which can be regarded as an unknown intervention target [Rowe et al., 2010, Oliva et al., 2022]. However, patient datasets for highly specific drug combinations or novel treatments are inherently scarce, making large-scale data collection costly. Therefore, identifying the specific brain regions affected by a drug from only a handful of patient records would greatly aid in understanding drug mechanisms and advancing targeted clinical care.

Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence (UAI 2026), PMLR 337:5060–5080.

Motivated by these challenging yet important data-scarce scenarios, we propose **MetaCaDI** (*Meta*-Learning for *Ca*usal *D*iscovery with Unknown *I*nterventions), a meta-learning approach to causal discovery from multiple environments. MetaCaDI is a Bayesian framework for inferring the joint posterior distribution over the shared causal graph and the distinct intervention targets specific to each small dataset. Using the estimated posterior probabilities, we can quantify the inference uncertainty of **both** the shared graph structure and the distinct intervention targets, which is essential for knowledge discovery under data-scarce scenarios.

MetaCaDI has two notable strengths. First, unlike multi-task learning approaches [Hägele et al., 2023], MetaCaDI can address real-time applications like **Example 1**, as meta-learning does not require the full model retraining to incorporate a newly arrived dataset. Second, MetaCaDI is designed to achieve high meta-learning performance in few-shot settings like **Example 2** by overcoming the computational inefficiency of standard gradient-based approaches [Bertinetto et al., 2019, Finn et al., 2017]. To bypass the need for gradient-based updates in few-shot adaptation, we develop an interventional data likelihood model that admits a closed-form, analytical solution, allowing MetaCaDI to rapidly adapt to new datasets with only a handful of samples.

Our contributions are threefold:

- This work is the first to formalize the joint inference of a causal graph and unknown intervention targets as a meta-learning problem (Section 3). By framing the intervention target identification from each dataset as a distinct meta-learning *task*, our model extracts the *task-shared* knowledge of the causal graph structure and then leverages it to improve the performance of inferring *task-specific* intervention targets.
- To effectively tune the task-specific parameters for each dataset, we formulate a likelihood model that admits a closed-form, analytical solution (Section 4.2). By eliminating the need for computationally demanding gradient-based optimization for these parameters, this closed-form solver enables computationally efficient and stable adaptation to new datasets.
- Through extensive experiments on synthetic and gene expression simulation datasets, we demonstrate that MetaCaDI outperforms existing methods in recovering both the causal graph structure and the intervention targets, even in the challenging small-sample setting.

2 PRELIMINARIES

2.1 CAUSAL GRAPHS AND INTERVENTIONS

A causal graph describes the data-generating process over variables $\mathcal{X} = \{X_1, \dots, X_d\}$, defined by a **Structural Causal Model (SCM)** [Pearl, 2009]. An SCM defines the *structural equation* for each variable X_i ($i \in \{1, \dots, d\}$), which determines the variable values as the output of a deterministic function f_i:

$$X_i = f_i(\mathrm{PA}(X_i), E_i),$$

where $\mathrm{PA}(X_i) \subseteq \mathcal{X} \backslash X_i$ is a variable subset called the *parents* of X_i, and E_i is an exogenous variable (e.g., noise).

A **causal graph** $\mathcal{G}$ is a graph that contains a directed edge $X_j \to X_i$ if and only if $X_j \in \mathrm{PA}(X_i)$. We make two standard assumptions on the causal graph $\mathcal{G}$ and the SCM. First, we assume that causal graph $\mathcal{G}$ is a directed acyclic graph (DAG). Second, we assume *causal sufficiency*, the condition that $E_i \perp\!\!\!\perp E_j$ holds for any $i \neq j$, which requires that there are no unobserved common causes (i.e., confounders) between any pair of variables in $\mathcal{X}$. As with the literature in this field [Hägele et al., 2023, Brouillard et al., 2020], to effectively instantiate the likelihood, we make an additional but **non-mandatory** assumption that each structural equation is expressed as a specific class of functions; in this paper, we consider an **Additive Noise Model (ANM)** [Hoyer et al., 2008]:

$$X_i = f_i(\mathrm{PA}(X_i)) + E_i.$$

Theoretically, data generated under such restricted functional forms as ANMs exhibit distributional asymmetries between $X_i \to X_j$ and $X_j \to X_i$, enabling the unique identification of causal graphs [Peters et al., 2014]. In the absence of such functional assumptions, from observational data alone, the causal graph can only be identified up to its *Markov Equivalence Class (MEC)*—a set of causal graphs that encode the same conditional independence relations.

An **intervention** is a replacement operation for the structural equations. In this paper, we consider both **hard** and **soft interventions**: The former replaces the structural equation of variable X_i with constant $X_i = x_i$, whereas the latter modifies the structural equation while keeping its dependence on the values of parents $\mathrm{PA}(X_i)$ [Pearl, 2009].

Such structural equation modification for X_i changes the distribution of its descendants but not that of its ancestors, allowing for the disambiguation of causal directions that are otherwise indistinguishable. Without functional assumptions, an MEC is known to be narrowed down to an Interventional MEC (I-MEC) [Eberhardt and Scheines, 2007]. Combined with the distributional asymmetries under ANMs, interventional data yield a robust source of inference because the asymmetries in the observational data distribution can be easily obscured by sampling variability, particularly when the sample size is small.

2.2 RELATED WORK

Table 1 presents a summary of methods that have been proposed for causal discovery from multiple environments.

Table 1: A comparison of methods for causal discovery from multiple environments. Our method, MetaCaDI, is the first to formalize the problem using a meta-learning paradigm, enabling rapid adaptation to unseen environments.

Method	Continuous Optimization	Uncertainty Quantification	Few-Shot Adaptation
JCI-PC	×	×	×
UT-IGSP	×	×	×
DCDI-G	✓	×	×
BaCaDI	✓	✓	×
MetaCaDI (Ours)	✓	✓	✓

Early methods such as **JCI-PC** [Mooij et al., 2020] and **UT-IGSP** [Squires et al., 2020], which take *constraint-based* and *score-based* approaches, respectively, suffer from scalability issues, particularly when the number of variables d is large. **DCDI-G** [Brouillard et al., 2020] aims to improve the scalability using continuous optimization. However, all these methods provide only a single point estimate of the causal graph and cannot quantify the uncertainty inherent in small datasets. This weakness is serious in practice, especially when multiple causal graphs are equally plausible.

A principled way to capture inference uncertainty is through Bayesian inference [Eaton and Murphy, 2007]. The state-of-the-art Bayesian approach, **BaCaDI** [Hägele et al., 2023], frames the problem as multitask learning, where it treats *all* datasets as training data to jointly infer a shared causal graph and distinct intervention targets.

However, this multitask learning approach is ill-suited for real-time applications where new data arrives sequentially, as illustrated in **Example 1** in Section 1. To perform multitask learning in such streaming settings, it requires retraining the entire model from scratch to incorporate newly arrived datasets, which is computationally prohibitive for practical applications. This limitation highlights the need for a framework that can rapidly adapt to unseen datasets without full retraining, motivating a **meta-learning** approach.

3 META-LEARNING PROBLEM AND INFERENCE TARGETS

3.1 PROBLEM SETUP

We frame the problem of causal discovery from multiple environments as a meta-learning problem.

Figure 1 summarizes our meta-learning problem for inferring the shared causal graph and the distinct intervention targets. During the **meta-training phase**, we are given a collection of inference tasks $t = 1, \dots, T$, associated with *meta-training datasets* $\mathcal{D}_{\text{train}} = \{D_1, \dots, D_T\}$. Each dataset D_t is a set of N_t i.i.d. observations of $\boldsymbol{X} = [X_1, \dots, X_d]^\top$, given as $D_t = \{\boldsymbol{x}_n^t\}_{n=1}^{N_t} \overset{i.i.d.}{\sim} \mathrm{P}^{(t)}(\boldsymbol{X})$, where $\boldsymbol{x}_n^t \in \mathbb{R}^d$

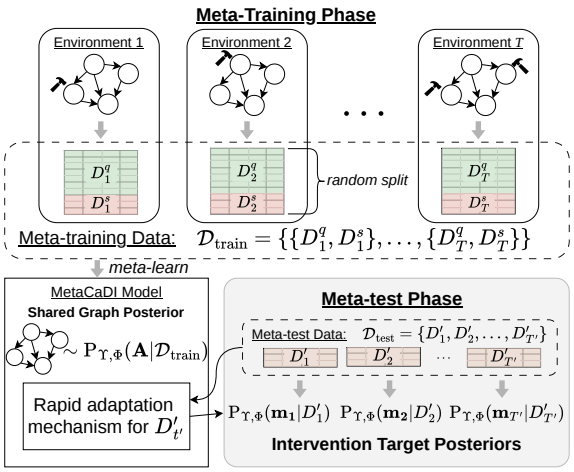

Figure 1: Our meta-learning setup. **Meta-training Phase**: Given a collection of datasets $\mathcal{D}_{\text{train}} = \{D_1, \dots, D_T\}$, MetaCaDI infers shared causal DAG adjacency matrix $\mathbf{A}$ from $\mathcal{D}_{\text{train}}$ and intervention target $\boldsymbol{m}_t$ for each D_t. **Meta-test Phase**: Using a new, small dataset $D'_{t'}$, MetaCaDI performs fine-tuning to infer its specific intervention targets.

($n \in \{1, \dots, N_t\}$) is an observation drawn from joint distribution $\mathrm{P}^{(t)}(\boldsymbol{X})$, obtained under a (hard or soft) intervention on distinct variables in $\boldsymbol{X}$. During the **meta-test phase**, we are given a new inference task t' with a *meta-test dataset* $D'_{t'}$, whose sample size can be substantially smaller than those of meta-training datasets $\mathcal{D}_{\text{train}}$.

Assigning samples to the task indices $t = 1, \dots, T$ requires domain knowledge about which data instances share identical, but unknown, intervention targets. We assume such domain knowledge; for example, in Example 2 in Section 1, patients administered the same pharmacological treatment are grouped into the same task. However, we do not assume knowledge of the intervention targets themselves. Moreover, test-time domains do not have to match training domains; their intervention targets may coincide with, overlap with, or differ from those seen during training.

Unlike the meta-learning setups that focus solely on observational data [Wu et al., 2024, Dhir et al., 2025], we consider two inference targets. One is causal graph $\mathcal{G}$ shared across all inference tasks $t = 1, \dots, T$, whose structure is represented by DAG adjacency matrix $\mathbf{A} \in \{0,1\}^{d \times d}$, which takes $A_{i,j} = 1$ if $X_i \to X_j$ in $\mathcal{G}$; otherwise, $A_{i,j} = 0$. The other is the binary intervention target vector $\boldsymbol{m}_t \in \{0,1\}^d$, which takes $m_{t,i} = 1$ if and only if the i-th variable X_i in $\boldsymbol{X}$ is intervened on; otherwise, $m_{t,i} = 0$.

To quantify the inference uncertainty of both $\mathbf{A}$ and $\{\boldsymbol{m}_t\}$, we aim to learn the parameters of variational distributions that approximate their posterior distributions.

3.2 VARIATIONAL INFERENCE VIA META-LEARNING

Our objective is to infer the joint posterior distribution over the shared graph $\mathbf{A}$ and distinct intervention targets $\{\boldsymbol{m}_t\}$ from the full collection of datasets $\mathcal{D}_{\mathcal{T}} = \{\mathcal{D}_{\text{train}}, D'_{t'}\}$ in tasks $\mathcal{T} = \{1, \ldots, T, t'\}$:

$$\mathrm{P}(\mathbf{A}, \{\boldsymbol{m}_t\}_{t\in\mathcal{T}}|\mathcal{D}_{\mathcal{T}}) = \mathrm{P}(\{\boldsymbol{m}_t\}_{t\in\mathcal{T}}|\mathbf{A}, \mathcal{D}_{\mathcal{T}})\,\mathrm{P}(\mathbf{A}|\mathcal{D}_{\mathcal{T}}). \tag{1}$$

Unfortunately, this posterior is intractable in general, as the integral for computing the marginal likelihood $\mathrm{P}(\mathbf{A}|\mathcal{D}_{\mathcal{T}})$ cannot analytically be computed without making restrictive assumptions (e.g., linear Gaussian ANMs).

For this reason, we perform posterior approximation using a variational distribution $\mathrm{P}_{\Upsilon,\Phi}$, parameterized by *task-specific parameters* Υ and *task-shared parameters* Φ, where Υ represent the components specific to each task (e.g., the intervention targets), and Φ capture the components shared across all tasks (e.g., the causal graph structure).

Using these variational parameters, we approximate the posterior in Eq. (1) as the variational distribution factorized over meta-training and meta-test tasks $\mathcal{T} = \{1, \ldots, T, t'\}$:

$$\begin{aligned}&\mathrm{P}(\mathbf{A}, \{\boldsymbol{m}_t\}_{t\in\mathcal{T}} \mid \mathcal{D}_{\mathcal{T}})\\ &\approx \mathrm{P}_{\Upsilon,\Phi}(\mathbf{A} \mid \mathcal{D}_{\mathcal{T}}) \prod_{t\in\mathcal{T}} \mathrm{P}_{\Upsilon,\Phi}(\boldsymbol{m}_t \mid \mathbf{A}, D_t).\end{aligned} \tag{2}$$

This factorization reflects two modeling assumptions. First, we assume that intervention targets for two distinct tasks $a, b \in \mathcal{T}$ ($a \neq b$) are conditionally independent given the task-shared parameters Φ: $\boldsymbol{m}_a \perp\!\!\!\perp \boldsymbol{m}_b \mid \Phi$. Second, we assume that the shared graph $\mathbf{A}$ and task-shared parameters Φ act as sufficient statistics that distill all generalizable knowledge from the other tasks. This leads to the conditional independence relation: $\boldsymbol{m}_t \perp\!\!\!\perp \mathcal{D}_{\mathcal{T}} \setminus D_t \mid \mathbf{A}, \Phi$ for any task $t \in \mathcal{T}$. These assumptions justify the factorization over $t \in \mathcal{T}$ in Eq. (2), enabling us to treat the identification of $\boldsymbol{m}_t$ as an independent task. Note that while we index all distributions by Υ and Φ for notational simplicity, Υ is empty for the shared graph $\mathbf{A}$.

Remark 3.1 (**Identifiability under infinite sample regime**). Since the posterior is dominated by likelihood in this regime, we can directly apply the existing well-established identifiability results for causal discovery from multiple environments (e.g., Brouillard et al. [2020]) to show that both $\mathbf{A}$ and $\boldsymbol{m}_t$ are **identifiable**. If functional assumptions like ANMs do not hold, the causal graph can be identified only up to the I-MEC [Tian and Pearl, 2001]; we empirically investigate the performance of our method in such cases (Appendix F.6).

Remark 3.2 (**Difficulty of finite-sample guarantees**). Due to the absence of meta-learning theory on its nonconvex, non-smooth, and bi-level optimization problem, establishing the consistency of finite-sample estimates (e.g., the estimated causal graph structure) is highly non-trivial. Given that recent neural-network-based approaches also lack such finite-sample guarantees [Brouillard et al., 2020, Hägele et al., 2023], we regard this as out of scope for this paper.

4 PROPOSED METHOD

This section presents **MetaCaDI**, a meta-learning framework for causal discovery from multiple environments.

4.1 VARIATIONAL MODEL COMPONENTS

MetaCaDI learns the variational distributions over shared causal DAG adjacency matrix $\mathbf{A}$ and distinct intervention targets $\boldsymbol{m}_t$ in task $t \in \mathcal{T}$. Below we present the three model components: (i) the causal DAG distribution, (ii) the intervention target distribution, and (iii) the likelihood model.

4.1.1 Differentiable Causal DAG Sampler

We formulate the variational distribution, $\mathrm{P}_{\Phi}(\mathbf{A} \mid \mathcal{D}_{\mathcal{T}})$ in Eq. (2), using the Differentiable Probabilistic DAG (DP-DAG) sampler [Charpentier et al., 2022], which is a DAG sampler whose parameters can be learned via a gradient-based optimization, despite the discrete nature of DAGs.

To enable gradient-based learning, the DP-DAG sampler employs a continuous relaxation of the DAG adjacency matrix. By decomposing the DAG adjacency matrix $\mathbf{A}$ using a permutation matrix $\mathbf{\Pi} \in \{0,1\}^{d\times d}$ and a strictly upper-triangular matrix $\mathbf{U} \in \{0,1\}^{d\times d}$ as $\mathbf{A} = \mathbf{\Pi}^\top \mathbf{U} \mathbf{\Pi}$, it approximately computes the gradient using continuous relaxations $\tilde{\mathbf{U}} \in \mathbb{R}^{d\times d}$ and $\tilde{\mathbf{\Pi}} \in \mathbb{R}^{d\times d}$, each of which can be sampled using the Gumbel-Softmax distribution [Jang et al., 2017, Maddison et al., 2017] and the Gumbel-Top-k trick [Kool et al., 2020], respectively.

The resulting causal DAG distribution model $\mathrm{P}_{\Phi}(\mathbf{A}|\mathcal{D}_{\mathcal{T}})$ only contains the parameters of these distributions, denoted by $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$. Since causal graph $\mathbf{A}$ is shared across all tasks, we treat them as task-shared parameters Φ.

4.1.2 Differentiable Intervention Target Identifier

Analogous to the DAG distribution model in Section 4.1.1, we model the intervention target distribution $\mathrm{P}_{\Upsilon,\Phi}(\boldsymbol{m}_t \mid \mathbf{A}, D_t)$ by designing a differentiable sampler.

Since the Bernoulli sampling operation over the binary vector $\boldsymbol{m}_t \in \{0,1\}^d$ is non-differentiable, we use a sampler of its continuous relaxation $\tilde{\boldsymbol{m}}_t \in \mathbb{R}^d$, formulated as a Gumbel-Softmax distribution with logit parameters $\boldsymbol{\zeta}^t \in \mathbb{R}^d$.

However, learning this logit vector only from dataset D_t is challenging, as intervention target inference ideally requires comparison between interventional and observational data distributions, which is infeasible in our setting. As we will

describe in Section 4.2, we address this challenge not by comparing datasets but by comparing the *predicted* values of the variables with and without interventions. We obtain these predicted values using the likelihood model.

4.1.3 Likelihood Model

To learn the parameters of the variational distributions defined in Eq. (2), we optimize the Evidence Lower Bound (ELBO) by using the sampled $\mathbf{A}$ and $\boldsymbol{m}_t$, to model the likelihood $\mathrm{P}_{\Upsilon,\Phi}(D_t \mid \mathbf{A}, \boldsymbol{m}_t)$. To enable efficient task adaptation, we instantiate the model with ANM; however, other models can be used in our general meta-learning framework.

To model the likelihood, we represent the parents $\mathrm{PA}(X_i)$ of each variable $X_i \in \boldsymbol{X}$ by masking the input variable vector $\boldsymbol{X}$ with $\boldsymbol{A}_i \in \{0,1\}^d$, obtained from the i-th column of the DAG adjacency matrix $\mathbf{A}$. Using the sampled binary intervention indicator $m_{t,i} \in \{0,1\}$ as a switch between the observational and interventional mechanisms, we formulate the structural equation of each X_i as

$$X_i = (1 - m_{t,i}) \cdot f_i(\boldsymbol{A}_i \circ \boldsymbol{X}; \boldsymbol{\Theta}_{\text{obs}}) + m_{t,i} \cdot f_i^{\text{I}}(\boldsymbol{A}_i \circ \boldsymbol{X}; \boldsymbol{\Theta}_{\text{int}}) + \epsilon_i, \quad (3)$$

where $\epsilon_i \sim \mathcal{N}(0,1)$ is standard Gaussian noise, $\circ$ denotes the Hadamard product, and f_i and f_i^{I} are functions representing the observational and interventional mechanisms, respectively. We parameterize these functions as multi-layer perceptrons (MLPs) with parameters $\boldsymbol{\Theta}_{\text{obs}}$ and $\boldsymbol{\Theta}_{\text{int}}$; we describe how we split them into Υ and Φ in Section 4.2.2.

By parameterizing f_i^{I} as a flexible function and using the binary indicator $m_{t,i}$ in Eq. (3), MetaCaDI inherently accommodates hard, soft, and complex mixtures of interventions across variables without requiring model architecture changes.

4.2 MODELING STRATEGIES FOR EFFICIENT TASK ADAPTATION

Directly sampling from the variational model (Section 4.1) to infer the graph and interventions is highly challenging, as it requires estimating a massive number of parameters from a few-shot target dataset. To address this, the following subsections introduce our feature construction (Section 4.2.1) and analytical adaptation (Section 4.2.2) methods, which efficiently reduce the parameter space.

4.2.1 Towards Efficient Logit Parameter Learning

The first challenge in few-shot adaptation in our setup is to train the logit parameters $\boldsymbol{\zeta}^t$ of the intervention target distribution, under the infeasibility of comparing interventional and observational data distributions (Section 4.1.2).

Our strategy for this challenge is twofold. First, we extract a feature vector $\boldsymbol{F}_t$ that contains the differences between the predicted variable values obtained by applying the observational and interventional mechanisms in (3) to the support set D_t^s. Then we obtain the logit parameter values by

$$\boldsymbol{\zeta}^t = g_{\boldsymbol{\eta}}(\boldsymbol{F}_t), \quad (4)$$

where $g_{\boldsymbol{\eta}}$ is a neural network with parameters $\boldsymbol{\eta}$.

This modeling strategy has two advantages. First, by leveraging the dependence of input feature $\boldsymbol{F}_t$ on distinct task t, we can treat the parameters $\boldsymbol{\eta}$ as task-shared parameters, thus reducing the number of task-specific parameters Υ, which need to be trained on small support sets. Second, it can effectively specify informative features for identifying the intervention target $\boldsymbol{m}_t$ by including them in the input feature $\boldsymbol{F}_t$.

Detailed feature design. We obtain feature vector $\boldsymbol{F}_t$ by taking three steps. First, we make predictions by applying the observational and interventional mechanisms in Eq. (3) to each data instance $\boldsymbol{x}_n^s$ in support set D_t^s:

$$[\hat{\boldsymbol{x}}^s_{\text{obs},n}]_i = f_i(\boldsymbol{A}_i \circ \boldsymbol{x}_n^s; \boldsymbol{\Theta}_{\text{obs}}) \text{ and } [\hat{\boldsymbol{x}}^s_{\text{int},n}]_i = f_i^{\text{I}}(\boldsymbol{A}_i \circ \boldsymbol{x}_n^s; \boldsymbol{\Theta}_{\text{int}}).$$

By concatenating these vectors in a row-wise manner, we obtain the two $N_t^s \times d$ matrices $\hat{\mathbf{X}}^s_{\text{obs}}$ and $\hat{\mathbf{X}}^s_{\text{int}}$, whose n-th row vectors are $\hat{\boldsymbol{x}}^s_{\text{obs},n}$ and $\hat{\boldsymbol{x}}^s_{\text{int},n}$, respectively.

Second, using the matrices $\hat{\mathbf{X}}^s_{\text{obs}}$ and $\hat{\mathbf{X}}^s_{\text{int}}$, together with the support set data matrix $\mathbf{X}_t^s \in \mathbb{R}^{N_t^s \times d}$, we engineer a feature matrix, $\mathbf{C}_t \in \mathbb{R}^{N_t^s \times 9d}$ by concatenating the following nine $N_t^s \times d$ matrices in a column-wise manner:

$$\mathbf{C}_t = \big[\mathbf{X}_t^s, \hat{\mathbf{X}}^s_{\text{obs}}, \mathbf{X}_t^s - \hat{\mathbf{X}}^s_{\text{obs}}, (\mathbf{X}_t^s - \hat{\mathbf{X}}^s_{\text{obs}})^2, \hat{\mathbf{X}}^s_{\text{int}}, \mathbf{X}_t^s - \hat{\mathbf{X}}^s_{\text{int}}, (\mathbf{X}_t^s - \hat{\mathbf{X}}^s_{\text{int}})^2, \boldsymbol{\mu}(\mathbf{X}_t^s), \boldsymbol{\sigma}(\mathbf{X}_t^s)\big],$$

where $\boldsymbol{\mu}(\mathbf{X}_t^s)$ and $\boldsymbol{\sigma}(\mathbf{X}_t^s)$ are the broadcasted column-wise mean and standard deviation of $\mathbf{X}_t^s$, respectively. This design explicitly compares the interventional and observational distributions through their residuals, i.e., the difference between observed data and SCM predictions.

Finally, we flatten the matrix $\mathbf{C}_t$ into a feature vector $\boldsymbol{F}_t$ so that the values of $\boldsymbol{F}_t$ are invariant to the order of data instances in D_t^s. To do so, we formulate the two permutation-invariant pooling layers based on the Deep Sets architecture [Zaheer et al., 2017]. Using an MLP $h_{\boldsymbol{\varsigma}} : \mathbb{R}^{9d} \to \mathbb{R}^k$ with task-shared parameters $\boldsymbol{\varsigma}$, these layers compute the **mean of embeddings** and the **embedding of the mean**:

$$\mathbf{z}_{\text{ME}} = \frac{1}{N_t^s} \sum_{n=1}^{N_t^s} h_{\boldsymbol{\varsigma}}(\boldsymbol{c}_{t,n}) \text{ and } \mathbf{z}_{\text{EM}} = h_{\boldsymbol{\varsigma}} \left(\frac{1}{N_t^s} \sum_{n=1}^{N_t^s} \boldsymbol{c}_{t,n} \right),$$

where embedding $\boldsymbol{c}_{t,n}$ is the n-th row of $\mathbf{C}_t$. Using $\mathbf{z}_{\text{ME}}$ and $\mathbf{z}_{\text{EM}}$, we obtain $\boldsymbol{F}_t$ as a vector of size $2k$:

$$\boldsymbol{F}_t = [\mathbf{z}_{\text{ME}}^\top, \mathbf{z}_{\text{EM}}^\top]^\top.$$

Remark 4.1 (**Performance gain by feature design**). Our extensive ablation studies demonstrate the strong performance gain by this feature design. The inference performance significantly degrades when removing the residual matrices in $\mathbf{C}_t$ or replacing the Deep Sets pooling with naive pooling architectures. See Appendix F.1 for details.

4.2.2 Analytical Adaptation via Closed-Form Solvers

The second challenge is to learn the MLP parameters $\boldsymbol{\Theta}_{\text{int}}$ for the interventional mechanisms f_i^{I} in Eq. (3), which are specific to each task t but whose number of parameters is typically large in our few-shot settings.

Our key idea to address this challenge is to reduce the number of task-specific parameters Υ by treating only the last MLP layer of f_i^{I}, denoted by $\boldsymbol{w}_i$, as task-specific parameters:

$$f_i^{\text{I}}(\boldsymbol{A}_i \circ \boldsymbol{X}; \boldsymbol{\Theta}_{\text{int}}) = \boldsymbol{w}_i^\top h_{\boldsymbol{\Theta}'_{\text{int}}}(\boldsymbol{A}_i \circ \boldsymbol{X}), \tag{5}$$

where $h_{\boldsymbol{\Theta}'_{\text{int}}}$ is the preceding layers parameterized by task-shared parameters $\boldsymbol{\Theta}'_{\text{int}} = \boldsymbol{\Theta}_{\text{int}} \setminus \boldsymbol{w}_i$. Crucially, this linear adaptation is applied strictly to the highly non-linear outputs of the feature extractor $h_{\boldsymbol{\Theta}'_{\text{int}}}$, preserving the expressivity to capture complex, non-linear relationships while leveraging the robust linear solver to prevent overfitting on scarce data.

Under this formulation, we can analytically obtain the optimal weights $\hat{\boldsymbol{w}}_i^{(t)}$ best fitted to the support set D_t^s by solving a least-squares problem with an ℓ_2 regularizer (i.e., Ridge regression):

$$\min_{\boldsymbol{w}_i} \|(\mathbf{X}_t^s)_{:,i} - \mathbf{H}_i \boldsymbol{w}_i\|_2^2 + \lambda \|\boldsymbol{w}_i\|_2^2, \tag{6}$$

where $(\mathbf{X}_t^s)_{:,i}$ is the i-th column of the support set data matrix $\mathbf{X}_t^s$, $\mathbf{H}_i \in \mathbb{R}^{N_t^s \times d_h}$ is the hidden feature matrix extracted by applying the preceding layers of the MLP $h_{\boldsymbol{\Theta}'_{\text{int}}}$ to each support set instance, and $\lambda \geq 0$ is a regularization parameter. The regularized least-squares problem in Eq. (6) has a well-known closed-form solution:

$$\hat{\boldsymbol{w}}_i^{(t)} = \left(\mathbf{H}_i^\top \mathbf{H}_i + \lambda \mathbf{I}\right)^{-1} \mathbf{H}_i^\top (\mathbf{X}_t^s)_{:,i}, \tag{7}$$

where $\mathbf{I}$ is the identity matrix.

As discussed in existing work on meta-learning [Snell et al., 2017, Bertinetto et al., 2019], such analytical task-specific parameter optimization enables fast and data-efficient task adaptation on D_t^s. While general-purpose meta-learning methods (e.g., MAML [Finn et al., 2017]) offer high expressivity via an iterative gradient-based optimization, they are prone to overfitting and instability when adapting to extremely small support sets (e.g., $N_t^s = 10$). Our analytical approach achieves good balance between expressivity (via the shared feature extractor) and robust estimation (via the closed-form solver), leading to superior empirical performance, as we will demonstrate in Section 5.

4.3 TRAINING OBJECTIVE

Prior to training, we randomly partition each meta-training dataset D_t into a fixed **support set** D_t^s and a **query set** D_t^q, which are illustrated in Figure 1 as the two disjoint sets with green and red data instances, respectively. The small support set mimics the few-shot setting, while the query set evaluates task adaptation via a loss function.

During the meta-training phase, MetaCaDI takes two steps for each meta-training task t. First, it fits task-specific parameters $\Upsilon = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_d\}$ to the support set D_t^s by leveraging the closed-form solution in Eq. (7). Then, with the trained Υ, it performs gradient-based updates for the task-shared parameters $\Phi = \{\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\varsigma}, \boldsymbol{\Theta}_{\text{obs}}, \boldsymbol{\Theta}'_{\text{int}}\}$ to minimize an expected loss over all meta-training tasks $t = 1, \ldots, T$.

To approximate the posterior by maximizing the ELBO, we minimize the expected loss

$$\min_{\Phi} \mathbb{E}_t[\mathbb{E}_{\mathbf{A}, \boldsymbol{m}_t}[-\log \mathrm{P}(D_t^q \mid \mathbf{A}, \boldsymbol{m}_t)] + \Omega_{\Phi,t}],$$

where $\log \mathrm{P}(D_t^q \mid \mathbf{A}, \boldsymbol{m}_t)$ is the log-likelihood over the query set D_t^q for task t, and $\Omega_{\Phi,t}$ is a regularization term.

We estimate this expected loss as an empirical average over meta-training tasks $t = 1, \ldots, T$. By approximating the expectation over $\mathbf{A}$ and $\boldsymbol{m}_t$ using a single sample from the differentiable samplers in Section 4.1, we formulate our meta-training objective as

$$\mathcal{L}(\Phi) = \frac{1}{T} \sum_{t=1}^{T} (\mathcal{L}_{\text{R},t} + \lambda_I \mathcal{L}_{\text{I},t} + \lambda_H \mathcal{L}_{\text{H},t}) + \lambda_G \mathcal{L}_G, \tag{8}$$

which is the sum of the following four loss components weighted by regularization parameters $\lambda_I, \lambda_H, \lambda_G \geq 0$:

- **Reconstruction Loss** $\mathcal{L}_{\text{R},t}$ which uses the mean squared error (MSE) over the query set D_t^q;
- **Intervention sparsity regularizer** $\mathcal{L}_{\text{I},t}$, which applies an l^1 regularizer to logit vector $\boldsymbol{\zeta}^t$ to enforce sparsity on the intervention target vector $\boldsymbol{m}_t$;
- **Residual independence regularizer** $\mathcal{L}_{\text{H},t}$, which penalizes the dependence of residuals using the Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2007] to enforce the causal sufficiency assumption;
- **Graph sparsity regularizer** $\mathcal{L}_G$, which encourages sparsity on the edge presence probability in the causal graph using a Kullback-Leibler (KL) divergence.

We detail the formulation of each component in Appendix B.

Meta-test adaptation. During the meta-test phase, we fine-tune the task-specific parameters Υ on the meta-test dataset $D'_{t'}$ using the closed-form solution in Eq. (7). While the task-shared parameters Φ could theoretically be fine-tuned as well, we hypothesize that the severe scarcity of the meta-test data would make such updates highly prone to

overfitting. Therefore, we keep Φ frozen during adaptation. We empirically validate this design choice in Section 5 by including variants (i.e., ablations) of our method that update these task-shared parameters.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Baselines. We compare MetaCaDI against four baselines: the constraint-based **JCI-PC** [Mooij et al., 2020], the score-based **UT-IGSP** [Squires et al., 2020], the continuous optimization-based **DCDI-G** [Brouillard et al., 2020], and the multitask Bayesian framework **BaCaDI** [Hägele et al., 2023]. To validate the effectiveness of our analytical adaptation (Eq. (7)), we compare against three variants that employ gradient-based MAML updates [Finn et al., 2017]: **MetaCaDI-I-MAML**, which updates only the interventional mechanism f_i^{I}; **MetaCaDI-IO-MAML**, which updates both the observational and interventional likelihood parameters; and **MetaCaDI-Full-MAML**, which updates all model parameters including those of the causal graph. We refer to our proposed method as **MetaCaDI-Analytical**.

Data. We use synthetic and semi-synthetic datasets, as a rigorous evaluation is **infeasible on real-world datasets** due to the absence of well-established benchmarks with ground-truth causal graphs and intervention targets.

Synthetic data is generated using non-linear Gaussian-Process-based ANMs with soft interventions. Semi-synthetic data is sampled from **SERGIO** [Dibaeinia and Sinha, 2020], a *realistic* simulator with parameters tuned to real gene expression data to capture complex nonlinearity and biological noise, with hard interventions.

The ground truth causal graphs for the synthetic and semi-synthetic datasets are randomly sampled from the Erdős-Rényi (ER) model and the scale-free model, respectively. In our main experiments, we set the number of variables to $d = 20$ for both datasets. We generate $T = 20$ training datasets with size $N_t = 110$ and $T' = 20$ test datasets with size $N_{t'}' = 10$. For MetaCaDI, we split each (meta-)training dataset into a support set of size $N_t^s = 10$ and a query set of size $N_t^q = 100$. See Appendix A.2 for details.

Performance Evaluation. For a fair comparison, we evaluate the test performance by enforcing a strict *single-task inference* protocol: All methods use only the single (meta-)test dataset (of size $N_{t'}' = 10$) for the specific test task being evaluated, without access to the full pool of test tasks (see Appendix C for the detailed procedure for baselines).

For graph discovery, we use the **Expected Structural Hamming Distance (E-SHD)** and the **Expected Structural Intervention Distance (E-SID)** [Peters and Bühlmann, 2015]. For intervention target identification, we use the **Interventional AUROC (Intv-AUROC)** and **Interventional AUPRC (Intv-AUPRC)**. See Appendix A for details.

5.2 RESULTS

We compute the mean and standard deviation of each performance metric over 20 independent simulations. Figure 2 shows the results on the SERGIO and synthetic datasets. Below we discuss them for each front of evaluation.

Intervention Target Identification. Despite the challenging setting with only $N_{t'}' = 10$ observations, our MetaCaDI-Analytical significantly outperforms all the four existing methods on SERGIO datasets, and achieves better or competitive performance on synthetic datasets, demonstrating its superior performance on such complex datasets as SERGIO.

On the SERGIO datasets, our method achieves the best mean Intv-AUROC of 0.740 and mean Intv-AUPRC of 0.094, whereas all of BaCaDI, DCDI-G, JCI-PC, and UT-IGSP struggle to exceed random chance (~ 0.50) in this challenging setting ($p < 0.001$). These results suggest that existing methods cannot capture the complex, realistic intervention mechanisms from the limited data, thus highlighting the importance of our meta-learning approach that enables the rapid adaptation to such scarce datasets.

Causal Graph Discovery. Compared to the existing methods, our MetaCaDI-Analytical achieves the best performance in the SERGIO datasets and proves to be better or highly competitive on the synthetic datasets, demonstrating its successful performance for extracting the shared causal graph knowledge across datasets.

On both datasets, DCDI-G and BaCaDI exhibit much poorer performance than our MetaCaDI-Analytical, despite all three methods being based on continuous optimization. A possible reason for this performance gap is that DCDI-G and BaCaDI cannot search over the space of *valid* DAGs, as they rely on a regularization term that only *encourages* acyclicity but does not strictly enforce it. By leveraging the DP-DAG sampler formulation, our MetaCaDI-Analytical effectively infers the posterior distribution over valid DAGs, thus yielding remarkable performance in graph discovery.

Ablation on Analytical Adaptation. To validate the benefit of our analytical task adaptation strategy (Section 4.2.2), we compare our MetaCaDI-Analytical against its three variants (MetaCaDI-I-MAML, MetaCaDI-IO-MAML, and MetaCaDI-Full-MAML), based on gradient-based updates.

In terms of intervention target identification (Figure 2 bottom), our MetaCaDI-Analytical outperforms all variants in 9 of 12 comparisons (3 variants $\times$ 2 datasets $\times$ 2 metrics), with statistical significance ($p < 0.05$). Moreover, as shown in the runtime comparison in Figure 3, it achieves approx-

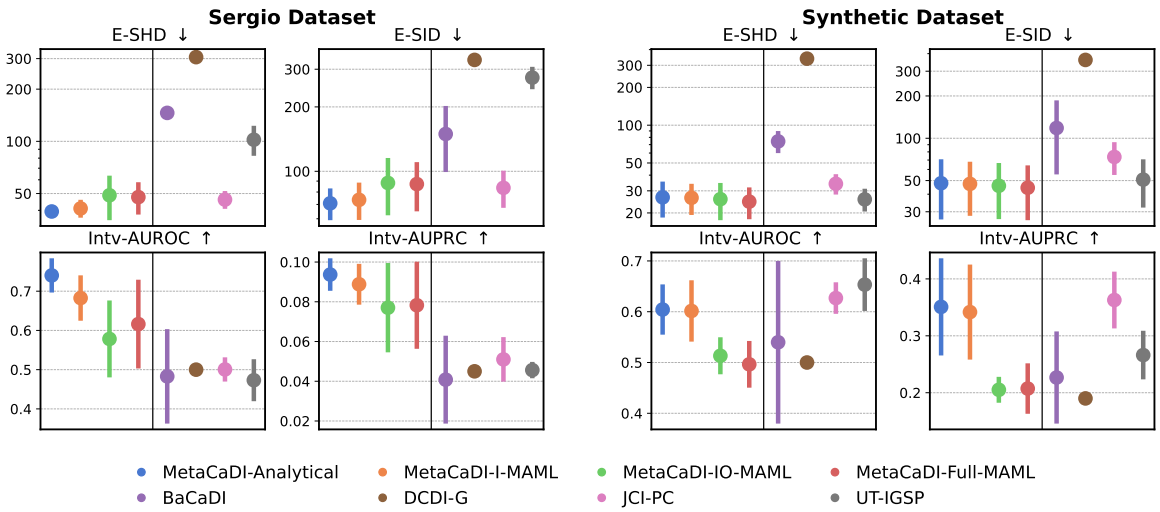

Figure 2: Performance comparison on the SERGIO and synthetic datasets. Results are averaged over 20 independent simulations per method, with error bars indicating the standard deviation. Four methods on the left of the black vertical line are based on our proposal: our method and its three variants. ↓ and ↑ indicate "lower is better" and "higher is better", respectively.

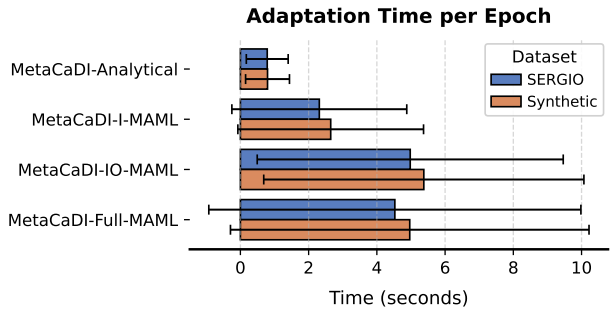

Figure 3: Runtime comparison with MetaCaDI variants

imately 5.7 times faster task adaptation than MetaCaDI-Full-MAML on average for the SERGIO datasets. This 5.7x faster adaptation directly addresses the real-time requirements of **Example 1**. Moreover, these results underscore the effectiveness and efficiency of our analytical adaptation, enabling real-time inference from limited data. More detailed runtime comparisons can be found in Appendix F.2.

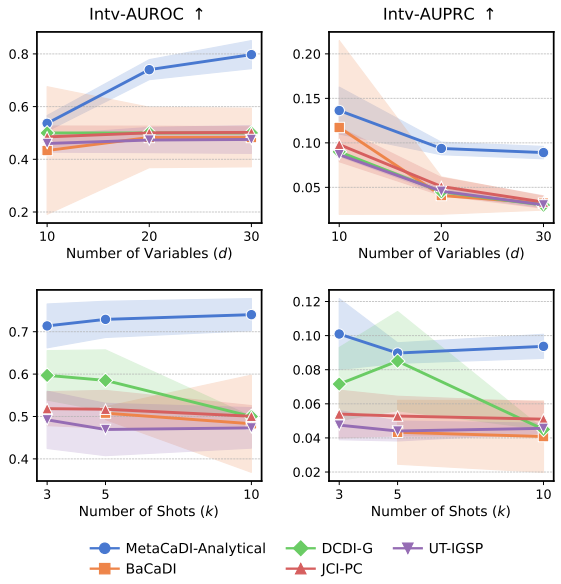

Figure 4: Performance sensitivity to the number of variables (d) and the support set size (k) on the SERGIO datasets.

5.3 ADDITIONAL EXPERIMENTS

We further test our MetaCaDI with additional experiments. We summarize the key findings, with details in Appendix F.

Computational Complexity and Scalability. We evaluate scalability by varying variables up to $d = 100$. MetaCaDI consistently outperforms baselines, with its efficiency advantage widening substantially at scale; while methods like BaCaDI and JCI-PC often timeout past 50 hours at high dimensions, MetaCaDI maintains superior performance and offers an order-of-magnitude overall speedup ($\sim$ 14.5 hours vs. $\sim$ 245 hours for BaCaDI on full pipelines). See Appendix F.3.

Robustness to Distribution Shifts. To test the robustness of MetaCaDI, we introduced two types of perturbations exclusively during meta-test data generation: (1) noise variance shifts, where the exogenous noise of each vari-

able is scaled by $\sigma \sim \text{Uniform}(0.5, 3.0)$, and (2) causal mechanism drift, where the structural functions are scaled by $\delta \sim \text{Uniform}(0.8, 1.2)$. MetaCaDI robustly maintained its structural recovery and intervention identification performance under both perturbation types. Detailed results and the formal data generation process are provided in Appendix F.5.

Few-Shot Robustness. We test MetaCaDI in the ultra-low data regime with the (meta-)test dataset size $k \in \{3, 5, 10\}$ by setting the support set size to k. In these tests, we employ a similar setup as the main experiments: using 20 meta-training tasks and 20 meta-test tasks and matching the support set size to k while fixing the query set size to 100. Figure 4 shows the results on the SERGIO datasets. Again, MetaCaDI outperforms all baselines. The performance gap is significant even in the ultra-low data regime ($k = 3$), where most baselines degrade to random chance (Intv-AUROC ≈ 0.5) or fail to adapt entirely (notably, BaCaDI). This high robustness on as few as 3 samples directly addresses the data-scarce medical challenges motivated in **Example 2**, highlighting the significance of our meta-learning framework for enabling robust inference under severe data scarcity, which is common in real-world settings. Detailed results and analysis are provided in Appendix F.4.

Ablation on Feature Design. Our MetaCaDI consistently outperforms the variants that do not use the designed features for logit parameter learning (Section 4.2.1). See Appendix F.1 for details.

6 CONCLUSION

To tackle the challenging few-shot scenarios in causal discovery from multiple environments, we introduced MetaCaDI, the first framework to cast the joint inference of causal graphs and unknown interventions as a meta-learning problem. By leveraging a closed-form analytical adaptation, our method bypasses the instability of standard meta-learning approaches, enabling rapid and robust adaptation to the few-shot scenarios typical of data-scarce real-world scenarios. Empirical results confirm that MetaCaDI significantly outperforms state-of-the-art baselines in both structure recovery and intervention identification, offering a practical solution for unraveling complex systems under severe data scarcity.

References

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *ICLR*, 2019.

Philippe Brouillard, Sébastien Lachapelle, Alexandre Lacoste, Simon Lacoste-Julien, and Alexandre Drouin. Differentiable causal discovery from interventional data. In *NeurIPS*, volume 33, pages 21865–21877, 2020.

Bertrand Charpentier, Simon Kibler, and Stephan Günnemann. Differentiable DAG sampling. In *ICLR*, 2022.

Anish Dhir, Matthew Ashman, James Requeima, and Mark van der Wilk. A meta-learning approach to Bayesian causal discovery. In *ICLR*, volume 2025, pages 14158–14178, 2025.

Payam Dibaeinia and Saurabh Sinha. SERGIO: A single-cell expression simulator guided by gene regulatory networks. *Cell Systems*, 11(3):252–271.e11, 2020. ISSN 2405-4712.

Daniel Eaton and Kevin Murphy. Exact Bayesian structure learning from uncertain interventions. In *AISTATS*, volume 2, pages 107–114, 2007.

Frederick Eberhardt and Richard Scheines. Interventions and causal inference. *Philosophy of Science*, 74(5):981–995, 2007.

Chelsea Finn, P. Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

K.J. Friston, L. Harrison, and W. Penny. Dynamic causal modelling. *NeuroImage*, 19(4):1273–1302, 2003.

Arthur Gretton, Kenji Fukumizu, Choon Teo, Le Song, Bernhard Schölkopf, and Alex Smola. A kernel statistical test of independence. In *NeurIPS*, volume 20, 2007.

Alexander Hägele, Jonas Rothfuss, Lars Lorch, Vignesh Ram Somnath, Bernhard Schölkopf, and Andreas Krause. BaCaDI: Bayesian causal discovery with unknown interventions. In *AISTATS*, volume 206, pages 1411–1436, 2023.

Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *NeurIPS*, volume 21, 2008.

Kasra Jalaldoust, Saber Salehkaleybar, and Negar Kiyavash. Multi-domain causal discovery in bijective causal models. In *CLeaR*, volume 275, pages 1268–1289, 2025.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2017.

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

Wouter Kool, Herke van Hoof, and Max Welling. Ancestral Gumbel-Top-k sampling for sampling without replacement. *JMLR*, 21(47):1–36, 2020.

Adam Li, Amin Jaber, and Elias Bareinboim. Causal discovery from observational and interventional data across multiple environments. In *NeurIPS*, volume 36, pages 16942–16956, 2023.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2017.

Leonardo Mariani, Cristina Monni, Mauro Pezzè, Oliviero Riganelli, and Rui Xin. Localizing faults in cloud systems. *ICST*, pages 262–273, 2018.

Joris M. Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *JMLR*, 21(99): 1–108, 2020.

Valeria Oliva, Ron Hartley-Davies, Rosalyn Moran, Anthony E Pickering, and Jonathan CW Brooks. Simultaneous brain, brainstem, and spinal cord pharmacological-fMRI reveals involvement of an endogenous opioid network in attentional analgesia. *eLife*, 11, 2022.

Judea Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

J. Peters and P. Bühlmann. Structural intervention distance (SID) for evaluating causal graphs. *Neural Computation*, 27(3):771–799, 2015.

Jonas Peters, Joris M. Mooij, Dominik Janzing, and Bernhard Schölkopf. Causal discovery with continuous additive noise models. *JMLR*, 15(58):2009–2053, 2014.

J.B. Rowe, L.E. Hughes, R.A. Barker, and A.M. Owen. Dynamic causal modelling of effective connectivity from fMRI: Are results reproducible and sensitive to parkinson's disease and its treatment? *NeuroImage*, 52(3):1015–1026, 2010.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, page 4080–4090, 2017. ISBN 9781510860964.

Chandler Squires, Yuhao Wang, and Caroline Uhler. Permutation-based causal structure learning with unknown intervention targets. In *UAI*, volume 124, pages 1039–1048, 2020.

Jin Tian and Judea Pearl. Causal discovery from changes. In *UAI*, page 512–521, 2001.

Pedro A. Valdes-Sosa, Alard Roebroeck, Jean Daunizeau, and Karl Friston. Effective connectivity: Influence, causality and biophysical modeling. *NeuroImage*, 58(2):339–361, 2011.

Burak Varici, Karthikeyan Shanmugam, Prasanna Sattigeri, and Ali Tajer. Intervention target estimation in the presence of latent variables. In *UAI*, volume 180, pages 2013–2023, 2022.

Hang Wu, Wenqi Shi, and May Wang. Developing a novel causal inference algorithm for personalized biomedical causal graph learning using meta machine learning. *BMC Medical Informatics and Decision Making*, 24, 05 2024.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbhakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J Smola. Deep sets. In *NeurIPS*, page 3394–3404, 2017.

MetaCaDI: A Meta-Learning Framework for Causal Discovery from Multiple Environments with Unknown Interventions (Supplementary Material)

Hans Jarett Ong[1] **Yoichi Chikahara**[2] **Tomoharu Iwata**[2]

[1]Nara Institute of Science and Technology, Nara, Japan
[2]NTT Communication Science Laboratories, Kyoto, Japan,

A BASELINES, DATASETS, AND METRICS

This section presents additional details regarding the baselines, data generation processes, and the metrics for performance evaluation.

A.1 BASELINES

We compare our proposed method, **MetaCaDI-Analytical**, against seven baselines. Four are existing methods designed for causal discovery from multiple environments with unknown interventions:

- Joint Causal Inference based on the Peter-Clark (PC) algorithm (**JCI-PC**) [Mooij et al., 2020], a constraint-based method that performs conditional independence tests involving auxiliary *context variables* representing the intervention targets.
- Unknown-Target Interventional Greedy Sparsest Permutation (**UT-IGSP**) [Squires et al., 2020], a score-based method that solves a discrete optimization problem using a greedy search, guided by a score function that evaluates the fit of a causal graph to the interventional datasets.
- Differentiable Causal Discovery from Interventional Data (**DCDI-G**) [Brouillard et al., 2020], a continuous-optimization method relying on a differentiable regularizer that encourages the acyclicity of the learned graph structure.
- Bayesian Causal Discovery with Unknown Interventions (**BaCaDI**) [Hägele et al., 2023], a multitask Bayesian framework that learns a posterior distribution over the causal graph structure and intervention targets, leveraging the shared information across multiple tasks.

The other three are variants of our method, which do not use our proposed analytical, closed-form solvers for task adaptation. In particular, we consider the following variants:

- **MetaCaDI-I-MAML**, which learns the parameters of the interventional mechanisms in our likelihood model (i.e., $\boldsymbol{\Theta}_{\text{int}}$) by performing gradient-based updates.
- **MetaCaDI-IO-MAML**, which trains all likelihood model parameters (i.e., $\boldsymbol{\Theta}_{\text{int}}$ and $\boldsymbol{\Theta}_{\text{obs}}$) using gradient-based updates.
- **MetaCaDI-Full-MAML**, which updates all model parameters, including those governing the causal graph structure, using gradient-based updates.

For a fair and standardized comparison, the implementations of all baseline methods are taken from the official BaCaDI repository[1].

[1]`https://github.com/haeggee/bacadi`

A.2 DATASETS AND GENERATION PROCESSES

Our experiments utilize fully synthetic and semi-synthetic datasets, both of which have known ground-truth causal graphs and intervention targets, thus enabling rigorous evaluation.

The **synthetic datasets** are generated using a custom pipeline adapted from Hägele et al. [2023]. We randomly generate the underlying causal graphs from the Erdős-Rényi (ER) graph models. Using the generated causal graphs, we simulate data from nonlinear ANMs, where the nonlinear function is sampled from a Gaussian Process (GP) prior with a Matérn kernel with parameter $\nu = 0.5$, and the additive noise is drawn from a Gaussian distribution with mean zero and standard deviation $\sigma = 1.5$. To generate interventional datasets, we perform soft interventions, where the function drawn from the GP prior with a Matérn kernel is replaced with a new function drawn from a GP prior with a Radial Basis Function (RBF) kernel.

The **semi-synthetic datasets** are simulated from the SERGIO simulator [Dibaeinia and Sinha, 2020], which generates realistic single-cell gene expression data. For direct comparability, we exactly follow the same simulation protocol with Hägele et al. [2023]. This protocol involves sampling the underlying causal graphs from a scale-free model with an expected node degree of 2 (SF-2), setting the number of nodes to $d = 20$, and simulating interventional datasets by performing $M = 10$ hard interventions, which force the production rate of each target gene to zero.

We use these data-generation procedures to create (meta-)training and (meta-)test datasets. For our MetaCaDI-Analytical and its variants, we generate $T' = 20$ datasets, each comprising only $N'_{t'} = 10$ data instances to simulate the challenging few-shot inference scenario.

A.3 EVALUATION METRICS

To provide a comprehensive evaluation, we assess performance on two distinct tasks: causal graph discovery and intervention target identification. For metrics calculated over a posterior distribution, we report the expected value by averaging the metric over all posterior samples.

To evaluate the performance of causal graph discovery, we use four metrics:

- Expected Structural Hamming Distance (**E-SHD**), which is the expected number of edge additions, deletions, or reversals needed to convert the inferred causal graph structure to the ground truth graph. Lower is better.
- Expected Structural Intervention Distance (**E-SID**), which evaluates the ability of the inferred causal graph structure to identify interventional distributions [Peters and Bühlmann, 2015]. In particular, this metric evaluates whether the inferred graph structure correctly specifies the valid adjustment set for the identification of interventional distributions. Roughly speaking, it takes a high value if the inferred graph structure leads to significantly incorrect causal implications. Lower is better.

Regarding intervention target identification, we use the following two classification metrics:

- Interventional AUROC **(Intv-AUROC)**, which measures the AUROC score in the task of identifying true intervention targets. Higher is better.
- Interventional AUPRC **(Intv-AUPRC)**, which evaluates the AUPRC score in intervention target identification. As with edge AUPRC, it is better suited than Intv-AUROC for the sparse intervention setting, where only a few variables are intervened in all datasets. Higher is better.

B DETAILED TRAINING OBJECTIVE

As described in Section 4.3, we learn the task-shared parameters $\Phi = \{\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\eta}, \boldsymbol{\Theta}_{\text{obs}}, \boldsymbol{\Theta}'_{\text{int}}, \boldsymbol{\varsigma}\}$ by minimizing the following loss function:

$$\mathcal{L}(\Phi) = \frac{1}{T}\sum_{t=1}^{T}(\mathcal{L}_{\text{R},t} + \lambda_I \mathcal{L}_{\text{I},t} + \lambda_H \mathcal{L}_{\text{H},t}) + \lambda_G \mathcal{L}_G.$$

Below we detail the four components of this loss function.

- **Reconstruction Loss** $\mathcal{L}_{\mathrm{R},t}$ is the expected negative log-likelihood of the query set D_t^q for task t. Under the standard Gaussian additive noise assumption, such negative log-likelihood equals the Mean Squared Error (MSE):

$$\mathcal{L}_{\mathrm{R},t} = -\log \mathrm{P}(D_t^q \mid \mathbf{A}, \boldsymbol{m}_t) \tag{9}$$

$$= \frac{1}{d \cdot N_t^q} \sum_{i=1}^{d} \sum_{n=1}^{N_t^q} (x_{n,i}^q - \hat{x}_{n,i}^q)^2, \tag{10}$$

  where $\hat{x}_{n,i}^q$ is the predicted value of variable X_i for the n-th observation $\boldsymbol{x}_n^q$ in the query set D_t^q.

- **Intervention Sparsity Loss** $\mathcal{L}_{\mathrm{I},t}$ is a sparsity constraint on the predicted probabilities that each variable X_i in $\boldsymbol{X}$ is intervened in task t. Formally, it imposes a ℓ^1 penalty on the probabilities derived from the logit vector $\boldsymbol{\zeta}^t$ of intervention target identification for task t as

$$\mathcal{L}_{\mathrm{I},t} = \|\sigma(\boldsymbol{\zeta}^t)\|_1,$$

  where $\sigma(\cdot)$ is the sigmoid function.

- **Residual Independence Loss** $\mathcal{L}_{\mathrm{H},t}$ is a penalty term for encouraging the independence among residuals $x_{n,i}^q - \hat{x}_{n,i}^q$ in Eq. (10) to enforce the causal sufficiency assumption. To measure this independence, we use the Hilbert-Schmidt Independence Criterion (HSIC) [Gretton et al., 2007] for this purpose.

- **Graph Sparsity Loss** $\mathcal{L}_G$ is a KL divergence term that encourages sparsity in edge presence probabilities. Letting $\mathrm{P}_{\boldsymbol{\phi}}(\mathbf{U})$ be the Gumbel-Softmax-based variational distribution over upper-triangular matrix $\mathbf{U} \in \{0,1\}^{d \times d}$ (Section 4.1.1), this KL divergence can be expressed as

$$\mathcal{L}_G = \mathrm{KL}(\mathrm{P}_{\boldsymbol{\phi}}(\mathbf{U}) \| \, \mathrm{P}(\mathbf{U})),$$

  where $\mathrm{P}(\mathbf{U})$ is a prior distribution, which we set to be a Bernoulli distribution with small success probabilities; see Table 2 for the setting of this success probability hyperparameter, p_G.

C BASELINE ADAPTATION PROTOCOL

This section presents the detailed procedure for making a fair comparison between our meta-learning-based method(s) and the baselines under their problem setup difference. Baselines are designed for the *pooled* setting, where each method accesses the entire batch of (meta-)test tasks simultaneously, denoted by $\mathcal{D}_{\text{test}} = \{D'_1, \ldots, D'_{T'}\}$. In contrast, our **MetaCaDI** operates in a *sequential* manner, adapting to a single meta-test task without access to other, concurrent test tasks.

For a fair comparison, we evaluate the test performance by enforcing a strict single-task inference protocol: All methods use only the single (meta-)test dataset for the specific test task being evaluated, without access to the full pool of test tasks.

To fairly compare the performance under such a single-task inference protocol, we take two steps.

First, for the point-estimation methods (JCI-PC, UT-IGSP, and DCDI-G), we measure the uncertainty in their estimates by employing the standard bootstrapping procedure. This bootstrapping procedure requires repeatedly running each method on multiple bootstrap samples, which is computationally prohibitive. For this reason, we used 5 bootstrap samples for DCDI and 20 samples for the other methods.

Second, we enforce the single-task inference protocol for each method as follows.

UT-IGSP and JCI-PC. These methods perform joint inference and hence require access to training and test tasks. To adapt them to the sequential setting, we provided the algorithms with all meta-training datasets plus a single meta-test dataset at a time, averaging the results per bootstrap run. For JCI-PC, due to its prohibitive computational time, we subsampled 10 random training contexts per bootstrap run rather than using the full training set.

DCDI-G. Standard DCDI jointly learns the graph and intervention targets via continuous optimization. Running this from scratch for every few-shot test task is computationally infeasible and prone to overfitting. We therefore implemented a two-stage strategy: In Stage 1 (**Graph Learning**), we train DCDI solely on the (meta-)training data to learn the shared causal graph structure $\mathcal{G}$. In Stage 2 (**Episodic Adaptation**), for each isolated (meta-)test task $D'_{t'}$, we freeze the graph structure learned in Stage 1 and optimize only the intervention targets (regime-specific parameters). This approach ensures fairness with MetaCaDI-Analytical (which effectively freezes task-shared parameters during adaptation) and prevents DCDI from overfitting structural parameters to the sparse ($N'_{t'} = 10$) test data.

BaCaDI. We utilized the publicly available scripts for BaCaDI in its standard multitask setting but adapted the data input to the episodic protocol. Instead of providing the full pool of test datasets, we applied the method to the set of all training datasets plus a single test dataset at a time. This ensures the method operates without information leakage from concurrent test environments, maintaining consistency with the evaluation pipeline used for the other baselines.

D METACADI IMPLEMENTATION DETAILS

Neural Network Architectures. To parameterize the likelihood model in Eq. (3), we use two multi-layer perceptrons (MLPs) for each variable X_i: one for the observational mechanism f_i and another for the feature extractor in the interventional mechanism f_i^{I}. We use rectified linear unit (ReLU) activation for all hidden layers and set the number of hidden layers and the number of hidden units in each layer as hyperparameters, which were tuned via random search over the value range presented in Table 2 (see Appendix E for details).

To formulate the intervention target identifier $g_{\boldsymbol{\eta}}$ in Eq. (4), we use an MLP. To obtain its input feature vector $\boldsymbol{F}_t$, we use an MLP $h_{\boldsymbol{\varsigma}}$ with a permutation-invariant architecture that yields the feature representation invariant to the order of data instances in a support set D_t^s (Section 4.2.1). To the logit outputs of $g_{\boldsymbol{\eta}}$, we applied layer normalization (`LayerNorm`) to stabilize training.

Parameter Optimization. We train the task-shared parameters Φ of our MetaCaDI-Analytical by performing differentiable sampling with the *straight-through* estimators [Bengio et al., 2013]. In the forward pass, we evaluate the objective function value by sampling the causal DAG adjacency matrix $\mathbf{A} \in \{0,1\}^{d\times d}$ and the intervention targets $\boldsymbol{m}_t \in \{0,1\}^d$, where $\mathbf{A} = \mathbf{\Pi}^\top \mathbf{U} \mathbf{\Pi}$ is drawn by sampling an upper triangular matrix $\mathbf{U} \in \{0,1\}^{d\times d}$ from the Gumbel Softmax distribution and a permutation matrix $\mathbf{\Pi} \in \{0,1\}^{d\times d}$ from the Gumbel-Top-k distribution. To sample these discrete matrices and vectors, we take $\arg\max$ over the sampled continuous relaxations: e.g., for $\mathbf{U}$, we compute its (i,j) element $U_{i,j} \in \{0,1\}$ by taking $U_{i,j} = \arg\max[1-\tilde{U}_{i,j}, \tilde{U}_{i,j}]$, where $\tilde{U}_{i,j}$ is the continuous relaxation of $U_{i,j}$ sampled from the Gumbel-Softmax distribution. In the backward pass, we approximately compute the gradients by using such continuous relaxations as $\tilde{U}_{i,j}$.

By employing such gradients with the Adam optimizer [Kingma and Ba, 2014], we trained the task-shared parameters Φ. We used a `ReduceLROnPlateau` learning rate scheduler and an early stopping mechanism, both monitored on the validation set's ELBO loss, to prevent overfitting. As with the standard Gumbel-Softmax-based optimization, we annealed the Gumbel temperature for the graph sampler from a higher initial value to a lower final value over a specified percentage of the training epochs to encourage exploration early and exploitation later. By contrast, for the intervention target sampler, we used a separate, fixed Gumbel temperature. All hyperparameters, including network architectures, learning rates, regularization weights, and temperature schedules, were tuned using random search, as detailed in Appendix E.

Regarding the task-specific parameters Υ, which consists of the final linear layer of the interventional mechanism f_i^I, we do not use gradient-based optimization for our MetaCaDI-Analytical. Instead, we analytically computed the optimal values of its weights $\hat{\mathbf{w}}_i^{(t)}$ for each task t using a closed-form solution to a Ridge regression problem (see Section 4.2).

E HYPERPARAMETER SEARCH AND SETTINGS

For each simulation run, we performed a 100-trial random search over the hyperparameter space detailed in Table 2. For this search, the meta-training datasets were partitioned into a training set (80% of the tasks) and a validation set (the remaining 20% of tasks). In each trial, a model was trained with a randomly sampled configuration on the 80% training partition. We used early stopping monitored on the validation set's ELBO to determine the optimal number of training epochs for that trial. After all 100 trials were completed, we selected the single best-performing hyperparameter configuration based on the highest Intv-AUPRC achieved on the validation set. The final metrics reported in the main paper were then computed by retraining a model with this best configuration on the full meta-training data and evaluating it on the held-out test set.

F ADDITIONAL EXPERIMENTS

F.1 ABLATION STUDY ON INTERVENTION TARGET PREDICTOR

To validate the effect of the feature and architectural design of the intervention target predictor described in Section 4.2, we performed an ablation study to isolate the contribution of two key design choices: (1) the engineering of the nine-component

Table 2: Hyperparameter search space for the random search conducted for each simulation run.

Parameter	Distribution	Range / Values
Learning Rate (lr)	Log-Uniform	[1e-4, 5e-2]
Weight Decay	Log-Uniform	[1e-5, 1.0]
Gradient Clipping Norm	Categorical	{0.0, 0.5, 1.0, 2.0}
MLP Hidden Layers	Categorical	{0, 1, 2, 3}
MLP Hidden Units Ratio	Uniform	[0.25, 1.5]
Intervention Predictor Hidden Layers	Categorical	{1, 2, 3, 4}
Intervention Predictor Hidden Units Ratio	Uniform	[0.5, 2.0]
Intervention Sparsity Weight (λ_I)	Uniform	[0.0, 5.0]
Residual Independence Weight (λ_H)	Uniform	[0.0, 10.0]
Reconstruction Loss Weight	Log-Uniform	[0.01, 10.0]
Graph Sparsity Prior (p_G)	Log-Uniform	[1e-4, 0.1]
LR Plateau Patience	Quantized Uniform	[15, 30]
Early Stopping Patience	Quantized Uniform	[40, 60]
LR Plateau Factor	Uniform	[0.1, 0.5]
Initial Gumbel Temperature (Graph)	Uniform	[1.0, 5.0]
Final Gumbel Temperature (Graph)	Uniform	[0.1, 0.5]
Gumbel Anneal Ratio (Graph)	Uniform	[0.7, 0.9]
Gumbel Temperature (Intervention)	Uniform	[0.5, 2.0]
MAML-Specific Parameters		
Inner Loop Learning Rate	Categorical	{0.005, 0.01, 0.05}
Inner Loop Steps	Categorical	{1, 3, 5}

input feature vector $\boldsymbol{F}_t$, and (2) the permutation-invariant Deep Sets architecture.

F.1.1 Input Feature Ablations

We assessed the effect of the engineered features by training variants of MetaCaDI with reduced input sets:

- **Raw Data Only (Ablation 1A):** The model receives only the support set data D_t^s, excluding all predicted values and residuals.
- **No Residuals (Ablation 1B):** The model receives the support set data and statistical summaries but explicitly excludes the predicted values ($\hat{\mathbf{X}}^s_{\text{obs}}, \hat{\mathbf{X}}^s_{\text{int}}$) and their residuals.
- **No Statistical Summaries (Ablation 1C):** The model uses all per-sample features (including residuals) but excludes the broadcasted mean and standard deviation.

F.1.2 Architecture Ablations

We evaluated the utility of the Deep Sets architecture against simpler aggregation methods:

- **Naive Mean Pooling (Ablation 2A):** The Deep Sets architecture is replaced by a simple column-wise mean of the feature matrix $\mathbf{C}_t$.
- **Single-Branch Pooling (Ablation 2B):** We tested using only the "Mean of Embeddings" branch ($\mathbf{z}_{\text{ME}}$) or only the "Embedding of Means" branch ($\mathbf{z}_{\text{EM}}$), rather than their concatenation.
- **Flattened MLP (Ablation 2C):** We removed the permutation invariance constraint by flattening the feature matrix $\mathbf{C}_t$ into a single vector and feeding it into a standard MLP.

F.1.3 Results and Analysis

We summarize the results on both SERGIO and Synthetic datasets in Table 3.

Necessity of Residuals. The "base" MetaCaDI-Analytical model significantly outperforms Ablation 1B (No Residuals). On the Synthetic dataset, removing residuals caused the Intv-AUPRC to drop from 0.351 to 0.233. This confirms that the model relies heavily on the *residuals*—the discrepancy between the observed data and the SCM predictions—to identify intervention targets, rather than simply inferring them from the raw data distribution.

Impact on Graph Recovery. Crucially, the degradation in the intervention identifier leads to a collapse in graph recovery performance. For example, in the Raw Data Only ablation (1A), the E-SID on Synthetic data worsened from 47.88 to 109.47. This highlights the tight coupling in our joint inference framework: accurate intervention target identification is a prerequisite for reliable causal discovery.

Benefits of Deep Sets. The architectural ablations demonstrate that the Deep Sets formulation is superior to Naive Mean Pooling (2A) and Flattened MLP (2C). The drop in performance for the Flattened MLP suggests that enforcing permutation invariance provides a strong inductive bias that prevents overfitting to the specific ordering of the few-shot support set.

Table 3: Ablation study results on SERGIO and Synthetic datasets. **Base** refers to the proposed MetaCaDI-Analytical method. Results are reported as mean ± standard deviation over 20 simulations. The best results are bolded. ↓ and ↑ indicate "lower is better" and "higher is better", respectively.

Method	SERGIO Dataset				Synthetic Dataset			
	Intv-AUROC ↑	Intv-AUPRC ↑	E-SHD ↓	E-SID ↓	Intv-AUROC ↑	Intv-AUPRC ↑	E-SHD ↓	E-SID ↓
MetaCaDI-Analytical	**0.740 ± 0.038**	**0.094 ± 0.007**	**39.39 ± 1.51**	**70.87 ± 10.32**	**0.604 ± 0.045**	**0.351 ± 0.081**	**26.63 ± 7.47**	**47.88 ± 20.44**
Feature Ablations								
1A: Raw Data Only	0.668 ± 0.082	0.078 ± 0.015	57.16 ± 23.53	100.49 ± 38.16	0.545 ± 0.040	0.242 ± 0.041	64.08 ± 35.73	109.47 ± 59.18
1B: No Residuals	0.688 ± 0.063	0.083 ± 0.013	57.32 ± 23.49	100.70 ± 37.85	0.537 ± 0.026	0.233 ± 0.019	63.75 ± 35.62	108.98 ± 59.16
1C: No Stat. Summaries	0.676 ± 0.087	0.085 ± 0.017	56.89 ± 23.34	100.06 ± 37.93	0.546 ± 0.040	0.244 ± 0.037	63.91 ± 35.58	109.21 ± 58.93
Architecture Ablations								
2A: Naive Mean Pool	0.675 ± 0.072	0.086 ± 0.016	57.15 ± 23.27	100.57 ± 37.78	0.581 ± 0.038	0.299 ± 0.061	64.17 ± 35.58	109.95 ± 59.08
2Bi: Only $\mathbf{z}_{\text{ME}}$	0.676 ± 0.068	0.081 ± 0.012	57.44 ± 23.53	101.04 ± 37.80	0.540 ± 0.035	0.235 ± 0.032	64.24 ± 35.73	109.71 ± 59.25
2Bii: Only $\mathbf{z}_{\text{EM}}$	0.680 ± 0.064	0.082 ± 0.013	57.43 ± 23.57	101.32 ± 38.08	0.545 ± 0.037	0.240 ± 0.034	63.97 ± 35.60	109.33 ± 58.90
2C: Flattened MLP	0.666 ± 0.082	0.086 ± 0.023	56.94 ± 23.34	99.84 ± 37.76	0.567 ± 0.037	0.287 ± 0.052	63.97 ± 35.58	109.59 ± 59.18

F.2 COMPUTATIONAL RUNTIME COMPARISON

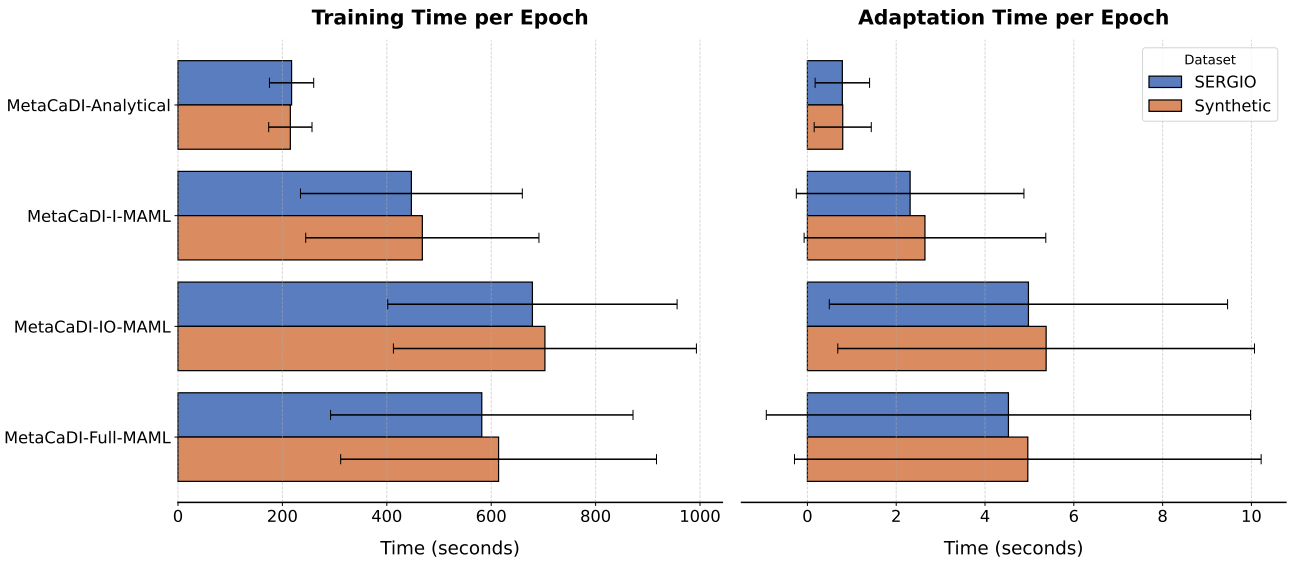

Figure 5: Per-epoch runtime analysis for MetaCaDI variants. The plots compare the average training time (left) and adaptation time (right) per epoch on the SERGIO and Synthetic datasets. Error bars represent the standard deviation over 20 simulations.

We present runtime from two perspectives: Figure 5 compares our proposed method against its MAML-based variants on a per-epoch basis for a precise analysis of the efficiency gains from our analytical adaptation strategy. Figure 6 shows the total runtime per dataset in order to make it comparable with the baselines.

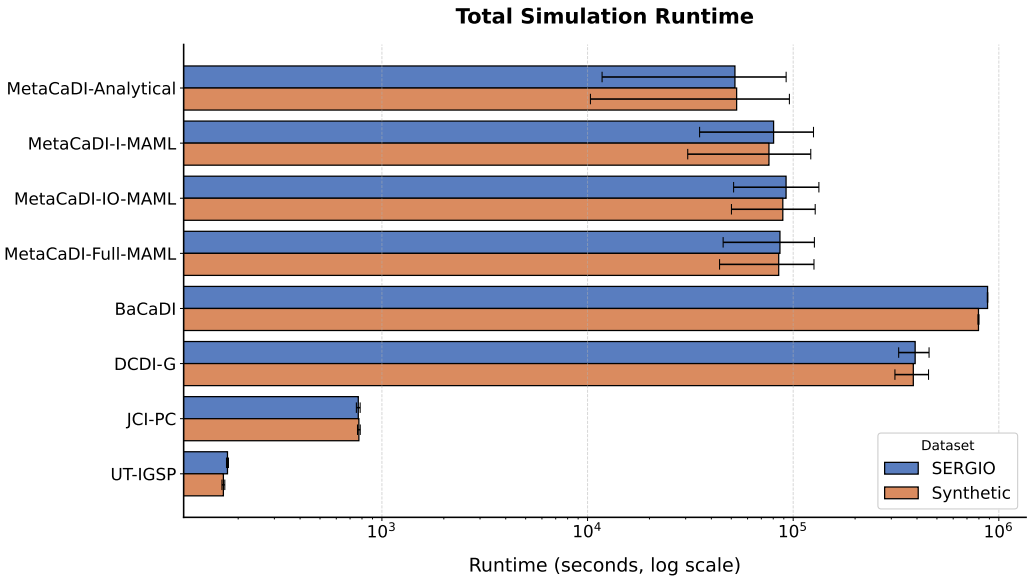

Figure 6: Average total runtime per dataset. The plot compares the time required to learn both the shared causal graph and the intervention targets for each of the 20 test tasks. Note that the x-axis uses a logarithmic scale to accommodate the wide range of values. Error bars represent the standard deviation.

The per-epoch results in Figure 5 clearly demonstrate the efficiency of the analytical approach. On the SERGIO dataset, MetaCaDI-Analytical is approximately **2.7 times faster** per training epoch than MetaCaDI-Full-MAML (217.59 vs. 581.95 [sec.]). The benefit is even more pronounced in the task-adaptation step, where MetaCaDI-Analytical is approximately **5.7 times faster** than MetaCaDI-Full-MAML (0.79 vs. 4.53 [sec.]).

In terms of total runtime per dataset, MetaCaDI-Analytical demonstrates superior scalability compared to other continuous optimization and Bayesian frameworks. BaCaDI is the most computationally intensive, requiring an average of ~ 245 hours to complete. DCDI-G also incurs a high cost, averaging ~ 109 hours. In contrast, MetaCaDI-Analytical completes the full pipeline for a dataset in ~ 14.5 hours, offering an order-of-magnitude speedup over comparable continuous baselines.

F.3 SENSITIVITY TO NUMBER OF VARIABLES d

To further characterize the behavior of MetaCaDI, we conducted a sensitivity analysis on both the SERGIO and Synthetic datasets by varying the number of variables $d \in \{10, 20, 30\}$.

Experimental Setup. To ensure a rigorous comparison, we adhered to the data generation and evaluation protocols described in Section 5 and Appendix A.2. For all settings ($d \in \{10, 20, 30\}$), we maintained the same sample size ($N_t = 110$) and meta-training/test splits. Crucially, to maintain a consistent level of difficulty relative to the system size, the number of ground-truth intervention targets in each environment was scaled proportionally to the number of nodes. Specifically, we targeted 20% of the variables in each experimental setting (i.e., 2, 4, and 6 targets for $d = 10, 20,$ and 30, respectively).

Performance Trends. The quantitative results are visualized in Figure 7. On the SERGIO dataset, we observe a divergence between the two intervention identification metrics. The Intv-AUROC performance of MetaCaDI-Analytical improves as the system size increases, rising from 0.536 at $d = 10$ to 0.797 at $d = 30$. While Intv-AUPRC does not show the same increasing trend, our MetaCaDI-Analytical still outperforms all baselines.

Regarding graph structure recovery, we observe that error metrics (E-SHD and E-SID) naturally increase with the super-linear growth of the graph search space. For instance, at $d = 30$, MetaCaDI achieves an E-SHD of 66.7, whereas BaCaDI and DCDI-G degrade significantly to 355.3 and 756.0, respectively. We note that MetaCaDI-Analytical performs competitively in all these cases.

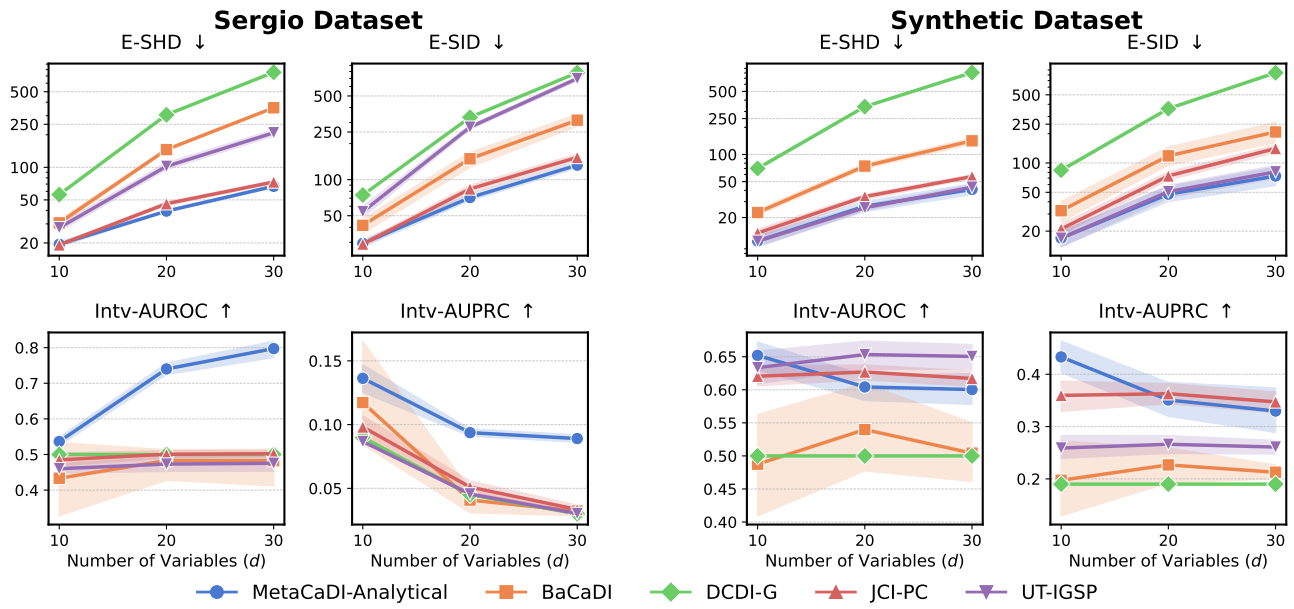

Figure 7: Performance sensitivity to graph size (d) on the SERGIO and synthetic datasets. Results are averaged over 20 independent simulations. MetaCaDI-Analytical (blue circles) is compared against baselines across $d \in \{10, 20, 30\}$. ↓ and ↑ indicate "lower is better" and "higher is better", respectively.

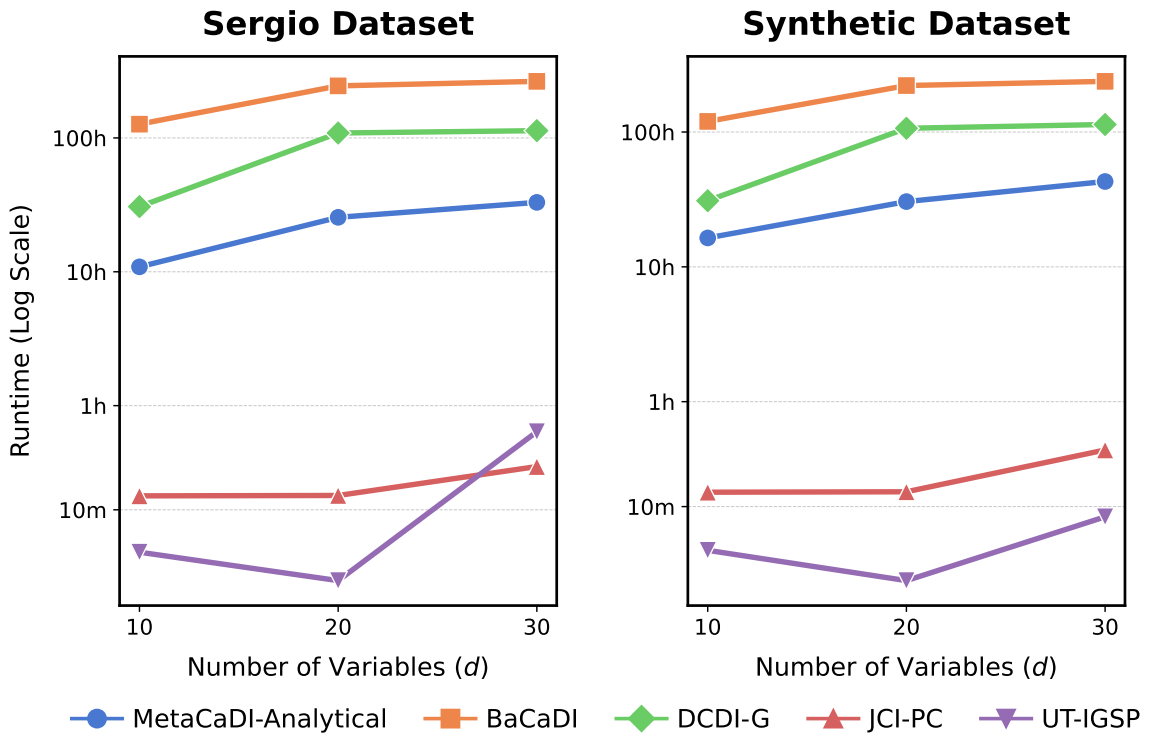

Figure 8: **Runtime scalability** (d)**.** The total runtime (training + evaluation) is plotted on a logarithmic scale. MetaCaDI-Analytical offers significant speedups over BaCaDI and DCDI-G, maintaining manageable runtimes even as the system size scales.

Runtime Scalability. A key advantage of our proposed analytical adaptation is its computational efficiency. Figure 8 illustrates the total runtime (in hours) required for a full simulation run. MetaCaDI-Analytical exhibits superior scalability

compared to the other continuous optimization frameworks. For the largest setting ($d = 30$) on the SERGIO dataset, MetaCaDI completes the full pipeline in approximately 33.0 hours. In stark contrast, DCDI-G requires 113.3 hours ($3.4\times$ slower), and BaCaDI requires 264.6 hours ($8\times$ slower). While the constraint-based (JCI-PC) and score-based (UT-IGSP) methods are faster due to their combinatorial nature on small graphs, they often fail to achieve comparable accuracy on the complex, non-linear SERGIO benchmarks. These results confirm that MetaCaDI provides an optimal balance, offering the high expressivity of neural networks without the prohibitive computational costs typically associated with bi-level optimization in larger causal graphs.

F.4 FEW-SHOT ROBUSTNESS

To assess the robustness of MetaCaDI in the extreme low-data regime, we evaluated performance by varying the support set size $k \in \{3, 5, 10\}$ used for identifying intervention targets in the meta-test tasks. The results on both the SERGIO and Synthetic datasets are visualized in Figure 9.

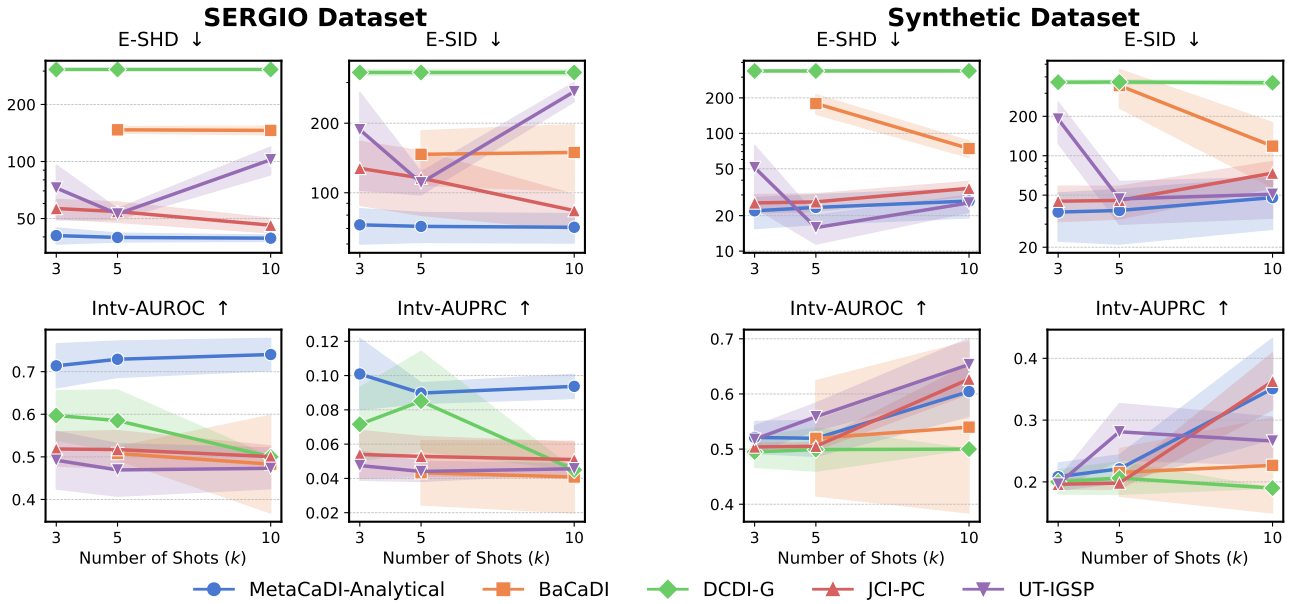

Figure 9: **Performance sensitivity to support set size (k).** The plots display the mean and standard deviation of performance metrics over 20 independent simulations for varying shot counts $k \in \{3, 5, 10\}$ on the SERGIO (left) and Synthetic (right) datasets.

Robustness in Complex Environments (SERGIO). On the realistic SERGIO dataset, MetaCaDI demonstrates exceptional stability. Crucially, it consistently achieves the highest Intv-AUROC and Intv-AUPRC and the lowest structural errors (E-SHD, E-SID) across all evaluated sample sizes (k).

- **Intervention Identification:** At $k = 3$, MetaCaDI achieves an Intv-AUROC of 0.714 ± 0.052 and an Intv-AUPRC of 0.101 ± 0.021. In contrast, most baseline methods fail to extract meaningful signals in this ultra-low data regime. While DCDI-G performs slightly better than random chance (Intv-AUROC ≈ 0.597), it still lags significantly behind our method. **Notably, BaCaDI fails to adapt to this extreme sparsity,** and JCI-PC (≈ 0.519) similarly yields results close to random guess. This gap confirms that our meta-learning framework successfully leverages shared knowledge to regularize inference when task-specific data is insufficient for standalone discovery.
- **Structural Stability:** The structural error metrics for MetaCaDI remain remarkably consistent across sample sizes (E-SHD ≈ 40, E-SID ≈ 71 for all k). Conversely, baselines exhibit high sensitivity or instability. JCI-PC degrades from an E-SHD of 46.1 ($k = 10$) to 56.5 ($k = 3$), while UT-IGSP shows erratic behavior, with E-SHD fluctuating from 102.2 ($k = 10$) to 52.9 ($k = 5$) and back to 72.5 ($k = 3$).

Threshold Effects in Synthetic Data. In the Synthetic setting, we observe a distinct performance threshold for intervention identification.

- **Sample Efficiency:** At $k = 3$ and $k = 5$, both MetaCaDI and the baselines struggle to identify intervention targets with high precision, with Intv-AUPRC scores hovering around $0.20 - 0.22$. However, at $k = 10$, MetaCaDI exhibits a sharp performance improvement, reaching an Intv-AUPRC of 0.351 ± 0.081. This behavior mirrors that of the strongest baseline, JCI-PC, which also sees a similar increase to 0.363 ± 0.046 at $k = 10$.
- **Interpretation:** This suggests that in these synthetic environments, a minimum sample size (around $k = 10$) is likely required for the statistical signatures of interventions to become distinguishable from observational noise. Once this threshold is met, MetaCaDI remains competitive with the best-performing baselines.

Summary. These results highlight a key advantage of MetaCaDI: while it performs competitively to standard methods in idealized settings with sufficient data ($k = 10$, Synthetic), it provides a critical safety net in complex, data-scarce scenarios ($k = 3$, SERGIO), where it is the only method capable of reliable inference.

F.5 ROBUSTNESS TO ENVIRONMENTAL DISTRIBUTION SHIFTS

To systematically evaluate robustness to distribution shifts, we modify the Structural Causal Model (SCM) generation process for the unseen meta-test tasks. Let $X_j = f_j(\text{Pa}_j) + \epsilon_j$ be the base SCM for a given environment. We apply the following perturbations during the meta-test phase:

1. **Task-Dependent Noise Variances (σ):** The exogenous noise is scaled such that the generated observational data follows $X_j = f_j(\text{Pa}_j) + (\sigma \cdot \epsilon_j)$, where $\sigma \sim \text{Uniform}(0.5, 3.0)$. This tests the algorithm's ability to recover structures when the signal-to-noise ratio drastically changes.
2. **Causal Mechanism Drift (δ):** The functional mapping from parents to children is altered such that $X_j = (\delta \cdot f_j(\text{Pa}_j)) + \epsilon_j$, where $\delta \sim \text{Uniform}(0.8, 1.2)$. This tests robustness against deterministic changes in the underlying causal dynamics.

Table 4 summarizes the performance of MetaCaDI against the baselines under these distribution shifts. MetaCaDI-Analytical maintains highly robust structural recovery (achieving the lowest E-SHD) despite the environmental drift. The closed-form analytical solver used for task adaptation effectively acts as a regularized linear layer over the deeply learned shared features, preventing the model from overfitting to the shifted noise or drifting mechanisms.

In contrast, standard continuous multi-task approaches (BaCaDI and DCDI-G) exhibit catastrophic structural degradation under both shifts, suffering from excessively high E-SHD scores as they fail to disentangle the shifted environments from the underlying causal structure. While the score-based UT-IGSP demonstrated partial robustness to noise variance shifts—performing competitively in intervention identification—it degraded significantly under the more complex causal mechanism drift, a scenario where MetaCaDI continued to dominate across all metrics.

Table 4: Performance under environmental distribution shifts on the Synthetic dataset ($d = 20$, $k = 10$).

Method	**Intv-AUPRC** $\uparrow$	**Intv-AUROC** $\uparrow$	**E-SHD** $\downarrow$
Shift 1: Task-Dependent Noise Variances ($\sigma \sim$ Uniform$(0.5, 3.0)$)			
MetaCaDI-Analytical	$\mathbf{0.293 \pm 0.080}$	0.592 ± 0.065	$\mathbf{8.845 \pm 3.668}$
BaCaDI	0.187 ± 0.055	0.462 ± 0.110	138.559 ± 8.013
DCDI-G	0.190 ± 0.000	0.500 ± 0.000	367.700 ± 4.169
JCI-PC	0.215 ± 0.017	0.552 ± 0.027	38.490 ± 5.921
UT-IGSP	0.276 ± 0.033	$\mathbf{0.647 \pm 0.037}$	11.980 ± 2.303
Shift 2: Causal Mechanism Drift ($\delta \sim$ Uniform$(0.8, 1.2)$)			
MetaCaDI-Analytical	$\mathbf{0.253 \pm 0.044}$	$\mathbf{0.578 \pm 0.041}$	$\mathbf{7.521 \pm 2.563}$
BaCaDI	0.211 ± 0.035	0.506 ± 0.110	136.737 ± 7.869
DCDI-G	0.190 ± 0.000	0.500 ± 0.000	368.300 ± 4.692
JCI-PC	0.197 ± 0.009	0.507 ± 0.024	47.250 ± 9.951
UT-IGSP	0.201 ± 0.020	0.524 ± 0.048	12.680 ± 3.097

F.6 PERFORMANCE ON LINEAR GAUSSIAN DATA

To evaluate the robustness of MetaCaDI against model misspecification, we assess its performance on data generated from linear Gaussian ANMs. Unlike non-linear ANMs, linear Gaussian models lack functional identifiability from observational data alone (Remark 3.1). Consequently, the model cannot exploit functional signatures to determine causal directionality and must rely strictly on interventional signals (i.e., distributional shifts across regimes) to recover the causal structure. Because MetaCaDI parameterizes mechanisms using non-linear neural networks, this setting serves as a challenging test case where the model's inductive biases diverge from the underlying data-generating process.

Under linear Gaussian assumptions, the causal graph is identifiable only up to I-MEC. Therefore, directly computing the SHD between the inferred and true DAGs is inappropriate. Instead, we evaluate structural recovery by converting each DAG into an Interventional Completed Partially Directed Acyclic Graph (I-CPDAG) and computing the expected SHD between these I-CPDAGs to accurately reflect the distance between I-MECs.

Table 5 summarizes the results. Despite lacking functional identifiability guarantees in this setting, MetaCaDI demonstrates robust causal structure recovery, achieving the lowest E-SHD (21.307) and significantly outperforming baseline continuous optimization methods such as BaCaDI (96.100) and DCDI-G (170.550). This confirms that MetaCaDI effectively isolates and exploits intervention signals even in non-ANM settings.

For intervention target identification, the score-based UT-IGSP achieves the highest performance (Intv-AUROC $= 0.628$, Intv-AUPRC $= 0.292$). Nevertheless, MetaCaDI remains highly competitive (Intv-AUROC $= 0.527$, Intv-AUPRC $= 0.233$), ranking second and outperforming the remaining baselines.

Table 5: Performance on Linear Gaussian Data ($d = 20$, averaged over 20 simulations). Structural recovery is measured via the expected SHD between I-CPDAGs to account for I-MEC identifiability limits.

Method	**Intv-AUPRC** $\uparrow$	**Intv-AUROC** $\uparrow$	**E-SHD** $\downarrow$
MetaCaDI-Analytical	0.233 ± 0.046	0.527 ± 0.040	$\mathbf{21.307 \pm 6.839}$
BaCaDI	0.206 ± 0.067	0.510 ± 0.126	96.100 ± 14.828
DCDI-G	0.190 ± 0.000	0.500 ± 0.000	170.550 ± 5.463
JCI-PC	0.196 ± 0.010	0.504 ± 0.008	25.610 ± 4.609
UT-IGSP	$\mathbf{0.292 \pm 0.053}$	$\mathbf{0.628 \pm 0.050}$	24.870 ± 12.453