Amortized Causal Discovery: Learning to Infer Causal Graphs from Time-Series Data

Sindy Löwe*
UvA-Bosch Delta Lab
University of Amsterdam

David Madras*, Richard Zemel
University of Toronto
Vector Institute

Max Welling
UvA-Bosch Delta Lab
University of Amsterdam

Abstract

On time-series data, most causal discovery methods fit a new model whenever they encounter samples from a new underlying causal graph. However, these samples often share relevant information – for instance, the dynamics describing the effects of causal relations – which is lost when following this approach. We propose Amortized Causal Discovery, a novel framework that leverages such shared dynamics to learn to infer causal relations from time-series data. This enables us to train a single, amortized model that infers causal relations across samples with different underlying causal graphs, and thus makes use of the information that is shared. We demonstrate experimentally that this approach, implemented as a variational model, leads to significant improvements in causal discovery performance, and show how it can be extended to perform well under added noise and hidden confounding.

1 Introduction

Inferring causal relations in observational time-series is central to many fields of scientific inquiry [5, 48]. Suppose you want to analyze fMRI data, which measures the activity of different brain regions over time – how can you infer the (causal) influence of one brain region on another? This question is addressed by the field of *causal discovery* [13]. Methods within this field allow us to infer causal relations from observational data – when interventions (e.g. randomized trials) are infeasible, unethical or too expensive.

In time-series, the assumption that causes temporally precede their effects enables us to discover causal relations in observational data [42]; with approaches relying on conditional independence tests [10], scoring functions [6], or deep learning [49]. All of these methods assume that samples share a single underlying causal graph and refit a new model whenever this assumption does not hold. However, samples with different underlying causal graphs may share relevant information such as the dynamics describing the effects of causal relations. fMRI test subjects may have varying brain connectivity but the same underlying neurochemistry; social networks may have differing structure but comparable interpersonal relationships; different stocks may relate differently to one another but obey similar market forces. Despite a range of relevant applications, inferring causal relations across samples with different underlying causal graphs is as of yet largely unexplored.

In this paper, we propose a novel causal discovery framework for time-series that embraces this aspect: Amortized Causal Discovery (Fig. 1). In this framework, we learn to infer causal relations across samples with different underlying causal graphs but shared dynamics. We achieve this by separating the causal relation prediction from the modeling of their dynamics: an amortized encoder predicts the edges in the causal graph, and a decoder models the dynamics of the system under the predicted causal relations. This setup allows us to pool statistical strength across samples and to

*equal contribution; correspondence to: loewe.sindy@gmail.com, madras@cs.toronto.edu

WHY-21 Workshop (Causal Inference and Machine Learning: Why now?), NeurIPS 2021, Virtual.

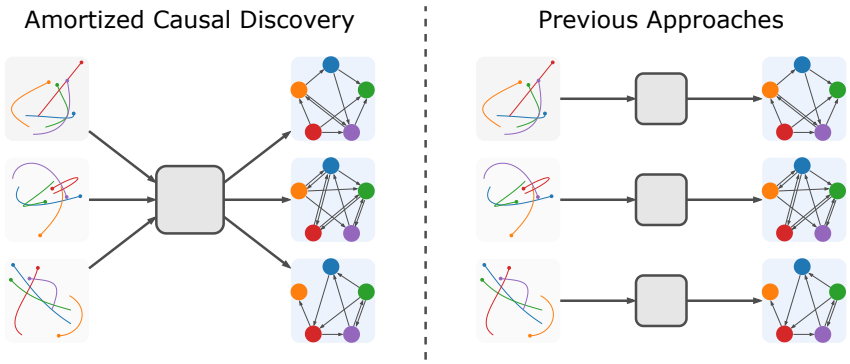

Figure 1: Amortized Causal Discovery. We propose to train a single model that predicts causal relations across samples with different underlying causal graphs but shared dynamics (Eq. (2)). This allows us to generalize across samples and to improve our performance with additional training data. In contrast, previous approaches (Section 2) fit a new model for every sample with a different underlying causal graph.

achieve significant improvements in performance with additional training data. It also allows us to infer causal relations in previously unseen samples without refitting our model. Additionally, we show that Amortized Causal Discovery can improve robustness under hidden confounding by modeling the unobserved variables with the amortized encoder. Our contributions are as follows:

- We formalize Amortized Causal Discovery (ACD), a novel framework for causal discovery in time-series, in which we learn to infer causal relations from samples with different underlying causal graphs but shared dynamics (Eq. (2)).
- We propose a variational model for ACD, applicable to multi-variate, non-linear data.
- We present experiments demonstrating the effectiveness of this model on a range of causal discovery datasets, in the fully observed setting, with added noise, and under hidden confounding.

2 Background: Granger Causality

Granger causality [15] is one of the most commonly used approaches to infer causal relations from observational time-series data. Its central assumption is that causes precede their effects: if the prediction of the future of time-series Y can be improved by knowing past elements of time-series X, then X "Granger causes" Y. Originally, Granger causality was defined for linear relations; we follow the more recent definition of Tank et al. [49] for non-linear Granger causality:

Definition 2.1. *Non-Linear Granger Causality*: Given N stationary time-series $\boldsymbol{x} = \{\boldsymbol{x}_1, ...\boldsymbol{x}_N\}$ across time-steps $t = \{1, ..., T\}$ and a non-linear autoregressive function g_j, such that

$$\boldsymbol{x}_j^{t+1} = g_j(\boldsymbol{x}_1^{\leq t}, ..., \boldsymbol{x}_N^{\leq t}) + \boldsymbol{\varepsilon}_j^t \quad , \tag{1}$$

where $\boldsymbol{x}_j^{\leq t} = (..., \boldsymbol{x}_j^{t-1}, \boldsymbol{x}_j^t)$ denotes the present and past of series j and $\boldsymbol{\varepsilon}_j^t$ represents independent noise. In this setup, time-series i Granger causes j, if g_j depends on $\boldsymbol{x}_i^{\leq t}$, i.e. if $\exists\, \boldsymbol{x}_i'^{\leq t} \neq \boldsymbol{x}_i^{\leq t} : \; g_j(\boldsymbol{x}_1^{\leq t}, ..., \boldsymbol{x}_i'^{\leq t}, ..., \boldsymbol{x}_N^{\leq t}) \neq g_j(\boldsymbol{x}_1^{\leq t}, ..., \boldsymbol{x}_i^{\leq t}, ...\boldsymbol{x}_N^{\leq t})$.

Granger causal relations are equivalent to causal relations in the underlying directed acyclic graph (DAG) if all relevant variables are observed and no instantaneous[1] connections exist [41, 42, Theorem 10.1].

Many methods for Granger causal discovery, including vector autoregressive [21] and more recent deep learning-based approaches [24, 49, 53], can be encapsulated by a particular framework:

1. Define a function f_θ (an MLP in Tank et al. [49], a linear model in Hyvärinen et al. [21]), which learns to predict the next time-step of a given test sequence $\boldsymbol{x}_{\text{test}}$.

[1] connections between two variables at the same time step

2. Fit f_θ to $\boldsymbol{x}_{\text{test}}$ by minimizing some loss $\mathcal{L}$: $\theta_\star = \text{argmin}_\theta\ \mathcal{L}(\boldsymbol{x}_{\text{test}}, f_\theta)$.
3. Apply some fixed function h (e.g. thresholding) to the learned parameters to produce the Granger causal graph estimate for $\boldsymbol{x}_{\text{test}}$: $\hat{\mathcal{G}}_{\boldsymbol{x}_{\text{test}}} = h(\theta_\star)$. For instance, Tank et al. [49] infer the Granger causal relations through examination of the weights $\theta_\star$: if all outgoing weights $\boldsymbol{w}_{ij}$ between time-series i and j are zero, then i does not Granger-cause j.

The shortcoming of this approach is that, when we have S samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_S$ with different underlying causal graphs, the parameters θ must be optimized separately for each of them. As a result, methods within this framework cannot take advantage of the information that might be shared between samples. This motivates us to question: can we amortize this process?

3 Amortized Causal Discovery

We propose Amortized Causal Discovery (ACD), a framework in which we learn to infer causal relations across samples with different underlying causal graphs but shared dynamics. To illustrate: Suppose you want to infer synaptic connections (i.e. causal relations) between neurons based on their spiking behaviour. You are given a set of S recordings (i.e. samples), each containing N time-series representing the firing of N individual neurons. Even though you might record across different populations of neurons with different wiring, the dynamics of how neurons connected by synapses influence one another stays the same. ACD takes advantage of such shared dynamics to improve the prediction of causal relations. Given a training set $\boldsymbol{X}_{\text{train}}$ and test sequence $\boldsymbol{x}_{\text{test}}$, it can be summarized as follows:

1. Define an encoding function f_ϕ which learns to infer Granger causal relations of any sample $\boldsymbol{x}_i$ in $\boldsymbol{X}_{\text{train}}$. Define a decoding function f_θ which learns to predict the next time-step of the samples under the inferred causal relations.
2. Fit f_ϕ and f_θ to $\boldsymbol{X}_{\text{train}}$ by minimizing some loss $\mathcal{L}$: $f_{\phi_\star}, f_{\theta_\star} = \text{argmin}_{f_\phi, f_\theta}\ \mathcal{L}(\boldsymbol{X}_{\text{train}}, f_\phi, f_\theta)$.
3. For test sequence $\boldsymbol{x}_{\text{test}}$, simply output the Granger causal graph estimate $\hat{\mathcal{G}}_{\boldsymbol{x}_{\text{test}}}$: $\hat{\mathcal{G}}_{\boldsymbol{x}_{\text{test}}} = f_{\phi_\star}(\boldsymbol{x}_{\text{test}})$.

By dividing the model into two parts, an encoder and a decoder, ACD can use the *activations* of $f_{\phi_\star}$ to infer causal structure. This increases the flexibility of our approach greatly compared to methods that use the learned *weights* $\theta_\star$ such as the prior Granger causal discovery methods described in Section 2. In this section, we describe our framework in more detail, and provide a probabilistic implementation thereof. We also extend our approach to model hidden confounders.

Preliminaries We begin with a dataset $\boldsymbol{X} = \{\boldsymbol{x}_s\}_{s=1}^S$ of S samples, where each sample $\boldsymbol{x}_s$ consists of N stationary time-series $\boldsymbol{x}_s = \{\boldsymbol{x}_{s,1}, \ldots, \boldsymbol{x}_{s,N}\}$ across timesteps $t = \{1, ..., T\}$. We denote the t-th time-step of the i-th time-series of $\boldsymbol{x}_s$ as $\boldsymbol{x}_{s,i}^t$ (sometimes omitting s for brevity). We assume there is an associated directed acyclic graph $\mathcal{G}_s^{1:T} = \{\mathcal{V}_s^{1:T}, \mathcal{E}_s^{1:T}\}$ underlying the generative process of each sample. This is a structural causal model (SCM) [40]. Its endogenous (observed) variables are vertices $v_{s,i}^t \in \mathcal{V}_s^{1:T}$ for each time-series i and each time-step t. Every set of incoming edges to an endogenous variable defines inputs to a deterministic function $g_{s,i}^t$ which determines that variable's value[2]. The edges are defined by ordered pairs of vertices $\mathcal{E}_s^{1:T} = \{(v_{s,i}^t, v_{s,j}^{t'})\}$, which we make two assumptions about:

1. No edges are instantaneous ($t = t'$) or go back in time. Thus, $t < t'$ for all edges.
2. Edges are invariant to time. Thus, if $(v_{s,i}^t, v_{s,j}^{t+k}) \in \mathcal{E}_s^{1:T}$, then $\forall 1 \leq t' \leq T - k : (v_{s,i}^{t'}, v_{s,j}^{t'+k}) \in \mathcal{E}_s^{1:T}$. The associated structural equations $g_{s,i}^t$ are invariant to time as well, i.e. $g_{s,i}^t = g_{s,i}^{t'}\ \forall t, t'$.

The first assumption states that causes temporally precede their effects and makes causal relations identifiable from observational data, when no hidden confounders are present [41, 42, Theorem 10.1]. The second simplifies modeling: it is a fairly general assumption which allows us to define dynamics that govern all time-steps (Eq. (2)).

[2]The SCM also includes an exogenous (unobserved), independently-sampled error variable ε_v as a parent of each vertex v, which we do not model and thus leave out for brevity.

Throughout this paper, we are interested in discovering the *summary graph* $\mathcal{G}_s = \{\mathcal{V}_s, \mathcal{E}_s\}$ [42]. It consists of vertices $v_{s,i} \in \mathcal{V}_s$ for each time-series i in sample s, and has directed edges whenever they exist in $\mathcal{E}_s^{1:T}$ at any time-step, i.e. $\mathcal{E}_s = \{(v_{s,i}, v_{s,j}) \mid \exists t, t' : (v_{s,i}^t, v_{s,j}^{t'}) \in \mathcal{E}_s^{1:T}\}$. Note that while $\mathcal{G}_s^{1:T}$ is acyclic (due to the first assumption above), the summary graph $\mathcal{G}_s$ may contain (self-)cycles.

Amortized Causal Discovery The key assumption for Amortized Causal Discovery is that there exists some fixed function g that describes the dynamics of *all* samples $\boldsymbol{x}_s \in \boldsymbol{X}$ given their past observations $\boldsymbol{x}_s^{\leq t}$ and their underlying causal graph $\mathcal{G}_s$:

$$\boldsymbol{x}_s^{t+1} = g(\boldsymbol{x}_s^{\leq t}, \mathcal{G}_s) + \boldsymbol{\varepsilon}_s^t \quad . \tag{2}$$

There are two variables in this data-generating process that we would like to model: the causal graph $\mathcal{G}_s$ that is specific to sample $\boldsymbol{x}_s$, and the dynamics g that are shared across all samples. This separation between the causal graph and the dynamics allows us to divide our model accordingly: we introduce an amortized causal discovery encoder f_ϕ which learns to infer a causal graph $\mathcal{G}_s$ given the sample $\boldsymbol{x}_s$, and a dynamics decoder f_θ that learns to approximate g:

$$\boldsymbol{x}_s^{t+1} \approx f_\theta(\boldsymbol{x}_s^{\leq t}, f_\phi(\boldsymbol{x}_s)) \quad . \tag{3}$$

We formalize Amortized Causal Discovery (ACD) as follows. Let $\mathbb{G}$ be the domain of all possible summary graphs on $\boldsymbol{x}_s$: $\mathcal{G}_s \in \mathbb{G}$. Let $\mathbb{X}$ be the domain of any single step, partial or full, observed sequence: $\boldsymbol{x}_s^t, \boldsymbol{x}_s^{\leq t}, \boldsymbol{x}_s \in \mathbb{X}$. The model consists of two components: a causal discovery encoder $f_\phi : \mathbb{X} \to \mathbb{G}$ which infers a causal graph for each input sample, and a decoder $f_\theta : \mathbb{X} \times \mathbb{G} \to \mathbb{X}$ which models the dynamics. This model is optimized with a sample-wise loss $\ell : \mathbb{X} \times \mathbb{X} \to \mathbb{R}$ which scores how well the decoder models the true dynamics of $\boldsymbol{x}_s$, and a regularization term $r : \mathbb{G} \to \mathbb{R}$ on the inferred graphs. For example, this function r may enforce sparsity by penalizing graphs with more edges. Note, that our formulation of the graph prediction problem is unsupervised: we do *not* have access to the true underlying graph $\mathcal{G}_s$. Then, given some dataset $\boldsymbol{X}_{\text{train}}$ with S samples, we optimize:

$$f_{\phi_\star}, f_{\theta_\star} = \text{argmin}_{f_\phi, f_\theta} \, \mathcal{L}(\boldsymbol{X}_{\text{train}}, f_\phi, f_\theta) \tag{4}$$

$$\text{where} \;\; \mathcal{L}(\boldsymbol{X}_{\text{train}}, \phi, \theta) = \sum_{s=1}^{S} \sum_{t=1}^{T-1} \ell(\boldsymbol{x}_s^{t+1}, f_\theta(\boldsymbol{x}_s^{\leq t}, f_\phi(\boldsymbol{x}_s))) + r(f_\phi(\boldsymbol{x}_s)) \quad . \tag{5}$$

Once we have completed optimization, we can perform causal graph prediction on any new input test sample $\boldsymbol{x}_{\text{test}}$ in two ways – we can feed $\boldsymbol{x}_{\text{test}}$ into the amortized encoder and take its output as the predicted edges (Eq. 6); or we can instantiate our estimate $\hat{\mathcal{G}}_{test} \in \mathbb{G}$ which will be our edge predictions, and find the edges which best explain the observed sequence $\boldsymbol{x}_{\text{test}}$ by minimizing the (learned) decoding loss with respect to $\hat{\mathcal{G}}_{test}$, which we term *Test-Time Adaptation (TTA)* (Eq. 7):

$$\hat{\mathcal{G}}^{\text{Enc}} = f_{\phi_\star}(\boldsymbol{x}_{\text{test}}) \quad ; \tag{6}$$

$$\hat{\mathcal{G}}^{\text{TTA}} = \text{argmin}_{\hat{\mathcal{G}}_{test} \in \mathbb{G}} \, \mathcal{L}(\boldsymbol{x}_{\text{test}}, \hat{\mathcal{G}}_{test}, f_{\theta_\star}) \quad . \tag{7}$$

By separating the prediction of causal relations from the modeling of their dynamics, ACD yields a number of benefits. ACD can learn to infer causal relations across samples with different underlying causal graphs, and it can infer causal relations in previously unseen test samples without refitting (Eq. (6)). By generalizing across samples, it can improve causal discovery performance with increasing training data size. We can replace either f_ϕ or f_θ with ground truth annotations, or simulate the outcome of counterfactual causal relations. Additionally, ACD can be applied in the standard causal discovery setting, where only a single causal graph underlies all samples, by replacing the amortized encoder f_ϕ with an estimated graph $\hat{\mathcal{G}}$ (or distribution over $\mathbb{G}$) in Eq. (4).

3.1 A Probabilistic Implementation of ACD

We take a probabilistic approach to ACD and model the functions f_ϕ and f_θ using variational inference (Fig. 2). We amortize the encoder f_ϕ with an encoder function $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, which outputs a distribution over $\boldsymbol{z}$ representing the predicted edges $\hat{\mathcal{E}}_{Enc}$ in the causal graph; and we learn a decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ which probabilistically models the dynamics of the time-series under the predicted causal relations. We choose a negative log-likelihood for the decoder loss ℓ and a KL-Divergence to a prior distribution over $\mathbb{G}$ for the regularizer r. As a result, our loss function $\mathcal{L}$ is a variational lower bound:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \text{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})] \quad . \tag{8}$$

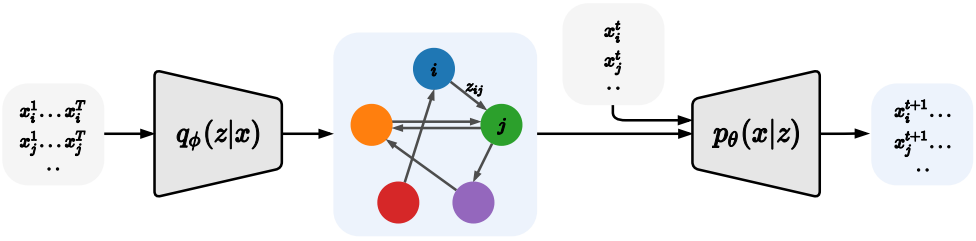

Figure 2: A Probabilistic Implementation of ACD. An amortized encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ predicts the causal relations between the input time-series $\boldsymbol{x}$. A decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ learns to predict the next time-step of the time-series $\boldsymbol{x}^{t+1}$ given their current values $\boldsymbol{x}^t$ and the predicted relations $\boldsymbol{z}$. This separation between causal relation prediction and modeling lets us train the model across samples with different underlying causal graphs but shared dynamics (Eq. (2)).

Encoder The encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ applies a graph neural network $f_{\text{enc},\phi}$ [12, 26, 32, 44] to the input, which propagates information across a fully connected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$. This graph includes vertices $v_i \in \mathcal{V}$ for each time-series i, and each pair of vertices (v_i, v_j) is connected by an edge.

$$\boldsymbol{\psi}_{ij} = f_{\text{enc},\phi}(\boldsymbol{x})_{ij} \tag{9}$$

$$q_\phi(\boldsymbol{z}_{ij}|\boldsymbol{x}) = \texttt{Softmax}\left(\boldsymbol{\psi}_{ij} \,/\, \tau\right) \quad . \tag{10}$$

To enable us to backpropagate through the samples of the discrete distribution $q_\phi(\boldsymbol{z}_{ij}|\boldsymbol{x})$, during training, we relax it by adding Gumbel distributed noise $\boldsymbol{g}$ [22, 36]:

$$\boldsymbol{z}_{ij} \sim \texttt{Softmax}\left(\left(\boldsymbol{\psi}_{ij} + \boldsymbol{g}\right) / \tau\right) \quad . \tag{11}$$

The output $\boldsymbol{z}_{ij}$ of the encoder represents the predicted edges $\hat{\mathcal{E}}_{\text{Enc}}$ in the causal graph $\hat{\mathcal{G}}_{\text{Enc}}$. We consider the possibility that there are $n_{\mathcal{E}}$ different edge-types expressing causal relationships; for instance, inhibitory or excitatory synaptic connections. Then, more specifically, $z_{ij,e} = 1$ expresses that there is a directed edge of type e from time-series i to j, where $e \in \{1, \dots, n_{\mathcal{E}}\}$.

Decoder The decoder $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ models the dynamics of the time-series under the predicted causal relations. It uses both the predicted causal relations $\boldsymbol{z}_{ij}$ and the feature vectors of the time-series at the current time-step t, $\boldsymbol{x}^t = \{\boldsymbol{x}_1^t, \dots \boldsymbol{x}_N^t\}$ as its input. First, it propagates information along the predicted edges by applying a neural network f_e, using the zero function for f_0:

$$\boldsymbol{h}_{ij}^t = \sum_{e>0} z_{ij,e} f_e([\boldsymbol{x}_i^t, \boldsymbol{x}_j^t]) \quad . \tag{12}$$

Then, the decoder accumulates the incoming messages to each node and applies a neural network f_v to predict the change between the current and the next time-step:

$$\boldsymbol{\mu}_j^{t+1} = \boldsymbol{x}_j^t + f_v\left(\left[\sum_{i \neq j} \boldsymbol{h}_{ij}^t, \boldsymbol{x}_j^t\right]\right) \tag{13}$$

$$p_\theta(\boldsymbol{x}_j^{t+1}|\boldsymbol{x}^t, \boldsymbol{z}) = \mathcal{N}(\boldsymbol{\mu}_j^{t+1}, \sigma^2 \mathbb{I}) \quad . \tag{14}$$

In other words, the decoder predicts $\Delta\hat{\boldsymbol{x}}^t$, which is added to the current value of the time-series to yield the prediction for the next time-step $\hat{\boldsymbol{x}}^{t+1} = \boldsymbol{x}^t + \Delta\hat{\boldsymbol{x}}^t$.

Prediction of Causal Relations In order to align our model with the philosophy of Granger Causality, we include a "no edge"-type edge function: If the encoder predicts the "no edge"-type edge $e = 0$ by setting $z_{ij,0} = 1$, the decoder uses the zero function and no information is propagated from time-series i to j (Eq. (12)). Due to this, time-series i will Granger cause the decoder-predicted time-series j only when the edge is predicted to exist (see Appendix A). Hence, by the same logic that justifies prior Granger causal work (Section 2), we expect the predicted edges to correspond to Granger causal relations. Finally, since we assume no hidden confounders and no instantaneous edges, these Granger causal relations will correspond to relations in the underlying SCM [42, Theorem 10.3].

3.2 Hidden Confounding

Hidden confounders are a critical problem in the time-series context: when they exist, Granger causality is not guaranteed to correspond to the true causal graph anymore [42, Theorem 10.3][3]. Inspired by proxy-based methods from causal inference (e.g. Louizos et al. [35], see Section 4), we present a method for applying ACD to the hidden confounding setting. First, we extend the amortized encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ to predict an additional variable. Then, we encourage this variable to model the hidden confounder by applying a structural bias – depending on the type of unobserved variable that we want to model, its predicted value is utilized differently by the remaining model. The decoder remains responsible for modeling the dynamics, and now also processes the predictions for the unobserved variable. While this setup might not allow us to identify the hidden confounders, and it is still true that the predicted Granger causal relations may not correspond to the underlying SCM, the data-driven approach underlying ACD can benefit our model: by training across samples with different underlying causal graphs, our model has access to substantially more information about the causal dynamics at hand and we show empirically that it can learn to mitigate the effects of the hidden confounders.

We consider two types of hidden confounders which were chosen to cover a wide range of potential confounders as one might encounter in practice. First, we introduce a temperature variable that confounds all observed variables by influencing the strength of their causal relations. This temperature is sampled separately for each sample, and remains constant throughout each sample. This is a realistic example of a global variable that may influence physical dynamics. Second, we introduce a hidden variable that behaves just like the observed variables, i.e. it may affect or be affected by the observed variables through the same causal relations, and its value changes across time. This confounder is inspired by the introductory example where one wants to infer the causal relations between neurons based on their spiking pattern. In this scenario, it is virtually impossible to record all relevant neurons. The resulting unobserved neurons behave and influence the observed neurons in a similar fashion as the unobserved time-series confounder introduced here. In both scenarios, we extend the encoder to predict this hidden variable, and feed that prediction into the decoder. We provide more details in Section 5.2.

4 Related Work

A range of approaches to causal discovery in both temporal and non-temporal data exist [16, 42, 48]. One common class is *constraint-based*, relying on conditional independence testing to uncover an underlying DAG structure or equivalence class [48]. These methods predict a single graph $\hat{\mathcal{G}}$ (or equivalence class) for all samples. There is no notion of fitting a dynamics model for time-series methods in this class [10]. Another common class of methods for causal discovery is *score-based* [4, 6]. Here, a score function h is chosen, and the methods perform a search through graph space to optimize this score, i.e. $\hat{\mathcal{G}} = \text{argmin}_{\mathcal{G}}\ h(\mathcal{G})$. Our proposed decoder-based inference (Eq. (7)) can be seen as score-based causal discovery with a *learned* score function $\mathcal{L} \circ f_{\theta_\star}$. A third class of methods fits a (possibly regularized) dynamics model f and then analyzes its form to produce a causal graph estimate, by using linear dynamics [21], recurrent models [24, 37, 49], or other deep-learning based approaches [30, 53, 55]. See Section 2 for discussion. In concurrent work, Zečević et al. [54] discuss the relationship between GNNs and causal effect inference. Other approaches to causal discovery in temporal data use independence or additivity assumptions [9, 41]. Many works have explored the idea of jointly learned causal structure across examples, including in the setting where a number of related datasets are collected [8, 18–20, 46, 50]. Li et al. [31] proposes learning a linear mixed effects model across samples, and concurrent work explores amortized deep learning of differing types of causal structure [23, 33]. There is little systematic study of hidden confounding in the time-series setting. Some empirical work in the non-temporal domain shows that encoder-based models with enough proxies (variables caused by hidden confounders) can improve causal inference under hidden confounding [35, 38], and theoretical work proves the identifiability of latent variables from proxies under some assumptions [2, 28].

Several works have used graph neural networks [3, 27, 43] or attention mechanisms [11, 14, 51, 52] to infer relations between time-series. Alet et al. [1] propose a meta-learning algorithm to additionally

[3] For instance, if an unobserved time-series U causes both time-series X and Y, then the past of X can help predict the future of Y, even though there is no causal link between them.

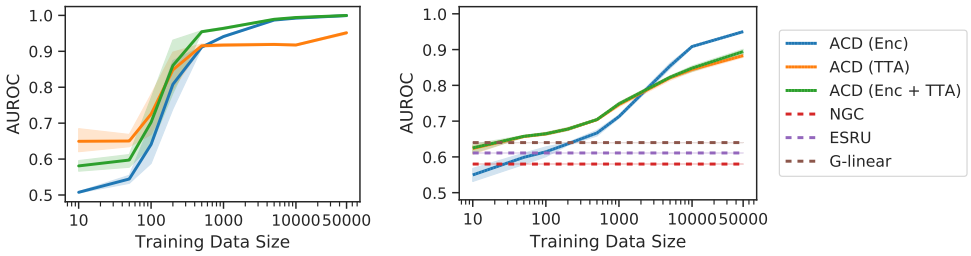

Figure 3: Causal discovery performance (in AUROC) on the particles dataset (A-left) and Kuramoto (B-right). ACD improves with more training data, outperforming previous approaches with as few as 50 available training samples on Kuramoto. In the high-data regime, encoder inference (*Enc*) is best, while test-time adaptation (*TTA* and *Enc+TTA*) is superior in low-data settings.

model unobserved variables. While these approaches model object relations in a number of ways, they are not explicitly designed to infer *causal* graphical structure.

The probabilistic implementation of ACD is based on the Neural Relational Inference (NRI) model [27]. We extend this work by new inference methods using test-time adaptation and new algorithms to handle hidden confounders. Moreover, we provide a proof that relates the zero-edge function to Granger causality which allows for a causal interpretation of the inferred edges. Subsequently, we apply our model to a different problem than NRI, namely (Granger) causal discovery, and show that it outperforms the current state of the art for this type of problem. Last but not least, we show that our model achieves strong causal discovery performance even under noise and hidden confounding, an accomplishment that is – to the best of our knowledge – new in this field.

5 Experiments

Implementation We measure causal discovery performance by area under the receiver operator curve (AUROC) of predicted edge probabilities over test samples. We compare to recurrent models (Khanna and Tan [24], Tank et al. [49]), a mutual-information (MI) based model by Wu et al. [53] and several baselines implemented by those authors, including MI (unmodified), transfer entropy [45], and linear Granger causality. More details in Appendix B, our code is available at `github.com/loeweX/AmortizedCausalDiscovery`.

5.1 Fully Observed Amortized Causal Discovery

We test ACD on three datasets: two fully-observed physics simulations (Kuramoto and Particles) and the Netsim dataset of simulated fMRI data [47]. Note, in contrast to the physics simulations used in Kipf et al. [27], we generate data with *asymmetric* connectivity matrices to represent causal relations. We show additional results for noisy versions of Kuramoto and Particles in Appendix B.2.2.

First, we test our method on the **Kuramoto dataset**, which contains five 1-D time-series of phase-coupled oscillators [29]. We find that ACD greatly outperforms all approaches for Granger causal discovery that we compare against (Table 1). In contrast to these approaches, ACD achieves this result *without* fitting to the test samples. Additionally, we find that ACD can indeed utilize samples with different underlying causal graphs – its performance improves steadily with increasing training data size (Fig. 3). Nonetheless, it is also applicable to the low-data regime: when applying ACD with test-time adaptation (TTA), it requires less than 50 training samples to outperform all previous approaches. We note that the baseline performance here is worse than presented elsewhere in the literature – this is because we do not evaluate the prediction of self-connectivity, which is the easiest connectivity to predict.

Method	AUROC
MPIR [53]	0.502 ± 0.006
Transfer Entropy [45]	0.560 ± 0.005
NGC [49]	0.574 ± 0.018
eSRU [24]	0.607 ± 0.001
Mutual Information	0.616 ± 0.000
Linear Granger Causality	0.647 ± 0.003
Amortized Causal Discovery	$\mathbf{0.952 \pm 0.003}$

Table 1: AUROC for causal discovery on Kuramoto dataset. 95% confidence interval shown.

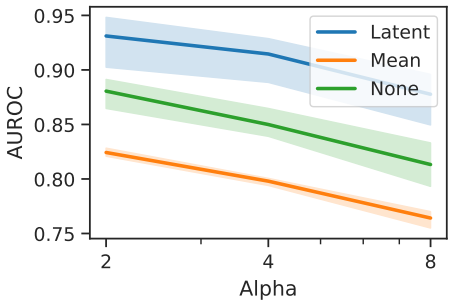

Figure 4: AUROC with unobserved temperature. ACD with a *latent* variable outperforms a baseline which imputes a *mean* temperature, and a learned fixed-temperature decoder (*None*).

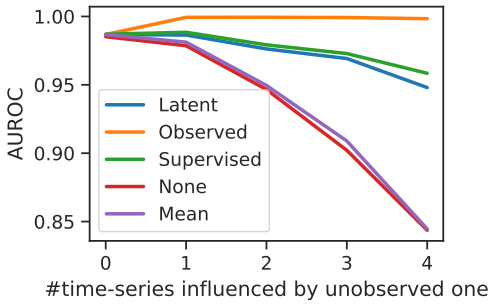

Figure 5: AUROC with unobserved time-series. As more time-series are influenced by the unobserved one (x-axis), the benefit of using an additional *latent* variable for modeling its effects grows.

In our second experiment, we apply ACD to the **particles dataset**. This dataset models five particles that move around a two-dimensional space, with some particles influencing others uni-directionally by pulling them with a spring. Since all previous methods were intended for one-dimensional time-series, we were unable to evaluate them in this domain. ACD, on the other hand, is readily applicable to higher-dimensional data, and performs almost perfectly on this dataset with **0.999 AUROC**.

In both experiments, causal relation prediction with the learned encoder (Enc - Eq. (6)) performs best in the high-data regime, while test-time adaptation (TTA - Eq. (7)) improves the performance in low-data settings (Fig. 3). This benefit of TTA can be largely attributed to two effects. First, TTA closes the amortization gap of the encoder [7]. Second, TTA overcomes the encoder's overfitting on the training data (as seen in the training curves in Appendix B.2.1) by adapting to the individual test samples. On the particles dataset, initializing TTA with the encoder's prediction (Enc+TTA) improves over a random initialization (TTA) as the encoder improves; but we do not observe this effect on the Kuramoto dataset.

Finally, we apply ACD to the **Netsim** dataset [47] of simulated fMRI data. Here, the task is to infer the underlying connectivity between 15 brain regions across 50 samples. A single graph underlies all samples, allowing us to demonstrate ACD's applicability to the classical setting. We replace the amortized encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with a global latent distribution $q(\boldsymbol{z})$, optimize it through the decoder, and then use test-time adaptation (TTA). Even though our model cannot benefit from its data-driven design here, it performs comparably to methods that are intended for use in the single-graph setting (Table 2).

Method	AUROC
MPIR [53]	0.484 ± 0.017
Transfer Entropy [45]	0.543 ± 0.003
NGC [49]	0.624 ± 0.020
eSRU [24]	0.670 ± 0.015
Mutual Information	$\mathbf{0.728 \pm 0.002}$
Linear Granger Causality	0.503 ± 0.004
Amortized Causal Discovery	$\mathbf{0.688 \pm 0.051}$

Table 2: AUROC for causal discovery on Netsim dataset. 95% confidence interval shown.

5.2 Amortized Causal Discovery under Hidden Confounding

5.2.1 Latent Temperature

In this experiment, we use the particles dataset and vary an unobserved temperature variable, which modulates how strongly the particles exert force on each other – higher temperatures result in stronger forces and a more chaotic system. For each $\boldsymbol{x}_s$, we sample an independent temperature $c \sim \text{Categorical}([\frac{\alpha}{2}, \alpha, 2\alpha])$ from a categorical distribution with $\alpha \in \mathbb{R}$ and equal probabilities. We predict this unobserved temperature by extending the amortized encoder with an additional latent variable which models a uniform distribution. Then, we add a KL-Divergence between this posterior and a uniform prior on the interval $[0, 4\alpha]$ to our variational loss. To allow for learning in this setting, we introduce an inductive bias: we use a decoder which matches the true dynamics g given the predicted temperature and causal relations. See Appendix C.1 for more details and additional results.

Results Fig. 4 shows the causal discovery results across different values of α. ACD enhanced with an additional latent variable (*Latent*) outperforms both tested baselines across all temperatures: *Mean*, which uses the same ground-truth decoder as *Latent* and fixes the decoder temperature to be the mean of the categorical distribution, and *None*, which does not model c explicitly and trains an MLP decoder. Additionally, this method achieves high predictive performance on the unobserved temperature variable: for $\alpha = 2$, temperature prediction obtains 0.888 R^2, 0.966 AUROC and 0.644 accuracy. These results indicate that we can model an unobserved temperature variable, and thus improve robustness under hidden confounding.

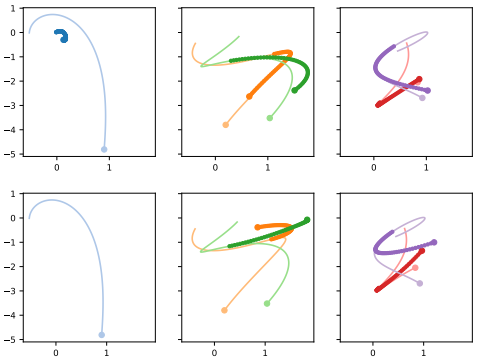

Figure 6: Trajectory prediction with an unobserved time-series (TS). Faded: ground truth. Bold: prediction, starts after observing the first half of the ground truth. Dots denote end of TS. Top: ACD with *Latent*, bottom: *None* baseline - does not model unobserved TS. Left: unobserved TS, middle: TS directly influenced by unobserved, right: remaining TS. Though we underestimate the unobserved TS, observed TS prediction improves.

5.2.2 Unobserved Time-Series

Here, we treat one of the original time-series in the particles dataset as unobserved. It exhibits the same dynamics as the observed time-series, evolving and causally influencing others the same way as before. This challenging setting has received little attention in the literature; Alet et al. [1] tackled it with mixed success. We model the unobserved time-series by extending the amortized encoder with an additional latent variable and applying a suitable structural bias: the latent prediction $\boldsymbol{z}_u^t$ for time-steps $t = \{1, ..., T\}$ is treated in the same way as the observed time-series $\boldsymbol{x}$. Its entire trajectory is used by the encoder to predict causal relations, and its value at the current time-step is fed into the decoder. See Appendix C.2 for more details and additional results.

Results Fig. 5 shows how the causal discovery AUROC depends on the number of observed time-series directly influenced by the unobserved one. When this number is zero, all tested approaches perform the same. With growing numbers of influenced time-series, the baselines that either ignore the missing time-series (*None*) or impute its value with the average of the observed time-series over time (*Mean*) deteriorate strongly. In contrast, the proposed ACD with a *Latent* variable stays closer to the performance of the fully *Observed* baseline. As shown in Fig. 6, it also improves the future trajectory prediction of the observed time-series. A *Supervised* baseline that uses the (usually unavailable) ground-truth trajectories to optimize the prediction of the unobserved time-series, improves only slightly over our approach. These results indicate ACD can use latent variables to improve robustness to unobserved time-series.

6 Conclusion

We introduce ACD, a framework for causal discovery in time-series data which can leverage the information that is shared across samples. We provide a probabilistic implementation of this framework, and demonstrate significant performance gains over the existing literature when predicting causal relations, both in the fully observed setting and with noise and hidden confounding. Despite this improvement in performance over previous work, several limitations remain. Our assumptions from Sec. 3 regarding shared dynamics and edges in the graph, are not verifiable in practice; however, this is standard in causal inference, which frequently relies on untestable assumptions (e.g. ignorability, consistency).We conducted all experiments on simulated data; as a result, they are not so realistic: real-world data is more complex and potentially mis-specified. This limitation is in line with the data used in related works (e.g. Khanna and Tan [24]), and it remains an exciting direction for future work to improve modelling and experimentation in more realistic settings. Finally, our contribution is primarily empirical: we propose and attack a novel version of the causal discovery problem – where samples have different underlying causal graphs but shared dynamics – and demonstrate that ACD outperforms prior methods by a large margin in this setting. It remains an important future direction to understand the conditions under which ACD, as well as other approaches that inspired ACD (e.g. Khanna and Tan [24], Tank et al. [49]), can be guaranteed to identify the correct causal structures.

Acknowledgments

Thanks to Thomas Kipf for helpful discussions and to Marco Federici, Gabriele Bani, Joop Pascha, Patrick Forre, Pascal Esser, Maja Rudolph, Sara Magliacane, Elliot Creager, Taylor Killian, and Renjie Liao for their feedback on the manuscript. David Madras was supported by an NSERC Alexander Graham Bell Canada Graduate Scholarship-Doctoral (CGS-D). Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute (`www.vectorinstitute.ai/#partners`).

References

[1] F. Alet, E. Weng, T. Lozano-Pérez, and L. P. Kaelbling. Neural relational inference with fast modular meta-learning. In *Advances in Neural Information Processing Systems*, pages 11804–11815, 2019.

[2] E. S. Allman, C. Matias, J. A. Rhodes, et al. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A):3099–3132, 2009.

[3] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems*, pages 4502–4510, 2016.

[4] Y. Bengio, T. Deleu, N. Rahaman, N. R. Ke, S. Lachapelle, O. Bilaniuk, A. Goyal, and C. Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2019.

[5] C. Berzuini, P. Dawid, and L. Bernardinell. *Causality: Statistical perspectives and applications*. John Wiley & Sons, 2012.

[6] D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002.

[7] C. Cremer, X. Li, and D. Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pages 1078–1086, 2018.

[8] A. Dhir and C. M. Lee. Integrating overlapping datasets using bivariate causal discovery. In *AAAI*, pages 3781–3790, 2020.

[9] M. Eichler. *Causal inference in time series analysis*. Wiley Online Library, 2012.

[10] D. Entner and P. O. Hoyer. On causal discovery from time series data using FCI. *Probabilistic Graphical Models*, pages 121–128, 2010.

[11] F. B. Fuchs, A. R. Kosiorek, L. Sun, O. P. Jones, and I. Posner. End-to-end recurrent multi-object tracking and trajectory prediction with relational reasoning. *arXiv preprint arXiv:1907.12887*, 2019.

[12] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pages 1263–1272, 2017.

[13] C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019.

[14] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*, 2019.

[15] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[16] C. Heinze-Deml, M. H. Maathuis, and N. Meinshausen. Causal structure learning. *Annual Review of Statistics and Its Application*, 5:371–391, 2018.

[17] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[18] B. Huang, K. Zhang, P. Xie, M. Gong, E. P. Xing, and C. Glymour. Specific and shared causal relation modeling and mechanism-based clustering. In *Advances in Neural Information Processing Systems*, pages 13510–13521, 2019.

[19] B. Huang, K. Zhang, M. Gong, and C. Glymour. Causal discovery from multiple data sets with non-identical variable sets. In *AAAI*, pages 10153–10161, 2020.

[20] B. Huang, K. Zhang, J. Zhang, J. D. Ramsey, R. Sanchez-Romero, C. Glymour, and B. Schölkopf. Causal discovery from heterogeneous/nonstationary data. *J. Mach. Learn. Res.*, 21(89):1–53, 2020.

[21] A. Hyvärinen, K. Zhang, S. Shimizu, and P. O. Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(56): 1709–1731, 2010.

[22] E. Jang, S. Gu, and B. Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations*, 2017.

[23] N. R. Ke, J. Wang, J. Mitrovic, M. Szummer, D. J. Rezende, et al. Amortized learning of neural causal representations. *arXiv preprint arXiv:2008.09301*, 2020.

[24] S. Khanna and V. Y. Tan. Economy statistical recurrent units for inferring nonlinear granger causality. *International Conference on Learning Representations*, 2019.

[25] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[26] T. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

[27] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697, 2018.

[28] J. B. Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, 18(2): 95–138, 1977.

[29] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics*, pages 420–422. Springer, 1975.

[30] S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien. Gradient-based neural dag learning. In *International Conference on Learning Representations*, 2019.

[31] X. Li, S. Xie, P. McColgan, S. J. Tabrizi, R. I. Scahill, D. Zeng, and Y. Wang. Learning subject-specific directed acyclic graphs with mixed effects structural equation models from observational data. *Frontiers in genetics*, 9:430, 2018.

[32] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel. Gated graph sequence neural networks. In *International Conference on Learning Representations*, 2016.

[33] Y. Li, A. Torralba, A. Anandkumar, D. Fox, and A. Garg. Causal discovery in physical systems from videos. *arXiv preprint arXiv:2007.00631*, 2020.

[34] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio. A structured self-attentive sentence embedding. In *International Conference on Learning Representations*, 2017.

[35] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pages 6446–6456, 2017.

[36] C. J. Maddison, A. Mnih, and Y. W. Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[37] M. Nauta, D. Bucur, and C. Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):312–340, 2019.

[38] S. Parbhoo, M. Wieser, A. Wieczorek, and V. Roth. Information bottleneck for estimating treatment effects with systematically missing covariates. *Entropy*, 22(4):389, 2020.

[39] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[40] J. Pearl. *Causality*. Cambridge University Press, 2009.

[41] J. Peters, D. Janzing, and B. Schölkopf. Causal inference on time series using restricted structural equation models. In *Advances in Neural Information Processing Systems*, pages 154–162, 2013.

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

[43] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, pages 4967–4976, 2017.

[44] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

[45] T. Schreiber. Measuring information transfer. *Physical Review Letters*, 85(2):461, 2000.

[46] S. Shimizu. Joint estimation of linear non-gaussian acyclic models. *Neurocomputing*, 81:104–107, 2012.

[47] S. M. Smith, K. L. Miller, G. Salimi-Khorshidi, M. Webster, C. F. Beckmann, T. E. Nichols, J. D. Ramsey, and M. W. Woolrich. Network modelling methods for fMRI. *Neuroimage*, 54(2): 875–891, 2011.

[48] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, prediction, and search*. MIT Press, 2000.

[49] A. Tank, I. Covert, N. Foti, A. Shojaie, and E. Fox. Neural granger causality for nonlinear time series. *arXiv preprint arXiv:1802.05842*, 2018.

[50] R. E. Tillman and F. Eberhardt. Learning causal structure from multiple datasets with similar variable sets. *Behaviormetrika*, 41(1):41–64, 2014.

[51] S. Van Steenkiste, M. Chang, K. Greff, and J. Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[53] T. Wu, T. Breuel, M. Skuhersky, and J. Kautz. Discovering nonlinear relations with minimum predictive information regularization. *arXiv preprint arXiv:2001.01885*, 2020.

[54] M. Zečević, D. S. Dhami, P. Veličković, and K. Kersting. Relating graph neural networks to structural causal models. *arXiv preprint arXiv:2109.04173*, 2021.

[55] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. Xing. Learning sparse nonparametric dags. In *International Conference on Artificial Intelligence and Statistics*, pages 3414–3425. PMLR, 2020.

A Granger Causality of ACD

Here, we show that when constraining the edge-type $e = 0$ to be the zero function, time series i does not Granger cause the model prediction of j in ACD. In the noiseless setting, at the global optimum of sufficiently expressive model classes, this is equivalent to saying that we recover the true Granger causal relations.

Claim. If $z_{ij,0} = 1$, i does not Granger cause the model prediction of j in ACD.

Proof. According to Definition 2.1, time-series i does not cause j, if g_j is invariant to $\boldsymbol{x}_i^{\leq t}$. In our model, the decoder represents this non-linear model g_j and consists of two functions. First, it propagates information across edges using Eq. (12). This function returns a value of zero, if $z_{ij,0} = 1$. This output is used for the second function, described in Eq. (1), which does not introduce any new terms that depend on i. Thus, if $z_{ij,0} = 1$, the decoder's prediction for j is invariant to $\boldsymbol{x}_i^{\leq t}$, and i does not Granger cause these predictions.

B Fully Observed Amortized Causal Discovery

B.1 Experimental Details

B.1.1 Datasets

Physics Simulations To generate these simulations, we follow the description of the underlying physics of Kipf et al. [27] for the phase-coupled oscillators (Kuramoto) [29] and the particles connected by springs. In contrast to their simulations, however, we allow the connectivity matrix, which describes which time-series influences another, to be asymmetric. This way, it describes causal relations instead of correlations.

For both datasets, we generate 50,000 training and 10,000 validation samples. We restrict the number of test samples to 200, since the previous methods we compare to must be refit for each individual sample. We simulate systems with $N = 5$ time-series. Our training and validation samples consist of $T = 49$ time-steps, while the test-samples are $T = 99$ time-steps long. This increased length allows us to infer causal relations on the first half of the data, and to test the future prediction performance on the second half (with $k = \{1, ..., 49\}$).

Netsim The Netsim dataset simulates blood-oxygen-level-dependent (BOLD) imaging data across different regions within the human brain and is described in Smith et al. [47]. The task is to infer the directed connections, i.e. causal relations, between different brain areas.

The Netsim dataset includes simulations with different numbers of brain regions and different underlying connectivity matrices. In our experiments, we use the data from the third simulation Sim-3.mat as provided by Khanna and Tan [24]. It consists of samples from 50 subjects, each with the same underlying causal graph, each of length $T = 200$ and including $N = 15$ different brain regions. Note, that we report worse results than Khanna and Tan [24], since we assume self-connectivity for all time-series and only evaluate the causal discovery performance between *different* time-series.

The dataset is very small (50 samples) and due to this, we do not use a training/validation/test split, but use the same 50 points at each phase instead. While this is not standard machine learning practice, it still facilitates a fair comparison to the other methods, each of which are fit to individual test points. The purpose of including experiments on this dataset is not to demonstrate generalization ability, but rather to show that our method is flexible enough to work reasonably well even in the classical causal discovery setting (with one shared causal graph, and fitting the model on the test set).

B.1.2 Architecture and Hyperparameters

Our model is implemented in PyTorch [39]. We did no hyperparameter optimization for model training, but used the settings as described for the NRI model [27]. The latent dimension throughout the model is set to size 256. We optimize our model using ADAM [25] with a learning rate of 0.0005. In the experiments on the particles dataset, the learning rate is decayed by a factor of 0.5 every 200

epochs. We set our batchsize to 128 and train for 500 epochs. The temperature of the Gumbel-Softmax is set to $\tau = 0.5$. During testing, this concrete distribution is replaced by a categorical distribution to obtain discrete edge predictions.

There was no thorough hyperparameter optimization done for test-time adaptation (TTA). Since there was no pre-existing implementation, some hand-tuning was performed. We use a learning rate of 0.1 for the Kuramoto and particles datasets and 0.01 for Netsim. For each, we run 1000 iterations.

Encoder In our experiments, the amortized encoder applies a graph neural network $f_{enc,\phi}$ on the input. It implements two edge-propagation steps along the causal graph:

$$\boldsymbol{\psi}_j^1 = f_{\text{emb}}(\boldsymbol{x}_j) \tag{15}$$

$$\boldsymbol{\psi}_{ij}^1 = f_e^1([\boldsymbol{\psi}_i^1, \boldsymbol{\psi}_j^1]) \tag{16}$$

$$\boldsymbol{\psi}_j^2 = f_v^2(\sum_{i \neq j} \boldsymbol{\psi}_{ij}^1) \tag{17}$$

$$\boldsymbol{\psi}_{ij} = f_e^2([\boldsymbol{\psi}_i^2, \boldsymbol{\psi}_j^2]) \quad . \tag{18}$$

f_e^1, f_e^2 and f_v^2 are fully-connected networks (MLPs). On both the particles dataset and Netsim, f_{emb} is an MLP as well (MLP Encoder); on the Kuramoto dataset, we use a 1D CNN with attentive pooling [34] instead (CNN Encoder).

When conducting test-time adaptation as described in Eq. (7), we remove the encoder and model a distribution over $\mathbb{G}$ using a non-amortized variational distribution $q(\boldsymbol{z})$ with its initial values sampled from a unit Gaussian.

Decoder The decoder implements a single edge-propagation step according to equations 12-14. It uses MLPs for both f_e and f_v. To improve performance, we train the decoder to predict several time-steps into the future. For this, we replace the true input $\boldsymbol{x}^t$ with the predicted $\boldsymbol{\mu}^t$ for $k = 10$ steps.

Following our causal formulation of the NRI model, we implement Eq. (12) by masking out the values of the corresponding edges. Thus, the ordering of the edge types is not arbitrary in our setting.

We note that while this implementation of the decoder assumes a full-time graph of Markov order 1, the full ACD framework does not, and could be implemented using a recurrent architecture to remove this assumption.

Since out physics simulations are differentiable, we can replace the decoder with the ground-truth dynamics and backpropagate through them. We call this setup the simulation decoder.

Variance When we report the variance on the ACD results, we collected these across five different random seeds. Baselines in Kuramoto/Netsim use three seeds each, except for NGC, which uses only one due to a longer runtime (the confidence intervals shown for NGC are confidence intervals on the AUROC itself, whereas all other confidence intervals are based on variance of AUROC across seeds).

B.1.3 Baselines

We compare ACD against several baselines:

Neural Granger Causality From Tank et al. [49], we optimized an MLP or LSTM to do next step prediction on a sample. We found that the MLP worked best. The causal links are wherever an input weight is non-zero. We used ADAM and then line search to find exact zeros. In this method, we calculate AUROC by running with a range of sparsity hyperparameters ($\lambda = [0, 0.1, 0.2, 0.4, 0.8]$ for Kuramoto and $\lambda = [0, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 1]$ for Netsim). As in Tank et al. [49], we calculate a score s for each edge, where $s = \min\{\lambda : z_{ij,0} = 1\}$, and use that score to calculate AUROC. Code was used from `https://github.com/icc2115/Neural-GC`.

ESRU Khanna and Tan [24] take a similar approach to Tank et al. [49], but they use economy statistical recurrent units (eSRU), instead of LSTMs. We found one layer worked best, and used their hyperparameters otherwise. We use sparsity hyperparameters $[0.1, 0.2, 0.3, 0.4, 0.5]$ for Kuramoto, and $[0, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 1]$ for Netsim. Code was used from `https://github.com/sakhanna/SRU_for_GCI`.

MPIR Wu et al. [53] determine where causal links exist by examining the predictive performance change when noise is added on an input variable. Code for this method and the baselines below was used from `https://github.com/tailintalent/causal`.

Transfer Entropy Schreiber [45] suggest this entropy-like measure between two variables to produce a metric which is likely to be higher when a causal connection exists. We use the implementation by Wu et al. [53].

Mutual Information Using the implementation by Wu et al. [53], we calculate the mutual information between every pair of time series.

Linear Granger Causality Using the implementation by Wu et al. [53], this is a linear version of Granger causality where non-zero linear weights are taken as greater causal importance.

We did not run the baselines on the particles dataset since it is two-dimensional and most baselines did not provide an obvious way for handling multi-dimensional time series. When training ACD on the particles and Kuramoto datasets, we additionally input the velocity (and phase for Kuramoto) of the time-series. Since our chosen NRI encoders and decoders are not recurrent we cannot recover this information in any other way in this model. This enables a more fair comparison to the recurrent methods, which are able to aggregate this information over several time steps.

B.2 Additional Experimental Results

B.2.1 Training Curves

Fig. 7 shows the training curves when training on 100 training samples of the particles dataset. We observe that the encoder overfits on the training samples, as indicated by the AUROC performance. In contrast, the decoder shows less overfitting as indicated by the negative log-likelihood (NLL) performance.

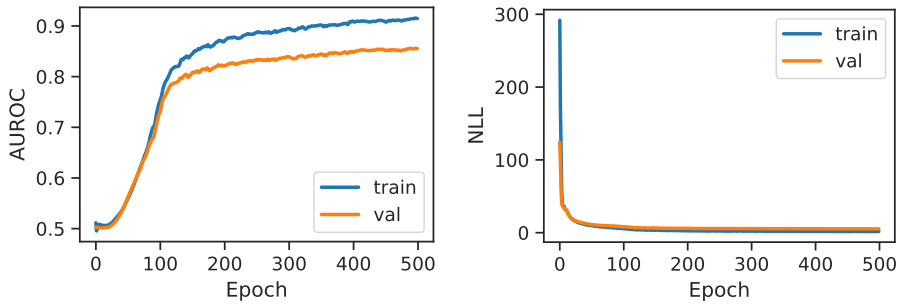

Figure 7: Training curves when training on 100 samples of the particles dataset. The encoder performance (AUROC - left) shows stronger signs of overfitting than the decoder performance (NLL - right).

B.2.2 Noisy Data

Handling noisy data is a key challenge in causal discovery. Here, we show that ACD is robust to a certain amount of observational noise in both the Particles and Kuramoto tasks.

We test ACD's performance for different levels of observation noise added to the Particles and Kuramoto datasets (Fig. 8). We sample this additive noise from a zero-mean normal distribution with standard deviations varying between 0.0 and 0.2. For comparison, the input samples in both the Particles and Kuramoto dataset have a standard deviation of 0.6. For each dataset and each standard deviation, we train across five random seeds. In Fig. 9, we show the average AUROC on a test set with the same noise scheme applied. ACD's performance degrades gracefully when stronger observation noise is added. In the highest noise scenario, when the noise is sampled from a normal distribution with a standard deviation of 0.2, the performance of ACD matches the performance of the baselines from Table 1 on *noiseless* data.

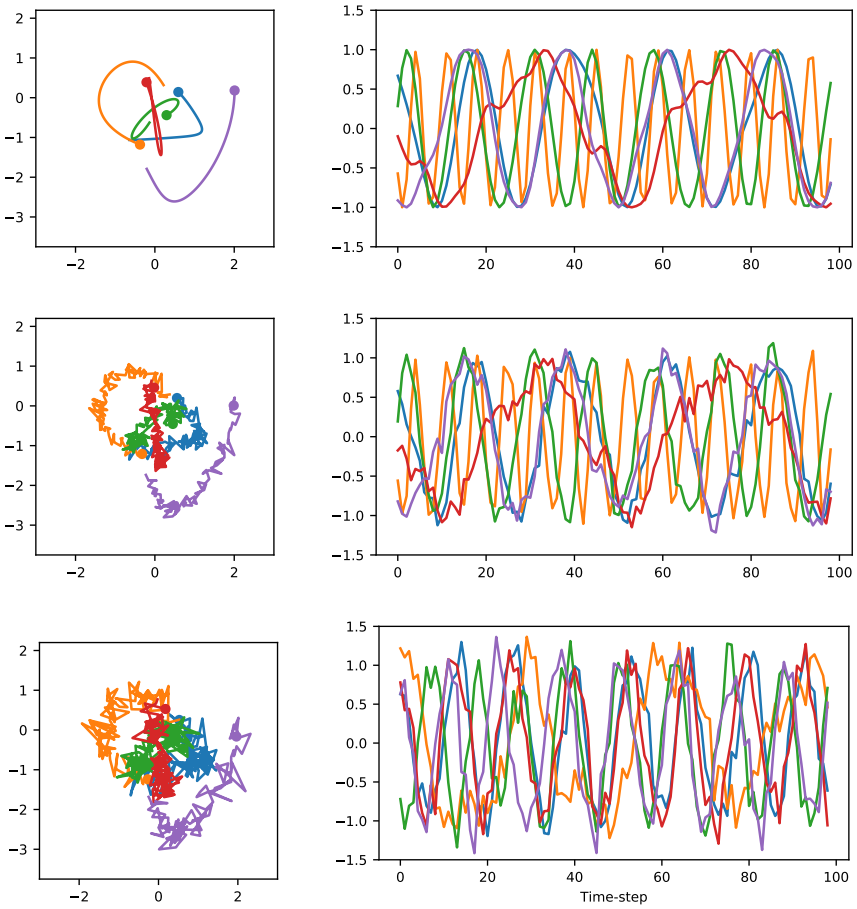

Figure 8: Example trajectories with different levels of observation noise for the Particles dataset (left) and Kuramoto (right). The observation noise is sampled from a zero-mean Gaussian distribution with standard deviation of (from top to bottom) 0.0, 0.1 and 0.2, respectively.

C Amortized Causal Discovery with Unobserved Variables

C.1 Temperature Experiments

Implementation Details In this experiment, we use the CNN encoder and a simulation decoder matching the true generative ODE process. Our optimization scheme is the same as before.

For modeling the latent temperature, we output a uniform distribution as our posterior $q_{\phi_c}(c|\boldsymbol{x})$. One tricky aspect about this is the KL-Divergence:

$$KL(q_{\phi_c}(c|\boldsymbol{x})||p(c)) = -\int q_{\phi_c}(c|\boldsymbol{x}) \log \frac{q_{\phi_c}(c|\boldsymbol{x})}{p(c)} d\boldsymbol{z} \quad . \tag{19}$$

We must ensure that our posterior support is a subset of our prior support. Otherwise, the KL-Divergence is undefined and optimization impossible. Recall that our prior is a uniform distribution over $[0, 4\alpha]$.

We output two latent parameters $a, b \in \mathbb{R}$ for each input and use these values to parametrize a mean m and a half-width w for the uniform distribution. First, we bound these values to represent a uniform distribution u_1 in $[0, 1]$. To achieve this, we let $m_1 = \sigma(a)$ and $w_1 = \sigma(b) * \min(m_1, 1 - m_1)$ with $\sigma(x) = \frac{1}{1+\exp(-x)}$. We then sample a temperature $\hat{c}_1 \sim u_1 = U(m_1 - w_1, m_1 + w_1)$, which is guaranteed to be bounded within $[0, 1]$. Stopping gradients, we use this temperature sample in the encoder $q_\phi(\boldsymbol{z}|\boldsymbol{x}, c)$ to improve the causal discovery performance.

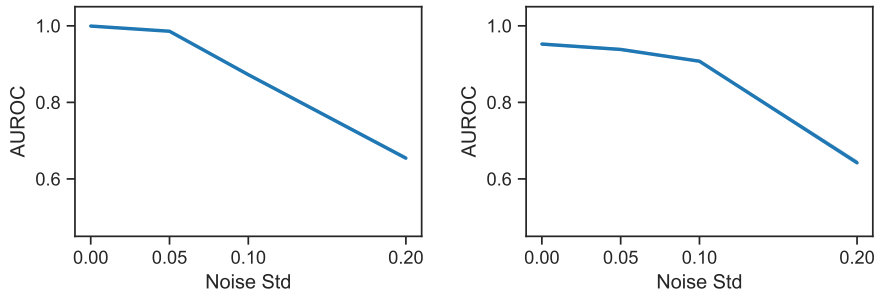

Figure 9: Causal discovery performance (in AUROC) on the particles dataset (A-left) and Kuramoto (B-right) for different levels of observation noise across five seeds. For comparison, the standard deviation of both noiseless datasets is about 0.6.

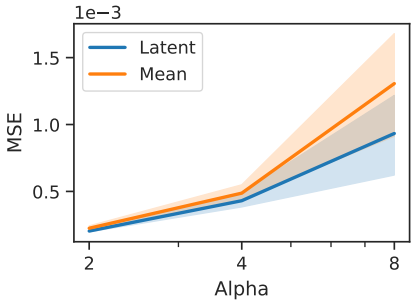

Figure 10: MSE (lower better) averaged across 5 random seeds for hidden temperature experiment. MSE for *None* baseline was much worse with MSE $= 0.009, 0.02, 0.04$ for $\alpha = 2, 4, 8$ (not shown in plot).

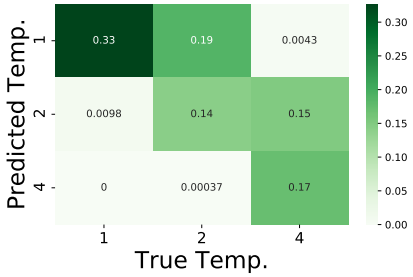

Figure 11: Confusion matrix for latent temperature prediction with $\alpha = 2$. ACD with Latent's prediction tends to be conservative: it is more likely to predict a too low temperature than a too high one.

Next, we scale this result to the desired interval $[0, 4\alpha]$. To achieve this, we feed the scaled temperature $\hat{c} = 4\alpha\hat{c}_1$ into the decoder, and use the scaled distribution $u = U(4\alpha m_1 - 4\alpha w_1, 4\alpha m_1 + 4\alpha w_1)$ to find our KL term. We allow gradients to flow through the temperature sample in both the decoder and the distribution in the KL term, which informs our parameter updates.

Additional Results Similarly to Fig. 5, we show the future prediction performance in MSE across different values of α in Fig. 10. Again we find a slight improvement in performance when using ACD with *Latent* compared to the baselines, although this is a noisier indicator.

Additionally, we evaluate how well the introduced latent variable learns to predict the unobserved temperature. To do so, we use the mean of the predicted posterior uniform distribution. When a discrete categorical prediction is needed for evaluation, we quantize our results into three bins based on their distance in log-space. To calculate AUROC in this three category ordinal problem, we average the AUROC between the two binary problems: category 1 vs not category 1, and category 3 vs not category 3 (category 2 vs not 2 is not a valid regression task for the purposes of AUROC which is concerned with ordering, since it is the middle temperature and hence the labels would not be linearly separable).

The confusion matrix between true and predicted temperature in Fig. 11 indicates that ACD with Latent's prediction tends to be conservative: it is more likely to predict a too low temperature than a too high one. This is probably due to higher temperatures incurring larger MSEs, since higher temperature systems are more chaotic and thus less predictable.

Table 3 lists the temperature prediction results across all tested values of α. We find that we can predict the unobserved temperature quite well, especially with respect to ordering (as measured by correlation and AUROC).

	α		
	2	4	8
Correlation	0.888	0.844	0.661
Accuracy	0.644	0.384	0.346
AUROC (1vAll)	0.966	0.935	0.843

Table 3: Latent Temperature Prediction Metrics. We treat the mean of the outputted interval of the uniform posterior as the predicted temperature. For accuracy, this value discretized in log space to get a ternary prediction. *AUC (1vAll)* averages the two one-vs-all AUC values which can be calculated in a 3-category ordinal problem.

C.2 Unobserved Time-series

Implementation Details For modeling the unobserved time-series, we employ a two-layered, bi-directional long short-term memory (LSTM) [17] with a latent dimension of size 256.

Additional Results The full evaluation of our experiments with an unobserved time-series can be found in Table 4. Our results indicate that our proposed method *ACD with latent* predicts the trajectory of the unobserved time-series (unobserved MSE) more accurately than the *Mean* imputation baseline. Even though this prediction is worse than for the *Supervised* baseline, ACD with *Latent* manages to recover the performance of the fully *Observed* baseline better than the *None* and the *Mean* imputation baselines.

Method	AUROC	Accuracy	MSE	unobserved MSE
Observed	(0.99)	(0.993)	(0.00301)	-
Supervised	0.982	0.931	0.00822	0.0164
None	0.946	0.882	0.0119	-
Mean	0.951	0.881	0.0106	0.0397
ACD with latent	0.979	0.918	0.00747	0.0375

Table 4: Experiments with an unobserved time-series.

Fig. 12 shows the performance of the tested methods dependent on the number of time-series that are influenced by the unobserved one. In addition to Fig. 5 in our Experiments section, these plots show the achieved accuracy and MSE results. The general trends are the same. Fig. 13 shows example trajectories and the corresponding predictions for all tested methods.

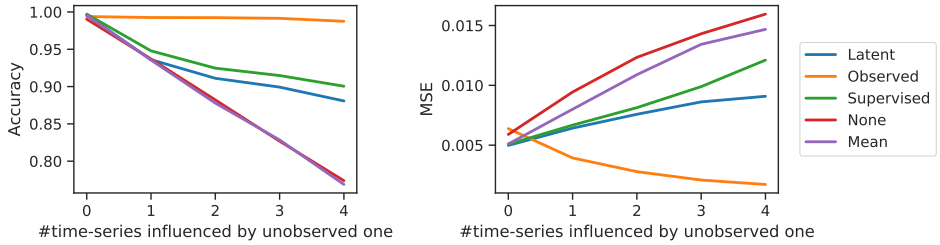

Figure 12: Experiments with an unobserved particle. Performance of the various methods depends strongly on how many observed particles are influenced by the unobserved one (x-axis). The more particles that are influenced by the unobserved particle, the stronger the benefit of using an additional *Latent* variable for modeling its effects. Left - causal relation prediction accuracy (higher = better), right - MSE (lower = better).

Additional Experiment: Uninfluenced Influencer Predicting the trajectory of a time-series that influences only a small number of observed time-series and is (invisibly) influenced by them is arguably very difficult. In this follow-up experiment, we reduce the difficulty of this problem by adding two assumptions: (1) the unobserved time-series influences *all* observed time-series and (2) it is not influenced by any of the observed time-series. This way, we gain more information about its trajectory (due to (1)) and its trajectory becomes easier to predict (due to (2)). Indeed, in this setup, *ACD with latent* manages to almost completely recover the performance of the fully observed

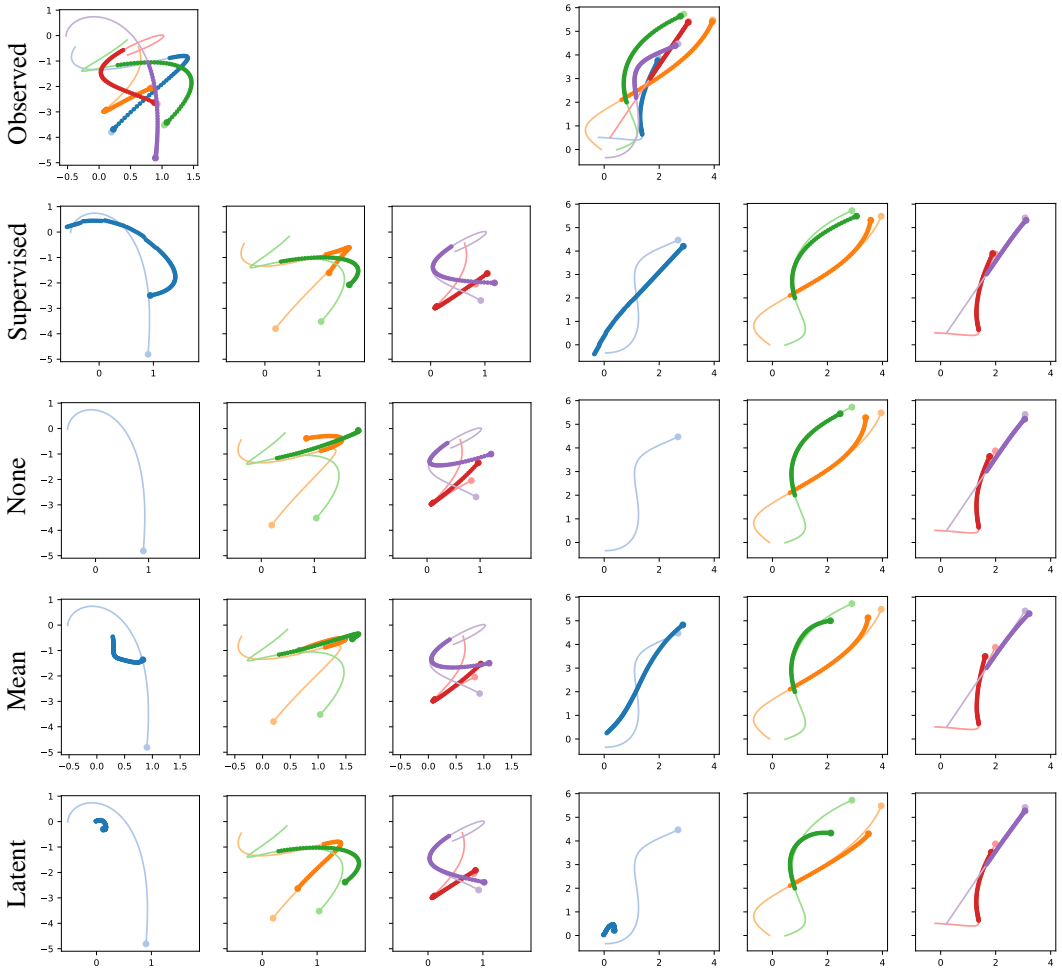

Figure 13: Predicted trajectories for all tested methods in the unobserved time-series experiment for two samples (left/right). From top to bottom: Baselines – observed, supervised, none and mean; proposed ACD with latent. The faded lines depict the ground truth trajectory; bold lines are the trajectories predicted by the model and they start after initializing the model using first half of the ground truth. Dots denote the end of the trajectories. Except for the fully observed baseline, the first panel shows the ground truth and prediction for the unobserved time-series. The second panel shows the trajectories of all time-series that are directly influenced by the unobserved one. The third panel shows the trajectories of all time-series that are not directly influenced by the unobserved one.

baseline (Table 5). In contrast, the performance of the *None* and *Mean* imputation baselines worsens considerably in this setting. Now, all time-series are influenced by the unobserved particle - making their prediction harder when not taking into account this hidden confounder. Fig. 14 shows example trajectories and the corresponding predictions for all tested methods in this setting.

Method	AUROC	Acuracy	MSE	unobserved MSE
Observed	(1.0)	(0.997)	(0.0193)	-
Supervised	1.0	0.993	0.024	0.000615
None	0.829	0.76	0.0431	-
Mean	0.853	0.782	0.0365	0.0357
ACD with latent	1.0	0.994	0.0251	0.137

Table 5: Experiments with an unobserved time-series that influences all observed time-series, but is not influenced by them.

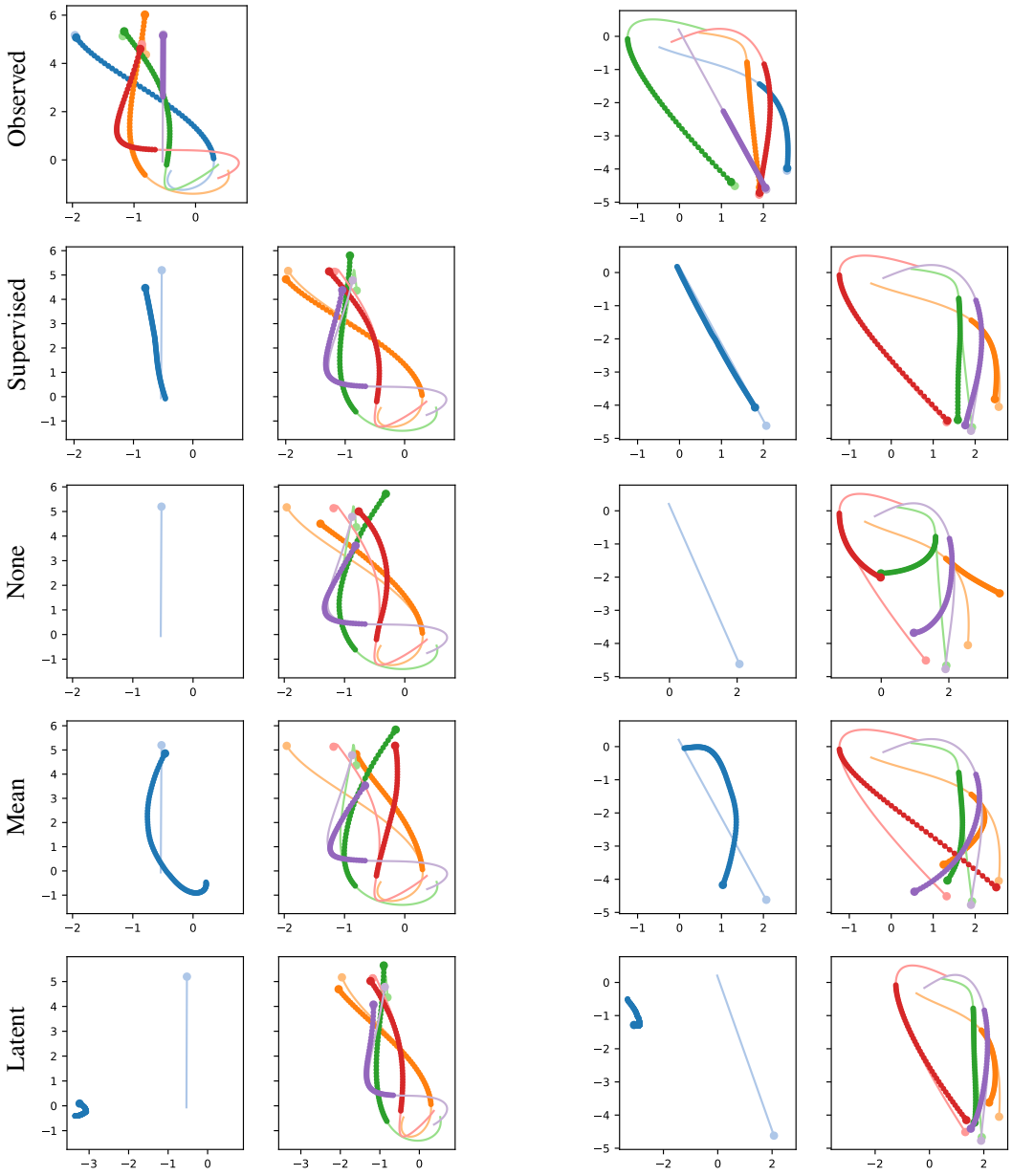

Figure 14: Predicted trajectories for all tested methods when the unobserved time-series influences all observed ones, but stay uninfluenced itself for two samples (left/right). From top to bottom: Baselines – observed, supervised, none and mean; proposed ACD with latent. The faded lines depict the ground truth trajectory; bold lines are the trajectories predicted by the model and they start after initializing the model using first half of the ground truth. Dots denote the end of the trajectories. Except for the fully observed baseline, the first panel shows the ground truth and prediction for the unobserved time-series. The second panel shows the trajectories of all observed time-series (which are all influenced by the unobserved one).