A VAE-based Framework for Learning Multi-Level Neural Granger-Causal Connectivity

Anonymous authors
Paper under double-blind review

Abstract

Granger causality has been widely used in various application domains to capture lead-lag relationships amongst the components of complex dynamical systems, and the focus in extant literature has been on a single dynamical system. In certain applications, one has access to data from a collection of *related* such systems, wherein the modeling task of interest is to extract the shared common structure that is embedded across them, as well as to identify the idiosyncrasies within individual ones. This paper introduces a Variational Autoencoder (VAE) based framework that *jointly* learns Granger-causal relationships amongst components in a collection of related-yet-heterogeneous dynamical systems, and handles the aforementioned task in a principled way. The performance of the proposed framework is evaluated on several synthetic data settings and benchmarked against existing approaches designed for individual system learning. The method is further illustrated on a real dataset involving neuroimaging time series data and produces interpretable results.

1 Introduction

The concept of Granger causality introduced in Granger (1969) leverages the temporal ordering of time series data. It is defined in terms of predictability of future values of a time series; namely, whether the inclusion of past information (lag values) of other time series as well as its own (self lags) leads to a reduction in the variance of the prediction error of the time series under consideration. Since its introduction, it has become a widely-used approach in the analysis of economic (Stock & Watson, 2001), financial (Hong et al., 2009) and neuroimaging (Seth et al., 2015) time series data. The standard setting in these applications is that one is interested in estimating Granger causal relationships in a dynamical system (e.g., a national economy, a brain) comprising of p variables.

Granger causality can also be expressed through the language of graphical models (Dahlhaus & Eichler, 2003; Eichler, 2012). The node set of the graph corresponds to the p variables at different time points; *directed* edges between nodes at past time points to those at present one capture Granger causal relationships. Traditionally, Granger causality was operationalized through linear vector autoregressive (VAR) models (Granger, 1969), in which case the entries of the estimated transition matrices correspond precisely to the edges of the Granger causal graph. More recent work has explored how Granger causal relationships can be learned through nonlinear models; e.g., see review paper Shojaie & Fox (2022) and references therein.

In certain application domains, one has access to data from a collection of *related* dynamical systems. A motivating example is described next. Consider electroencephalography (EEG) recordings obtained from p electrodes placed on the scalp of a subject (e.g., a patient or an animal). The resulting time series data constitute measurements from a complex neurophysiological dynamical system (Stam, 2005). On many instances, one has access to such measurements for a collection of M *related* subjects (or "entities", equivalently); for example, they may be performing the same cognitive task (e.g., visual counting, geometric figure rotation) or exhibit a similar neurological disorder (e.g., epilepsy, insomnia, dementia). In such a setting, one can always opt to perform separate analyses on each subject's data; however, it would be useful to develop methodology that models the data from all subjects *jointly*, so as to simultaneously extract the embedded structure shared across subjects and identify the idiosyncracies (heterogeneity) in any single one. In other

words, if one views all subjects as belonging to a common group, the quantities of interest are the shared group-level connectivity structure (amongst nodes) and the entity-level ones.

Conceptually, the above-mentioned modeling task is not difficult to fulfill in a linear setting where one can decompose the transition matrices into a "shared" component and an idiosyncratic (entity-specific) one, with some orthogonality-type constraint to enforce identifiability of the parameters. However, the task becomes more challenging and involved in non-linear settings where one hopes to use flexible models to capture the underlying complex dynamics. In particular, a decomposition-based approach, which requires the exact specification of the functional form of the shared component or how the shared and the idiosyncratic components interact, would be rather restrictive. To this end, we adopt a generative model-based approach, which circumvents the issue by encoding the Granger causal relationships through graphs. By postulating a model with a hierarchical structure between the shared and entity-specific components, the problem can be addressed in a flexible, yet principled manner.

Summary of contributions. We develop a two-layer Variational Autoencoder (VAE) based framework for estimating Granger-causal connections amongst nodes in a collection of related dynamical systems — jointly for the common group-level and the entity-level ones — in the presence of entity-specific heterogeneity. Depending on the assumed connection type (continuous or binary) amongst the nodes, the proposed framework can accommodate the scenario accordingly by imposing a commensurate structure on the encoded/decoded distributions, leveraging conjugacy between pairs of distributions. The proposed model enables extracting the embedded common structure in a principled way, without resorting to any ad-hoc or post-hoc aggregation. Finally, the framework can be generalized to the case where multiple levels of nested groups are present and provides estimates of the group-level connectivity for all levels of groups.

The remainder of the paper is organized as follows. In Section 2, we provide a review of related literature on Granger-causality estimation, with an emphasis on neural network-based methods. The main building block used in the proposed framework, namely, a multi-layer VAE is also briefly introduced. Section 3 describes in detail the proposed framework, including the encoder/decoder modules and the training/inference procedure. In Section 4, model performance is assessed on synthetic datasets and benchmarked against several existing methods. An application to a real dataset involving EEG signals from 22 subjects is discussed in Section 5. Finally, Section 6 concludes the paper.

2 Related Work and Preliminaries

In this section, we review related work on inferring Granger causality based on time series data, with an emphasis on deep neural network-based approaches. Further, as the proposed framework relies on variational autoencoders (VAE) with a hierarchical structure, we also briefly review VAEs in the presence of multiple latent layers.

2.1 Inference of Granger causality

Linear VAR models have historically been the most popular approach for identifying Granger causal relationships. Within the linear setting, hypothesis testing frameworks with theoretical guarantees have been used (Granger, 1980; Geweke, 1984), while more recently regularized approaches have enabled the estimation in the high-dimensional setting (Basu et al., 2015). Recent advances in neural network techniques have facilitated capturing non-linear dynamics and identifying Granger causality accordingly, as discussed next.

Note that estimation of Granger causality is an *unsupervised* task, in the sense that the connectivity as captured by the underlying graph is *not observed* and thus cannot serve as the supervised learning target. However, depending on the model family that the associated estimation procedure falls into, existing approaches suitable for estimating Granger causality based on neural networks (Montalto et al., 2015; Nauta et al., 2019; Wu et al., 2020; Khanna & Tan, 2020; Tank et al., 2021; Marcinkevičs & Vogt, 2021; Löwe et al., 2022) can be broadly categorized into supervised and generative model-based ones. We selectively review some of them next. In the remainder of this subsection, we use $x_{i,t}$ to denote the value of node i at time

t, $\mathbf{x}_t := (x_{1,t}, \cdots, x_{p,t})$ the collection of node values of the dynamical system, and $\mathbb{x} := \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ the trajectory over time.

Within the supervised modeling framework, recent representative works include Khanna & Tan (2020); Tank et al. (2021); Marcinkevičs & Vogt (2021), where the Granger-causal relationship is inferred from coefficients that govern the dynamics of the time series, and the coefficients are learned by formulating prediction tasks that can be generically represented as $\mathbf{x}_t = f(\mathbf{x}_{t-1}, ..., \mathbf{x}_{t-q}) + \boldsymbol{\varepsilon}_t$, with $\mathbf{x}_t \in \mathbb{R}^p$ being the multivariate time series signal and $\boldsymbol{\varepsilon}_t$ the noise term. In Tank et al. (2021), coordinates of the response are considered separately, that is, $x_{i,t} = f_i(\mathbf{x}_{t-1}, ..., \mathbf{x}_{t-q}) + \varepsilon_{i,t}$, and f_i is parameterized using either multi-layer perceptrons (MLP) or LSTM (Hochreiter & Schmidhuber, 1997). In the case of an L-layer MLP,

$$\widehat{x}_{i,t} = W^L \mathbf{h}_t^{L-1} + \mathbf{b}^L; \quad \mathbf{h}_t^l = \sigma\Big(W^l \mathbf{h}_t^{l-1} + b^l\Big),\ l = 2, \cdots, L; \quad \mathbf{h}_t^1 = \sigma\Big(\sum\nolimits_{k=1}^{q} W^{1k} \mathbf{x}_{t-k} + \mathbf{b}^1\Big);$$

the Granger-causal connection from the jth node to the ith node is then encoded in some "summary" (e.g., Frobenius norm) of $\{W_{:j}^{11}, \cdots, W_{:j}^{1q}\}$, with each component corresponding to the first hidden layer weight of lags $x_{j,t-1}, \cdots, x_{j,t-q}$. Various regularization schemes are considered and incorporated as penalty terms in the loss function, to encourage sparsity and facilitate the identification of Granger-causal connections. The case of LSTM-based parameterization is handled analogously. Marcinkevičs & Vogt (2021) parameterizes f as an additive function of the lags, i.e., $\mathbf{x}_t = \sum_{k=1}^{q} \Psi_k(\mathbf{x}_{t-k})\mathbf{x}_{t-k} + \boldsymbol{\varepsilon}_t$; the output of $\Psi_k : \mathbb{R}^p \mapsto \mathbb{R}^{p \times p}$ contains the generalized coefficients of $\mathbf{x}_{t-k}$, whose (i, j) entry corresponds to the impact of $x_{j,t-k}$ on $x_{i,t}$ and Ψ_k is parameterized through MLPs. The Granger causal connection between the jth node and the ith node is obtained by aggregating information from the coefficients of all lags $\{\Psi_k(\mathbf{x}_{t-k})_{ij}\}$, i.e., $\max_{1 \le k \le q}\{\text{median}_{q+1 \le t \le T}(|\Psi_k(\mathbf{x}_{t-k})_{ij}|)\}$. Finally, an additional stability-based procedure where the model is fit to the time series in the reverse order is performed for the final selection of the connections.

For generative model-based approaches, the starting point is slightly different. Notable ones include Löwe et al. (2022) that builds upon Kipf et al. (2018), and the focus is on *relational inference*. The postulated generative model assumes that the trajectories are collectively governed by an underlying latent graph $\mathbf{z}$, which effectively encodes Granger-causal connections:

$$p(\mathbb{x}|\mathbf{z}) = p(\{\mathbf{x}_{T+1}, \cdots, \mathbf{x}_1\}|\mathbf{z}) = \prod\nolimits_{t=1}^{T} p(\mathbf{x}_{t+1}|\mathbf{x}_t, \cdots, \mathbf{x}_1, \mathbf{z}).$$

Specifically, in their setting, $x_{i,t} \in \mathbb{R}^d$ is vector-valued; z_{ij} corresponds to a categorical "edge type" between nodes i and j. For example, it can be a binary edge type indicating presence/absence, or a more complex one having more categories. To simultaneously learn the edge types and the temporal dynamics, the model is formalized through a VAE that maximizes the evidence lower bound (ELBO), given by $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbb{x})}(\log p_\theta(\mathbb{x}|\mathbf{z})) - \text{KL}(q_\phi(\mathbf{z}|\mathbb{x}) \,\big\|\, p_\theta(\mathbf{z}))$, where $q_\phi(\mathbf{z}|\mathbb{x})$ is the probabilistic encoder, $p_\theta(\mathbb{x}|\mathbf{z})$ the decoder, and $p_\theta(\mathbf{z})$ the prior distribution.

In summary, at the formulation level, generative model-based approaches treat Granger-causal connections (relationships) as a latent graph and learn it jointly with the dynamics, whereas supervised ones extract Granger-causal connections from the parameters that govern the dynamics in a post-hoc manner. The former can readily accommodate vector-valued nodes whereas for the latter, it becomes more involved and further complicates how the connections can be extracted/represented based on the model parameters. At the task level, to learn the model parameters, supervised-model based approaches rely on prediction tasks where the values of future timestamps are of interest, whereas generative approaches amount to reconstructing the observed trajectories; prediction and reconstruction errors constitute part of the empirical risk minimization loss and the ELBO loss, respectively.

2.2 Multi-layer variational autoencoders

With a slight abuse of notation, in this subsection, we use $\mathbf{x}$ to denote the observed variable and $\mathbf{z}_l, l = 1, \cdots, L$ the latent ones for L layers.

A "shallow" VAE with one latent layer is considered in the seminal work of Kingma & Welling (2014), where the generative model is given by $p_\theta(\mathbf{x}, \mathbf{z}_1) = p_\theta(\mathbf{x}|\mathbf{z}_1)p_\theta(\mathbf{z}_1)$, with $p_\theta(\mathbf{z}_1)$ denoting the prior distribution.

Later works (Kingma et al., 2014; Burda et al., 2016; Sønderby et al., 2016) consider the extension into multiple latent layers, where the generative model can be represented through a cascading structure as follows:

$$p_\theta(\mathbf{x}, \{\mathbf{z}_l\}_{l=1}^L) = p_\theta(\mathbf{x}|\mathbf{z}_1)\Big(\prod\nolimits_{l=1}^{L-1} p_\theta(\mathbf{z}_l|\mathbf{z}_{l+1})\Big)p_\theta(\mathbf{z}_L);$$

the corresponding inference model (encoder) is given by $q_\phi(\mathbf{z}_1, \cdots, \mathbf{z}_L|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x}) \prod_{i=1}^L q_\phi(\mathbf{z}_l|\mathbf{z}_{l-1})$. The variational lower bound on $\log p(\mathbf{x})$ can be written as

$$\mathbb{E}_{q_\phi(\{\mathbf{z}\}_{l=1}^L|\mathbf{x})}\Big(\log p_\theta(\mathbf{x}|\{\mathbf{z}\}_{l=1}^L)\Big) - \mathrm{KL}\Big(q_\phi(\{\mathbf{z}\}_{l=1}^L|\mathbf{x}) \,\Big\|\, p_\theta(\{\mathbf{z}\}_{l=1}^L)\Big), \tag{1}$$

with the first term corresponding to the reconstruction error.

Conjugacy adjustment. Under the above multi-layer setting, Sønderby et al. (2016) considers an inference model that recursively merges information from the "bottom-up" encoding and "top-down" decoding steps. Concretely, in the case where each layer is specified by a Gaussian distribution, the original distribution at layer l after encoding is given by $q_\phi(\mathbf{z}_l|\mathbf{z}_{l-1}) \sim \mathcal{N}(\mu_{q,l}, \sigma_{q,l}^2)$ and the distribution at the same layer after decoding is given by $p_\theta(\mathbf{z}_l|\mathbf{z}_{l+1}) \sim \mathcal{N}(\mu_{p,l}, \sigma_{p,l}^2)$. The adjustment amounts to a precision-weighted combination that combines information from the decoder distribution into the encoder one, that is, $q_\phi(\mathbf{z}_l|\cdot) \sim \mathcal{N}\big(\tilde{\mu}_{q,l}, \tilde{\sigma}_{q,l}^2\big)$, where $\tilde{\mu}_{q,l} = (\mu_{q,l}\sigma_{q,l}^{-2} + \mu_{p,l}\sigma_{p,l}^{-2})/(\sigma_{q,l}^{-2} + \sigma_{p,l}^{-2})$ and $\tilde{\sigma}_{q,l}^2 = 1/(\sigma_{q,l}^{-2} + \sigma_{p,l}^{-2})$. This information-sharing mechanism leads to richer latent representations and improved approximation of the log-likelihood function. A similar objective is also considered in Burda et al. (2016) and operationalized through importance weighting.

Finally, it is worth noting that although it was not mentioned in the original paper (Sønderby et al., 2016), the precision-weighted adjustment coincides precisely with the conjugate analysis of normally distributed data in Bayesian statistics, where the prior distribution is also assumed to be Gaussian with known variance. For this reason, we term such adjustment as the "conjugacy adjustment", which will be used later in our technical development.

3 The Proposed Framework

Given a collection of trajectories for the same set of p variables (nodes) from M dynamical systems (entities), we are interested in estimating the Granger causal connections amongst the nodes in each system (i.e., entity-level connections), as well as the common "backbone" connections amongst the nodes that are shared across the entities (i.e., group-level connections).

To this end, we propose a two-layer VAE-based framework, wherein Granger-causal connections are treated as latent variables, segmented into multiple layers, and they are learned jointly with the dynamics of the trajectories. In Section 3.1, we present the posited generative process that is suitable for the modeling task of interest, and give an overview of the proposed VAE-based formulation; the details of the components involved and their exact modeling considerations are discussed in Section 3.2. Section 3.3 provides a summary of the end-to-end training process and the inference tasks that can be performed based on the trained model.

The generalization of the proposed framework to the case of multiple levels of grouping across entities is deferred to Appendix C, where the *grand* common and the *group* common structures can be simultaneously learned with those of the entities.

3.1 An overview of the formulation

Consider a setting where there are M entities, each of them having the same set of p nodes, that evolve as a dynamical system. Let $x_{i,t}^{[m]}$ denote the value of node i of entity $m \in \{1, \cdots, M\}$ at time t. It can be either scalar or vector-valued, with scalar node values being prevalent in traditional time-series settings; in the latter case, it can correspond to the values of node features (e.g., Kipf et al., 2018). Let $\mathbf{x}_t^{[m]} := (x_{1,t}^{[m]}, \cdots, x_{p,t}^{[m]})$ be the collection of node values at time t for entity m, and $\mathbb{x}^{[m]} := \{\mathbf{x}_1^{[m]}, \cdots, \mathbf{x}_T^{[m]}\}$ the corresponding trajectory over time. Further, let $\mathbf{z}^{[m]} \in \mathbb{R}^{p\times p}$ denote the Granger-causal connection matrix of entity m, whose (i, j)

entry $z_{ij}^{[m]}$ corresponds to the impact of the jth node on the ith and is a scalar; $\bar{\mathbf{z}} := [\bar{z}_{ij}] \in \mathbb{R}^{p\times p}$ denotes the common structure embedded in $\mathbf{z}^{[1]}, \cdots, \mathbf{z}^{[M]}$, and note that it does *not* necessarily correspond to the arithmetic mean of the $\mathbf{z}^{[m]}$'s. In the remainder of this paper, we may refer to these matrices as "graphs" interchangeably.

The posited generative process, whose true parameters are denoted by $\theta^\star$, is given by:

$$\begin{aligned} p_{\theta^\star}\Big(\{\mathbb{x}^{[m]}\}_{m=1}^M, \{\mathbf{z}^{[m]}\}_{m=1}^M, \bar{\mathbf{z}}\Big) &= p_{\theta^\star}\Big(\{\mathbb{x}^{[m]}\}_{m=1}^M | \{\mathbf{z}^{[m]}\}_{m=1}^M\Big) \cdot p_{\theta^\star}\Big(\{\mathbf{z}^{[m]}\}_{m=1}^M | \bar{\mathbf{z}}\Big) \cdot p_{\theta^\star}(\bar{\mathbf{z}}) \\ &= \prod_{m=1}^M p_{\theta^\star}(\mathbb{x}^{[m]}|\mathbf{z}^{[m]}) \prod_{m=1}^M p_{\theta^\star}(\mathbf{z}^{[m]}|\bar{\mathbf{z}}) \prod_{1\le i,j\le p} p_{\theta^\star}(\bar{z}_{ij}). \end{aligned} \tag{2}$$

The decomposition is based on the following underlying assumptions (see also Figure 1 for a pictorial illustration):

- conditional on the entity-specific graphs $\mathbf{z}^{[m]}$, their trajectories $\mathbb{x}^{[m]}$'s are *independent* of the grand common $\bar{\mathbf{z}}$, and they are conditionally independent from each other given their respective entity-specific graphs $\mathbf{z}^{[m]}$'s
- the entity-specific graphs $\mathbf{z}^{[m]}$ are conditionally independent given the common graph $\bar{\mathbf{z}}$
- the prior distribution $p_{\theta^\star}(\bar{\mathbf{z}})$ factorizes over the edges.

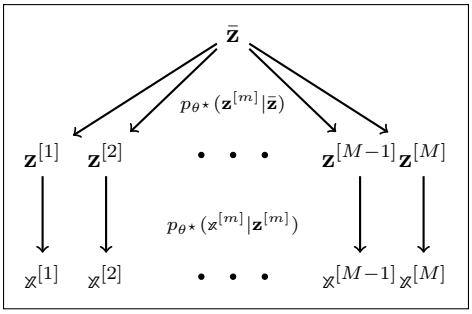


Figure 1: Diagram for the postulated top-down generative process.

The proposed model creates a hierarchy between the common graph and the entity-specific ones, which in turn naturally provides a coupling mechanism amongst the latter. The grand common structure can be estimated as one learns all the latent components jointly with the dynamics of the system through a VAE. Let $\mathcal{X} := \{\mathbb{x}^{[1]}, \cdots, \mathbb{x}^{[m]}\}$, $\mathcal{Z} := \{\bar{\mathbf{z}}, \mathbf{z}^{[1]}, \cdots, \mathbf{z}^{[m]}\}$, $q_\phi(\mathcal{Z}|\mathcal{X})$ denote the encoder, $p_\theta(\mathcal{X}|\mathcal{Z})$ the decoder and $p_\theta(\mathcal{Z})$ the prior distribution. Then, the ELBO is given by

$$\mathbb{E}_{q_\phi(\mathcal{Z}|\mathcal{X})}\Big(\log p_\theta(\mathcal{X}|\mathcal{Z})\Big) - \mathrm{KL}\Big(q_\phi(\mathcal{Z}|\mathcal{X}) \,\Big\|\, p_\theta(\mathcal{Z})\Big),$$

and serves as the objective function for the end-to-end encoding-decoding procedure as depicted in Figure 2.

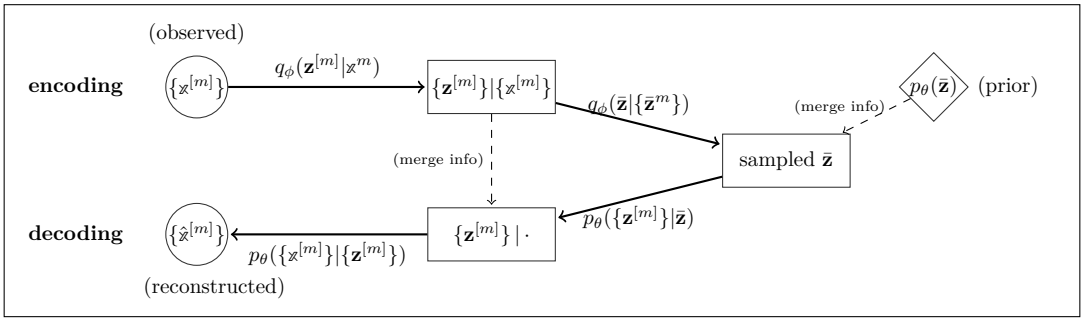


Figure 2: Diagram for the end-to-end encoding-decoding procedure. Solid paths with arrows denote modeling the corresponding distributions during the encoding/decoding process; dashed paths with arrows correspond to information merging based on (weighted) conjugacy adjustment. Quantities obtained after each step are given inside the circles/rectangles. $\{\mathbb{x}^{[m]}\}$ is short for the collection $\{\mathbb{x}^{[m]}\}_{m=1}^M$; $\{\mathbf{z}^{[m]}\}$ is analogously defined.

Remark 1 (on the proposed formulation). (1) Depending on the modeling scenario, the entity-level Granger-causal connections $\mathbf{z}^{[m]}$ can either be continuous with values reflecting the strength of the relationships, or binary thus indicating presence/absence of connections. Encoder/decoder distributions can then be selected accordingly. In particular, distributions that form conjugate pairs (e.g., Gaussian-Gaussian for the continuous case and Beta-Bernoulli for the binary case) can facilitate computations. (2) The proposed framework

naturally allows estimation of positive/negative connections in a principled way without resorting to ad-hoc aggregation schemes. It also enables incorporation of external information pertaining to the presence/absence of connections through the decoder. (3) In settings where a large collection of entities is available, but each entity has limited sample size, the joint learning framework can be advantageous over an individual entity learning one.

3.2 Modeling details

Next, we provide details on the specification of the encoder and the decoder, the sampling steps, and the loss function calculations for model (2).

3.2.1 Encoder

The goal of the encoder is to infer the latent graphs $\bar{\mathbf{z}}$ and $\mathcal{Z} := \{\mathbf{z}^{[1]}, \cdots, \mathbf{z}^{[M]}\}$ based on the observed trajectories $\mathcal{X} := \{\mathbb{x}^{[1]}, \cdots, \mathbb{x}^{[M]}\}$, $m = 1, \cdots, M$.

Let ϕ denote the collection of parameters in the encoder $q_\phi(\mathcal{Z}|\mathcal{X})$. To delineate the dependency between the trajectories and the graphs, the following assumptions are imposed:

- conditioning on $\{\mathbf{z}^{[m]}\}_{m=1}^M$, $\bar{\mathbf{z}}$ is independent of $\{\mathbb{x}^{[m]}\}_{m=1}^M$ and the conditional probability $q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}_{m=1}^M\big)$ factorizes across edges (i, j);
- the entity-specific graphs are conditionally independent given their corresponding trajectories, i.e., $q_\phi\big(\{\mathbf{z}^{[m]}\}_{m=1}^M|\{\mathbb{x}^{[m]}\}_{m=1}^M\big)$ factorizes across entities.

These assumptions are in line with the structure of the model in (2), in that the conditional dependencies posited in the generative model are respected during the "bottom-up" encoding process.

Consequently, the encoder can be decomposed into the following product components:

$$q_\phi\big(\mathcal{Z}\big|\mathcal{X}\big) = q_\phi\Big(\bar{\mathbf{z}}\,\Big|\,\{\mathbf{z}^{[m]}\}_{m=1}^M\Big)\prod_{m=1}^M q_\phi\Big(\mathbf{z}^{[m]}\big|\mathbb{x}^{[m]}\Big) = \prod_{1\le i,j\le p} q_\phi\Big(\bar{z}_{ij}\,|\{z_{ij}^{[m]}\}_{m=1}^M\Big)\prod_{m=1}^M q_\phi\Big(\mathbf{z}^{[m]}\big|\mathbf{x}^{[m]}\Big).$$

There are two types of terms in the above expression: $q_\phi(\mathbf{z}^{[m]}|\mathbb{x}^{[m]})$ that infers each entity's latent graph based on its trajectory, and $q_\phi(\bar{z}_{ij}|\{z_{ij}^{[m]}\}_{m=1}^M)$ that obtains the grand common based on the entity-level graphs, in an edge-wise manner. Note that for $q_\phi(\bar{z}_{ij}|\{z_{ij}^{[m]}\}_{m=1}^M)$, together with modeling $p_\theta(z_{ij}^{[m]}|\bar{z}_{ij})$, resembles prior-posterior calculations in Bayesian statistics using conjugate pairs of distributions; hence, depending on the underlying structural assumptions (continuous or binary) on the $\mathbf{z}^{[m]}$'s, one can choose emission heads (or equivalently, the output functional form) accordingly.

At the high level, the encoder can be abstracted into 3 modules, parameterized through $f_{x\to h}, f_{h\to z}$ and $f_{z\to\bar{z}}$, respectively:

(enc-a) trajectory to hidden representation $\mathbb{x}^{[m]} \to \mathbf{h}^{[m]} := f_{x\to h}(\mathbb{x}^{[m]})$;

(enc-b) hidden representation to the entity-specific graph: $\mathbf{h}^{[m]} \to \mathbf{z}^{[m]} := f_{h\to z}(\mathbf{h}^{[m]})$;

(enc-c) entity-level graphs to the grand common (edge-wise): $\{z_{ij}^{[m]}\}_{m=1}^M \to \bar{z}_{ij} := f_{z\to\bar{z}}(\{z_{ij}^{[m]}\}_{m=1}^M)$.

Modules (enc-a) and (enc-b) combined, model $q_\phi(\mathbf{z}^{[m]}|\mathbb{x}^{[m]})$, while module (enc-c) models $q_\phi(\bar{z}_{ij}|\{z_{ij}^{[m]}\}_{m=1}^M)$. On the other hand, given the above-mentioned conjugate pair consideration, the choices of $f_{h\to z}$ and $f_{z\to\bar{z}}$ are considered jointly.

Formally, for $f_{x\to h}$, we use a similar approach to that in Kipf et al. (2018), where $f_{x\to h}$ entails message-passing operations that are widely adopted in the literature related to graph neural networks (Scarselli et al., 2008; Gilmer et al., 2017). At a high level, these operations entail "node2edge" (concatenating the representation of the node stubs) and "edge2node" (aggregating the representation of incoming edges) iteratively and non-linear functions (e.g., MLPs) in between (full details provided in Appendix A.1). The operation ultimately leads to $\{\mathbf{h}_{ij}^{[m]}\}$, with $\mathbf{h}_{ij}^{[m]} \in \mathbb{R}^{n_{\text{hid}}}$ being a n_{hid}-dimensional hidden representation corresponding to $z_{ij}^{[m]}$.

Once the $\mathbf{h}_{ij}^{[m]}$'s are obtained, subsequent modeling in modules (enc-b) and (enc-c) can be generically represented as

$$z_{ij}^{[m]} \,|\, \mathbf{h}_{ij}^{[m]} \sim q_z(\cdot\,; \delta_{q,ij}^{[m]}), \qquad \text{and} \qquad \bar{z}_{ij} \,|\{z_{ij}^{[m]}\} \sim q_{\bar{z}}(\cdot\,; \bar{\delta}_{ij}),$$

where $q_z(\cdot\,; \delta_{q,ij}^{[m]})$ is some distribution with parameter $\delta_{q,ij}^{[m]} := f_{h\to z}(\mathbf{h}_{ij}^{[m]})$ being the function output of $f_{h\to z}$. Similarly, $q_{\bar{z}}(\cdot\,; \bar{\delta}_{q,ij})$ is some distribution with parameter $\bar{\delta}_{q,ij} := f_{z\to\bar{z}}(\{z_{ij}^{[m]}\})$ being the function output of $f_{z\to\bar{z}}$. The exact choices for $f_{h\to z}$ and $f_{z\to\bar{z}}$ bifurcate depending on the scenario:

- Case 1, $\mathbf{z}^{[m]}$'s entries being continuous: in this case, we consider a Gaussian-Gaussian emission head pair. Consequently, $\delta_{q,ij}^{[m]} = \{\mu_{q,ij}^{[m]}, (\sigma^{[m]})_{q,ij}^2\}$, $\bar{\delta}_{q,ij} = \{\bar{\mu}_{q,ij}, \bar{\sigma}_{q,ij}^2\}$;

$$q_z \sim \mathcal{N}\Big(\mu_{q,ij}^{[m]}, (\sigma^{[m]})_{q,ij}^2\Big); \quad \mu_{q,ij}^{[m]} := f_{h\to z}^1(\mathbf{h}_{ij}^{[m]}),\ (\sigma^{[m]})_{q,ij}^2 := f_{h\to z}^2(\mathbf{h}_{ij}^{[m]}); \tag{3}$$

$$q_{\bar{z}} \sim \mathcal{N}\Big(\bar{\mu}_{q,ij}, \bar{\sigma}_{q,ij}^2\Big); \quad \bar{\mu}_{q,ij} := f_{z\to\bar{z}}^1(\{z_{ij}^{[m]}\}),\ \bar{\sigma}_{q,ij}^2 := f_{\bar{z}\to z}^2(\{z_{ij}^{[m]}\}). \tag{4}$$

  $f_{h\to z}^1, f_{h\to z}^2$ are component functions of $f_{h\to z}$, each with an n_{hid}-dimensional input and a scalar output; they can be simple linear functions with $f_{h\to z}^2$ having an additional softplus operation to ensure positivity. Similarly, $f_{z\to\bar{z}}^1, f_{z\to\bar{z}}^2$ comprise $f_{z\to\bar{z}}$, each with an m-dimensional input and a scalar output; in practice their functional form can be as simple as taking the sample mean and standard deviation, respectively.

- Case 2, $\mathbf{z}^{[m]}$'s entries being binary: in this case, we consider a Beta-Bernoulli emission head pair, i.e.,

$$q_z \sim \text{Ber}\Big(\delta_{q,ij}^{[m]}\Big); \quad \delta_{q,ij}^{[m]} := f_{h\to z}(\mathbf{h}_{ij}^{[m]}), \tag{5}$$

$$q_{\bar{z}} \sim \text{Beta}\Big(\bar{\alpha}_{q,ij}, \bar{\beta}_{q,ij}\Big); \quad \bar{\alpha}_{q,ij} := f_{z\to\bar{z}}^1(\{z_{ij}^{[m]}\}),\ \bar{\beta}_{q,ij} := f_{z\to\bar{z}}^2(\{z_{ij}^{[m]}\}). \tag{6}$$

  The output of $f_{h\to z}$ corresponds to the Bernoulli success probability and it is parameterized with an MLP with the last layer performing sigmoid activation to ensure that the output lies in $(0, 1)$. $f_{z\to\bar{z}}^1$ and $f_{z\to\bar{z}}^2$ are component functions of $f_{z\to\bar{z}}$. Similar to the Gaussian case, their choice need not be complicated and is chosen based on moment-matching.

Note that the prior distribution $p_\theta(\bar{z}_{ij})$ is also selected according to the underlying scenario, with a standard Normal distribution used in the continuous case and a Beta$(1, 1)$ in the binary case. Once the distribution parameters for $\bar{z}_{ij}$ are obtained based on (4) or (6), we apply conjugacy adjustment to incorporate also the information from the prior, before the sampling step takes place.

3.2.2 Decoder

The goal of the decoder $p_\theta(\mathcal{X}|\mathcal{Z})$ is to reconstruct the trajectories based on the entity and group level graphs, and its components follow from the generative process described in (2), that is,

$$p_\theta(\mathcal{X}|\mathcal{Z}) = p_\theta\Big(\{\mathbb{x}^{[m]}\}_{m=1}^M | \{\mathbf{z}^{[m]}\}_{m=1}^M\Big) \cdot p_\theta\Big(\{\mathbf{z}^{[m]}\}_{m=1}^M | \bar{\mathbf{z}}\Big) = \prod\nolimits_{m=1}^M p_\theta(\mathbb{x}^{[m]}|\mathbf{z}^{[m]}) \prod\nolimits_{m=1}^M p_\theta(\mathbf{z}^{[m]}|\bar{\mathbf{z}}),$$

where θ denotes the collections of parameters in the decoder. The two components $p_\theta(\mathbf{z}^{[m]}|\bar{\mathbf{z}})$ and $p_\theta(\mathbb{x}^{[m]}|\mathbf{z}^{[m]})$, respectively capture the dependency between the entity-specific graphs $\mathbf{z}^{[m]}$'s and their grand common $\bar{\mathbf{z}}$, and the evolution of the trajectories given $\mathbf{z}^{[m]}$. Consequently, the decoder can be broken into two modules, parameterized through $g_{\bar{z}\to z}$ and $g_{z\to x}$:

(dec-a) $p_\theta(\mathbf{z}^{[m]}|\bar{\mathbf{z}})$, the grand common to entity-specific graphs $\mathbf{z} \to \mathbf{z}^{[m]} := g_{\bar{z}\to z}(\bar{\mathbf{z}})$, with $g_{\bar{z}\to z}(\cdot)$ acting on the sampled $\bar{\mathbf{z}}$ (edge-wise). Samples drawn from this distribution will be used to guide the evolution of the trajectories of the corresponding entity;

(dec-b) $p_\theta(\mathbb{x}^{[m]}|\mathbf{z}^{[m]})$, graph to trajectory $\mathbf{z}^{[m]} \to \mathbb{x}^m$; concretely,

$$p_\theta(\mathbb{x}^{[m]}|\mathbf{z}^{[m]}) = p_\theta(\mathbf{x}_1^{[m]}|\mathbf{z}^{[m]}) \prod\nolimits_{t=2}^T p_\theta\Big(\mathbf{x}_t^{[m]} \,|\, \mathbf{x}_{t-1}^{[m]}, ..., \mathbf{x}_1^{[m]}, \mathbf{z}^{[m]}\Big),$$

with $p_\theta(\mathbf{x}_t^{[m]} \,|\, \mathbf{x}_{t-1}^{[m]}, ..., \mathbf{x}_1^{[m]}, \mathbf{z}^{[m]})$ modeled through $g_{z\to x}(\mathbf{x}_{t-1}^{[m]}, \cdots, \mathbf{x}_{t-q}^{[m]}, \mathbf{z}^{[m]})$ assuming a fixed context length of q (q-lag dependency, equivalently).

We refer to these two modules as "common2entity" and "graph2trajectory", respectively.

Common2Entity. We consider a weighted conjugacy adjustment that merges the information from the encoder distribution into the decoder one, so that it contains both the grand common and the entity-specific information. Concretely, for some pre-specified weight $\omega \in [0, 1]$,

- Case 1, in the continuous case, let $p_\theta(z_{ij}^{[m]}|\bar{z}_{ij}) \sim \mathcal{N}(\mu_{p,ij}^{[m]}, (\sigma^{[m]})_{p,ij}^2)$ with $\mu_{p,ij}^{[m]} := f_{\bar{z}\to z}^1(\bar{\mathbf{z}}_{ij}^{[m]})$ and $(\sigma^{[m]})_{p,ij}^2 := g_{\bar{z}\to z}^2(\bar{\mathbf{z}}_{ij}^{[m]})$; $g_{\bar{z}\to z}^1, g_{\bar{z}\to z}^2 : \mathbb{R} \mapsto \mathbb{R}$ are component functions of $g_{\bar{z}\to z}$. This gives the "unadjusted" distribution that contains only the grand common information. With $\mu_{q,ij}^{[m]}$ and $(\sigma^{[m]})_{q,ij}^2$ obtained in (3), the weighted adjustment gives $p_\theta(z_{ij}^{[m]}|\cdot) \sim \mathcal{N}\Big(\tilde{\mu}_{p,ij}^{[m]}, (\tilde{\sigma}^{[m]})_{p,ij}^2\Big)$, where

$$\tilde{\mu}_{p,ij}^{[m]} := \frac{\omega\mu_{q,ij}^{[m]}(\sigma^{[m]})_{q,ij}^{-2} + (1-\omega)\mu_{p,ij}^{[m]}(\sigma^{[m]})_{p,ij}^{-2}}{\omega(\sigma^{[m]})_{q,ij}^{-2} + (1-\omega)(\sigma^{[m]})_{p,ij}^{-2}}, \quad (\tilde{\sigma}^{[m]})_{p,ij}^2 := \frac{1}{\omega(\sigma^{[m]})_{q,ij}^{-2} + (1-\omega)(\sigma^{[m]})_{p,ij}^{-2}}. \tag{7}$$

- Case 2, in the binary case, let $p_\theta(z_{ij}^{[m]}|\bar{z}_{ij}) \sim \text{Ber}(\delta_{p,ij}^{[m]})$, where $\delta_{p,ij}^{[m]} := g_{\bar{z}\to z}(\bar{z}_{ij})$. With $\delta_{q,ij}^{[m]}$ obtained in (5), the weighted adjustment gives

$$p_\theta(z_{ij}^{[m]}|\cdot) \sim \text{Ber}\Big(\tilde{\delta}_{p,ij}^{[m]}\Big); \quad \tilde{\delta}_{p,ij}^{[m]} = \frac{1}{\omega/\delta_{q,ij}^{[m]} + (1-\omega)/\delta_{p,ij}^{[m]}}. \tag{8}$$

Similar to the function $f_{z\to\bar{z}}$ in the encoder, here $g_{\bar{z}\to z}$ corresponds to $f_{z\to\bar{z}}$'s "reverse-direction" counterpart and its choice can be rather simple. Further, ω governs the mixing percentage of the entity-specific and the common information: when $\omega = 1$, the "tilde" parameters of the post-adjustment distribution effectively collapse into the encoder ones (e.g., $\tilde{\delta}_{p,ij} \equiv \delta_{q,ij}^{[m]}$ and analogously for $\tilde{\mu}_{p,ij}, \tilde{\sigma}_{p,ij}^2$); correspondingly, samples drawn from $p_\theta(z_{ij}^{[m]}|\cdot)$ essentially ignore the sampled $\bar{\mathbf{z}}$ and hence they can be viewed as entirely entity-specific. At the other extreme, for $\omega = 0$, the tilde parameters coincide with the unadjusted ones; therefore, apart from the grand common information carried in the sampled $\bar{\mathbf{z}}$, no entity-specific one is passed onto the sampled $\mathbf{z}^{[m]}$. By toggling ω between $(0, 1)$, one effectively controls the level of heterogeneity and how strongly the sampled entity-specific graphs deviate from the grand common one.

Graph2Trajectory. Module (dec-b) pertains to modeling the dynamics of the trajectory $\mathbb{x}^{[m]}$ given the sampled $\mathbf{z}^{[m]}$. Here, we focus on one-step Markovian dependency, i.e., $q = 1$ and thus $p_\theta(\mathbf{x}_t^{[m]} \,|\, \mathbf{x}_{t-1}^{[m]}, ..., \mathbf{x}_1^{[m]}, \mathbf{z}^{[m]}) \approx g_{z\to x}(\mathbf{x}_{t-1}^{[m]}, \mathbf{z}^{[m]})$. The extension to longer lag dependencies ($q > 1$) can be readily obtained by pre-processing the input accordingly, as discussed in Appendix A.2.

We consider the following parameterization of $g_{z\to x}$. At the high level, given that $z_{ij}^{[m]}$ corresponds to the Granger-causal connection from node j to node i, it should serve as a "gate" controlling the amount of information that can be passed from $x_{j,t-1}^{[m]}$ to $x_{i,t}^{[m]}$. To this end, each response coordinate $x_{i,t}^{[m]}$ is modeled as follows:

$$u_{i,t-1}^{[m],j} := \check{x}_{j,t-1}^{[m]} \circ z_{ij}^{[m]} \;\; \text{(gating)}, \quad \mathbf{u}_{i,t-1}^{[m]} = \{u_{i,t-1}^{[m],1}, \cdots, u_{i,t-1}^{[m],p}\}, \quad \text{and} \quad \check{\mathbf{u}}_{i,t-1}^{[m]} := \text{MLP}(\mathbf{u}_{i,t-1}^{[m]}); \tag{9}$$

$$x_{i,t}^{[m]} \sim \mathcal{N}\big(\mu_{x,it}^{[m]}, (\sigma^{[m]})_{x,it}^2\big), \quad \text{where} \quad \mu_{x,it}^{[m]} := \text{Linear}(\check{\mathbf{u}}_{i,t-1}^{[m]}), \;\; (\sigma^{[m]})_{x,it}^2 = \text{Softplus}\big(\text{Linear}(\check{\mathbf{u}}_{i,t-1}^{[m]})\big). \tag{10}$$

Note that in the gating operation in (9), we use $\check{x}_{j,t-1}^{[m]}$ to denote the output after some potential numerical embedding step (e.g., Gorishniy et al. (2022)) of $x_{j,t-1}^{[m]}$; in the absence of such embedding, $\check{x}_{j,t-1}^{[m]} \equiv x_{j,t-1}^{[m]}$. Through the gating step[1], $x_{j,t-1}^{[m]}$ exerts its impact on $x_{i,t}^{[m]}$ entirely through $u_{i,t-1}^{[m],j}$. The continuous case and the binary case $z_{ij}^{[m]}$ can be treated in a unified manner: in the former case, the value of $z_{ij}^{[m]}$ corresponds to

[1]Note that $z_{ij}^{[m]}$ is a scalar and is applied to all coordinates of $\check{x}_{j,t-1}^{[m]}$ in the case the latter is a vector.

the strength; in the latter case, it performs masking. Subsequently, $\mathbf{u}_{i,t-1}^{[m]}$ collects the $u_{i,t-1}^{[m],j}$'s of all nodes $j = 1, \cdots, p$, and serves as the predictor for $x_{i,t}^{[m]}$. Finally, if one simply sums all $u_{i,t-1}^{[m],j}$'s to obtain the mean of $x_{i,t}^{[m]}$, then it effectively coincides with the operation in a linear VAR system, with $z_{ij}^{[m]}$ corresponding precisely to the entries in the transition matrix.

Remark 2. The above-mentioned choice of $g_{z\to x}$ can be viewed as a "node-centric" one, wherein entries $z_{ij}^{[m]}$ control the information passing directly through the nodes. As an alternative, one can consider an "edge-centric" one, which leverages the idea of message-passing in GNNs and entails "node2edge" and "edge2node" operations. This resembles the technology adopted in Kipf et al. (2018); Löwe et al. (2022) that consider primarily having graph entries corresponding to categorical edge types, which, after some adaptation, can be used to handle the numerical case. In practice, we observe that the edge-centric graph2trajectory decoder can lead to instability for time series signals[2]. A more detailed comparison can be found in Appendix A.2, where additional illustrations are provided for the two.

3.2.3 Sampling

Given the stochastic nature of the sampled quantities, drawing samples from the encoded/decoded distributions requires special handling to enable the gradient to back propagate. Depending on whether entries of $\mathbf{z}^{[m]}$ are continuous or binary, there are three possible types of distributions involved; for notational simplicity, here we use z to represent generically the random variable under consideration.

- Normal $z \sim \mathcal{N}(\mu, \sigma^2)$. In this case, the "standard" reparameterization trick (Kingma & Welling, 2014) can be used, that is, $z = \mu + \sigma \circ \epsilon$, $\epsilon \sim \mathcal{N}(0, 1)$.
- Bernoulli $z \sim \text{Ber}(\delta)$. In this case, the discrete distribution is approximated by its continuous relaxation Maddison et al. (2017). Concretely, $z = \text{softmax}((\log(\boldsymbol{\pi}) + \boldsymbol{\epsilon})/\tau)$ where $\boldsymbol{\epsilon} \in \mathbb{R}^2$ whose coordinates are i.i.d. samples from Gumbel$(0, 1)$, $\boldsymbol{\pi} = (1 - \delta, \delta)$ is the binary class probability and τ is the temperature.
- Beta $z \sim \text{Beta}(\alpha, \beta)$. In this case, implicit reparameterization of the gradients (Figurnov et al., 2018) is leveraged and the construction of the reparameterized samples becomes much more involved. We refer interested readers to Figurnov et al. (2018); Jankowiak & Obermeyer (2018) for an in-depth discussion on how parameterized random variables can be obtained and become differentiable.

3.2.4 Loss Function

The loss function is given by the negative ELBO, that is,[3]

$$-\mathbb{E}_{q_\phi(\mathcal{Z}|\mathcal{X})}\Big(\log p_\theta(\mathcal{X}|\mathcal{Z})\Big) + \text{KL}\Big(q_\phi(\mathcal{Z}|\mathcal{X}) \,\Big\|\, p_\theta(\mathcal{Z})\Big) =: \text{reconstruction error} + \text{KL};$$

the first term corresponds to the reconstruction error that measures the deviation between the original trajectories and the reconstructed ones, while the KL term measures the "consistency" between the encoded and the decoded distributions, and can be viewed as a type of regularization.

Let $\boldsymbol{\mu}_{x,t}^{[m]} := (\mu_{x,1t}^{[m]}, \cdots, \mu_{x,pt}^{[m]})^\top$ and $\Sigma_{\mathbf{x}_t^{[m]}} := \text{diag}((\sigma^{[m]})_{x,1t}^2, \cdots, (\sigma^{[m]})_{x,pt}^2)^\top$ with the components defined in (10). The reconstruction error is the negative Gaussian log-likelihood loss given by

$$\sum_{m=1}^{M}\Big(\sum_{t=2}^{T}\big(\mathbf{x}_t^{[m]} - \boldsymbol{\mu}_{x,t}^{[m]}\big)^\top \Sigma_{\mathbf{x}_t^{[m]}}^{-1}\big(\mathbf{x}_t^{[m]} - \boldsymbol{\mu}_{x,t}^{[m]}\big) + \log|\Sigma_{\mathbf{x}_t^{[m]}}|\Big). \tag{11}$$

The KL term can be simplified after some algebra to (see Appendix A.3 calculation):

$$\mathbb{E}_{q_\phi(\mathcal{Z}|\mathcal{X})}\Big[\text{KL}\Big(q_\phi(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}) \,\Big\|\, p_\theta(\bar{\mathbf{z}})\Big)\Big] + \mathbb{E}_{q_\phi(\mathcal{Z}|\mathcal{X})}\Big[\text{KL}\Big(q_\phi(\{\mathbf{z}^{[m]}\}|\{\mathbf{x}^{[m]}\}) \,\Big\|\, p_\theta(\{\mathbf{z}^{[m]}\}|\bar{\mathbf{z}})\Big)\Big]; \tag{12}$$

[2] to contrast with the physical system (e.g., Springs) considered in the experiments of Kipf et al. (2018).

[3] Recall that $\mathcal{X} := \{\mathbf{x}^{[m]}; m = 1, \cdots, M\}$ and $\mathcal{Z} := \{\bar{\mathbf{z}}, \mathbf{z}^{[m]}; m = 1, \cdots, M\}$.

both terms can be viewed as "consistency matching" terms that measure the divergence between the distributions obtained in the encoder pass and that from the decoder pass. Finally, note that in the implementation, the quantities involved are replaced by their conjugacy adjusted counterparts wherever applicable, and this is similar to the treatment in Sønderby et al. (2016).

3.3 Training and Inference

The functions in the encoder ($f_{x\to h}$, $f_{h\to z}$ and $f_{z\to\bar{z}}$) and those in the decoder ($g_{\bar{z}\to z}$ and $g_{z\to x}$) are shared across all entities $m = 1, \cdots, M$, and thus the model is trained based on the "pooled" data of all entities, while keeping track of the entity id that each data block is associated with. The steps involved in the end-to-end training under the proposed framework are summarized in Exhibit 1.

Exhibit 1: Outline of steps for training under the two-layer VAE-based framework

Input: observed trajectories $\{\mathbb{x}^{[1]}, \cdots, \mathbb{x}^{[M]}\}$, hyperparameters
– **Forward pass, encoder:** $\{\mathbb{x}^{[m]}\} \to \{\mathbf{z}^{[m]}\} \to \bar{\mathbf{z}}$
0. $m = \{1, \cdots, M\}$: obtain the encoded distribution for entity-specific graphs $q_\phi(\mathbf{z}^{[m]}|\mathbb{x}^{[m]})$;
1. $m = \{1, \cdots, M\}$: sample $\mathbf{z}^{[m]}$ from $q_\phi(\mathbf{z}^{[m]}|\mathbb{x}^{[m]})$;
2. based on $\{\mathbf{z}^{[m]}\}_{m=1}^{M}$, obtain the encoded distribution for the common graph $q_\phi(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\})$;

– **Forward pass, decoder:** $\bar{\mathbf{z}} \to \{\mathbf{z}^{[m]}\} \to \{\mathbb{x}^m\}$
3. merge prior info $p_\theta(\bar{\mathbf{z}})$ into $q_\phi(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\})$ then sample $\bar{\mathbf{z}}$;
4. $m = \{1, \cdots, M\}$: obtain the decoded distribution for entity-specific graphs $p_\theta(\mathbf{z}^{[m]}|\bar{\mathbf{z}})$;
5. $m = \{1, \cdots, M\}$: merge entity-specific encoded info $q_\phi(\mathbf{z}^{[m]}|\mathbb{x}^{[m]})$ into $p_\theta(\mathbf{z}^{[m]}|\bar{\mathbf{z}})$, then sample $(\mathbf{z}^{[m]}|\cdot)$;
6. $m = \{1, \cdots, M\}$: using $\mathbf{z}^{[m]}$ and the lag info $\mathbf{x}_{t-1}^{[m]}$, decode to get $\hat{\mathbf{x}}_t^{[m]}$; $t = 2, \cdots, T$.

– **Loss calculation**
7. calculate the EBLO loss by summing up (11) and (12);

– **Backward pass**: update neural network parameters based on gradients (back-propagation)
Output: Trained encoder and decoder

Several pertinent remarks follow. (1) The data typically consist of "long" trajectories that contain all the available observations (time points); one needs to partition them to "short" ones of length T (that are typically between 20-50), which constitute the samples used in model training. (2) In the case where one has external information regarding presence or absence of edges in the $\mathbf{z}^{[m]}$'s, it can be incorporated by enforcing the corresponding entries to zero after the former are sampled in Step 5. (3) Once the encoder (inference model) and the decoder (generative model) are trained, the latent graphs can be obtained by applying the trained encoder on the trajectories. For entity-specific graphs $\mathbf{z}^{[m]}$'s, the inference model gives the encoded distribution $q_\phi(\mathbf{z}^{[m]}|\mathbf{x}^{[m]})$'s. In practice, the graph of interest is extracted by calculating the "mode" of the distribution; the grand common graph $\bar{\mathbf{z}}$ can be analogously handled. It is worth noting that for continuous $\mathbf{z}^{[m]}$'s, the proposed framework naturally provides signed estimates and thus positive/negative Granger causal connections can be readily differentiated. (4) The trained decoder can be utilized to quantify also the *predictive* strength of the Granger-causal connection, as discussed in Appendix A.4.

4 Synthetic Data Experiments

We evaluate the performance of the proposed framework, together with benchmarking methods on several synthetic data settings. For all experiments, we start from a common graph that corresponds to $\bar{\mathbf{z}}$, add perturbations to it for individual entities to produce heterogeneous Granger-causal connections (i.e., the $\mathbf{z}^{[m]}$'s), then simulate trajectories $\{\mathbb{x}^{[m]}\}$ corresponding to each entity based on their respective $\mathbf{z}^{[m]}$'s and the specified dynamics. The estimated entity-specific and grand common graphs are then evaluated against the underlying truth, for both the proposed and competing methods.

Supervised type competitors[4] include `NGC` (Tank et al., 2021), `GVAR` (Marcinkevičs & Vogt, 2021) and `TCDF` (Nauta et al., 2019), and a regularized linear VAR model based estimator (`Linear`; e.g., Basu & Michailidis (2015)). For generative model-based ones, we consider variations of Löwe et al. (2022). Note that the original paper and the accompanying code implementation only handles the case where each entry in the latent graph is a categorical variable denoting the "edge type". Consequently, we adapt the method and make necessary modifications to the code, so that it can handle numerical values[5]. Besides using the edge-centric graph2trajectory decoder adopted in Kipf et al. (2018); Löwe et al. (2022), we also consider another variant based on the proposed node-centric one. These two benchmarks are referred to as `One-Edge` and `One-Node`. Note that none of the above-mentioned methods readily handles the multi-entity setting where all graphs are estimated jointly; hence, for comparison purposes, the estimated grand common graph for the competitors is simply obtained by averaging the estimated entity ones.

4.1 Data generating mechanisms

The data generating mechanisms used are based on: (1) a linear VAR, (2) a non-linear VAR, and (3) multi-species Lotka-Volterra systems. Two additional mechanisms corresponding to the Lorenz96 and the Springs systems are also considered; their description and results are presented in Appendix B. Consistent with extant notation, p denotes the number of nodes and M the number of entities.

Linear VAR. The dynamics of a linear VAR(1) model are determined by $\mathbf{x}_t = A\mathbf{x}_{t-1} + \boldsymbol{\varepsilon}_t$, $\mathbf{x}_t \in \mathbb{R}^p$, wherein $A \in \mathbb{R}^{p\times p}$ is the transition matrix and coincides with the Granger-causal graph; for notational convenience, let $\bar{A} := \bar{\mathbf{z}}$ denote the grand common and $A^{[m]} := \mathbf{z}^{[m]}$ the entity-specific graphs. For this mechanism, we set $p = 30$ and $M = 20$, while the noise term $\mathbf{e}_t$ has i.i.d entries drawn from a standard Gaussian distribution.

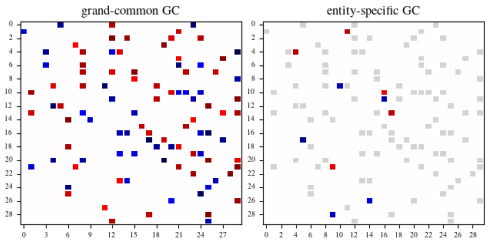


Figure 3: linear VAR: $\bar{A}$ and $A^{[1]}$; red:(+); blue:(−).

We first discuss generation of $\bar{A}$, whose skeleton $\mathcal{S}_{\bar{A}}$ (i.e., support set) is determined by independent draws from Ber(0.1); nonzero entries are first drawn from $\text{Unif}(-2,-1)\cup(1,2)$, then scaled so that the spectral radius (i.e., the maximum in absolute value eigenvalue) of $\bar{A}$ is 0.5. Next, we generate perturbations of $\bar{A}$ by "relocating" 10% of the entries (denote their index set by $\mathcal{S}_{\text{ptrb}}$) in $\mathcal{S}_{\bar{A}}$ to random locations in the non-support set $\mathcal{S}^c_{\bar{A}}$. This step generates the corresponding $A^{[m]}$'s. Note that the perturbation mechanism ensures that $\mathcal{S}_{\text{ptrb}} \subset \mathcal{S}_{\bar{A}}$. Further, the positions of the 10% of entries selected at random remain fixed for all M entities, and only the "new" locations are randomly selected and hence differ across the entities, thus inducing heterogeneity across the $A^{[m]}$'s. As a result of the perturbation, entries in $\mathcal{S}_{\text{ptrb}}$ are essentially "flipped" to zero and therefore the final grand common graph changes accordingly; see also Figure 3.[6]

Non-linear VAR. For this mechanism, we set $p = 20$ and $M = 10$. We first describe how $\bar{\mathbf{z}}$ and $\mathbf{z}^{[m]}$ are generated, as they dictate the connections and determine how the dynamics are specified. First, let $\bar{\mathbf{z}}^{(0)}$ be the "initial" common graph, set to a banded matrix that has non-zero entries on the diagonal and the adjacent upper and lower diagonals. Next, we perturb $\bar{\mathbf{z}}^{(0)}$ as follows: for all rows not divisible by 3 (e.g., rows, 1, 2, 4, etc.), the two off-diagonal entries are relocated to other positions at random within the same row. This is repeated for

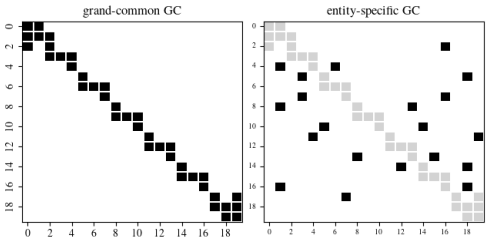


Figure 4: non-linear VAR: $\bar{\mathbf{z}}$ and $\mathbf{z}^{[1]}$, showing only the skeleton.

[4]The selection of these competitors is based on the results reported in Marcinkevičs & Vogt (2021). Specifically, we picked the ones that were demonstrated to be competitive. The code implementations for these competitors (except for the regularized Linear VAR) are directly taken from the repositories accompanying the papers.

[5]see Appendix A.2 for how the adaptation can be conducted.

[6]For illustration purposes, we show the grand-common and the entity-specific (using Entity 1 as an example) Granger-causal connection matrices. In the entity-specific one, nonzero entries that overlap with those in the common structure is grayed out for more distinctive visual on the idiosyncratic connections.

all m's to generate $\mathbf{z}^{[m]}$'s. The perturbation creates a zigzag pattern for the final $\bar{\mathbf{z}}$, since whenever a perturbation is present, the original off-diagonal entries on the ± 1 band are guaranteed to get flipped to zero – see Figure 4 for an illustration[6]. Within any entity m, response nodes indexed by $i = 2, \cdots, p-1$ have 3 parents; denote their indices by $k_i^1 < k_i^2 < k_i^3$ with subscript i corresponding to the response node id and superscript the parent id, and $k_i^2 \equiv i$ by construction.

The trajectories are generated as follows. For $i = 2, \cdots, p-1$, let $x_{i,t} = 0.25 x_{i,t-1} + \sin(x_{k_i^1,t-1} \cdot x_{k_i^3,t-1}) + \cos(x_{k_i^1,t-1} + x_{k_i^3,t-1}) + \varepsilon_{i,t}$, $\varepsilon_{i,t} \sim \mathcal{N}(0, 0.25)$. For the first node i and the last node p, their dynamics are slightly different given that they only have one "neighbor"[7]. The choice of such dynamics (in particular, using sine/cosine functions) is somewhat ad-hoc, but aim to induce non-linearities, while ensuring that the system is stable given that these functions are unformly bounded. Finally, note that we omit the superscript $[m]$ that indexes the entities, as the dynamic specification applies to the dynamical systems of all entities; the parent set for each response node i of entity m is dictated by row i of $\mathbf{z}^{[m]}$.

Multi-species Lotka-Volterra system. It comprises of coupled ordinary different equations (ODE) that model the population dynamics of multiple predators and preys based on their interactions, specified by the corresponding Granger causal graphs. We consider $p = 20$ and $M = 10$. The p nodes are separated equally into preys and predators (i.e., $\frac{p}{2}$ preys and predators each). Let $\mathbf{x}_t := (\mathbf{u}_t^\top, \mathbf{v}_t^\top)^\top$ with $\mathbf{u}_t \in \mathbb{R}^{p/2}$ and $\mathbf{v}_t \in \mathbb{R}^{p/2}$ denoting the population size of the preys and the predators at time t, respectively; $\mathbb{u}_i := \{\mathbf{u}_{i,t}\}$ corresponds to the continuous-time trajectory for the ith coordinate and $\mathbb{v}_j$ is analogously defined. The dynamics for each coordinate are specified through the following ODE system:

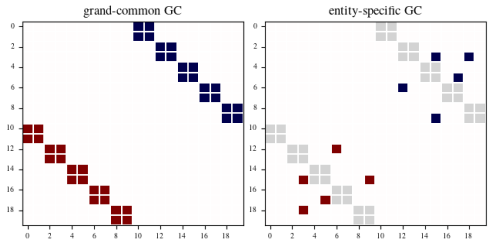


Figure 5: Lokta-Volterra: $\bar{\mathbf{z}}$ and $\mathbf{z}^{[1]}$, showing the signed skeleton. Red:(+), blue:(−).

$$\frac{\mathrm{d}\mathbb{u}_i}{\mathrm{d}t} = \alpha \mathbb{u}_i - \beta \mathbb{u}_i \Big(\sum_{j \in \mathcal{P}_i} \mathbb{v}_j\Big) - \alpha (\mathbb{u}_i/\eta)^2; \qquad \frac{\mathrm{d}\mathbb{v}_j}{\mathrm{d}t} = \delta \mathbb{v}_j \Big(\sum_{i \in \mathcal{P}_j} \mathbb{u}_i\Big) - \gamma \mathbb{v}_j. \tag{13}$$

The parameters are set to $\alpha = 1.1$, $\beta = 0.2$, $\gamma = 1.1$, $\delta = 0.2$ and $\eta = 200$. Once again, we omit superscript $[m]$ as this specification applies to all $m = 1, \cdots, M$. The heterogeneity at the entity level is contingent on their graphs $\mathbf{z}^{[m]}$'s that dictate the coupling mechanism; in particular, $\mathcal{P}_i$ and $\mathcal{P}_j$ are the parent set of nodes i and j, and are respectively dictated by the support set of the ith and jth rows of the corresponding $\mathbf{z}^{[m]}$. The generation mechanism of $\bar{\mathbf{z}}$ and $\mathbf{z}^{[m]}$ are described next. The common graph $\bar{\mathbf{z}}$ is generated identically to the one considered in Marcinkevičs & Vogt (2021), where the 20 nodes can be separated into 5 decoupled systems, each containing 2 predators and 2 preys. We add random perturbations to $\bar{\mathbf{z}}$ to arrive at the $\mathbf{z}^{[m]}$'s, by adding additional entries. These additional entries in the upper right/lower left blocks need to be symmetric w.r.t. the diagonal so that the predator-prey correspondence is respected, and they also provide coupling across the originally decoupled 5×4 systems – see also Figure 5 for an illustration.[6]

4.2 Performance evaluation

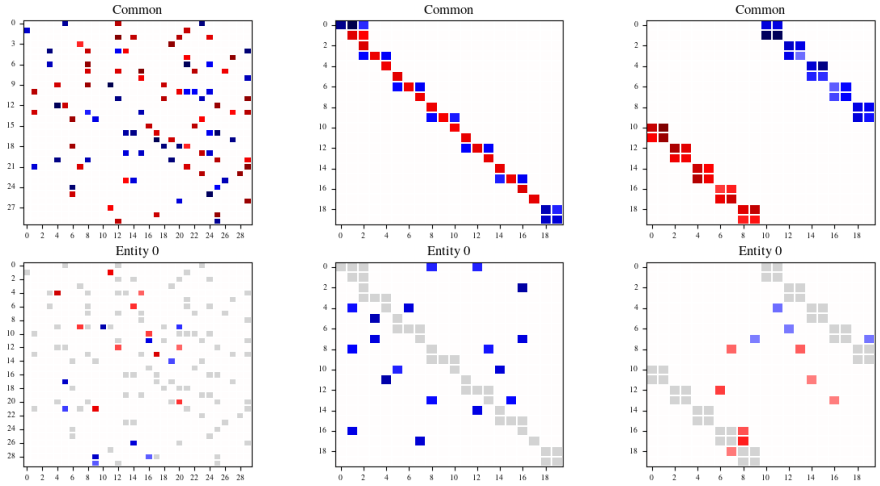
For all settings, we consider sample sizes of 10K. We run 5 data replicates and report the mean and standard deviation of the AUROC and AUPRC metrics for the competing methods considered. Given that the underlying true Granger-causal graphs in the examined settings are sparse, we also report the best attainable F1 score for each method after thresholding the entries of the group and entity-specific graphs. Results for two other experimental settings, –the Lorenz96 and the Springs systems–, are presented in Appendix B.1. Additional metrics such as true positive rate (TPR), true negative rate (TNR) and accuracy (ACC) based on different thresholding levels are deferred to Appendix B.2, together with visual illustrations of the estimates obtained by good performing competitors.

[7]For $i = 1$, the dynamics is given by $x_{1,t} = 0.4 x_{1,t-1} - 0.5 x_{2,t-1} + \varepsilon_{1,t}$; for $i = p$, the dynamics is given by $x_{p,t} = 0.4 x_{p,t-1} - 0.5 x_{p-1,t-1} + \varepsilon_{p,t}$

Table 1 displays the results for all methods. The proposed framework is referred to as `Multi-node` and `Multi-edge`, corresponding to the multi-entity joint learning approaches using the node- and edge-centric decoders, respectively; a visualization of the estimated $\bar{\mathbf{z}}$ and $\mathbf{z}^{[1]}$ for illustration purposes is provided in Figure 6 for the former.

Table 1: Performance evaluation for the estimated $\bar{\mathbf{z}}$ and $\mathbf{z}^{[m]}$'s: "common" corresponds to $\bar{\mathbf{z}}$ and "entity(avg)" the $\mathbf{z}^{[m]}$'s after averaging the performance metric across $m = 1, \cdots, M$. Numbers are in % and rounded to integers, and correspond to the mean results based on 5 data replicates; standard deviations are reported in the parenthesis.

		Generative model-based				Supervised model-based			
		Multi-node	Multi-edge	One-node	One-edge	NGC-cMLP	GVAR	TCDF	Linear
Linear VAR									
common	AUROC	100(0.0)	100(0.0)	95(6.6)	98(4.8)	100(0.4)	100(0.0)	79(2.0)	100(0.0)
	AUPRC	100(0.0)	100(0.0)	83(20.4)	91(15.9)	99(1.3)	100(0.0)	50(7.6)	100(0.0)
	F1(best)	100(0.0)	100(0.0)	81(17.4)	88(15.9)	96(3.5)	100(0.0)	52(5.1)	100(0.0)
entity (avg)	AUROC	100(0.1)	99(0.6)	100(0.1)	100(0.1)	96(1.8)	100(0.0)	77(1.4)	100(0.0)
	AUPRC	99(0.3)	95(2.4)	99(0.2)	98(0.4)	86(4.4)	99(0.1)	36(5.5)	100(0.0)
	F1(best)	97(0.8)	90(3.5)	96(0.6)	95(1.0)	79(4.7)	99(0.4)	44(3.4)	100(0.0)
Non-linear VAR									
common	AUROC	99(0.2)	82(1.7)	97(0.2)	93(0.8)	90(0.7)	99(0.1)	75(1.0)	99(0.1)
	AUPRC	96(0.9)	58(1.1)	80(0.8)	80(8.0)	64(1.1)	98(0.2)	53(0.5)	98(0.1)
	F1(best)	94(0.6)	60(0.7)	74(1.0)	83(6.9)	61(0.9)	98(0.7)	56(1.2)	98(0.7)
entity (avg)	AUROC	98(0.3)	85(0.9)	94(0.4)	95(0.5)	94(0.5)	99(0.3)	73(0.9)	96(0.7)
	AUPRC	93(1.0)	75(0.8)	76(0.2)	89(0.6)	87(0.6)	96(0.6)	44(1.8)	96(0.7)
	F1(best)	86(1.5)	73(1.0)	70(0.3)	86(0.8)	82(0.4)	91(0.8)	50(1.5)	97(0.6)
Lotka-Volterra									
common	AUROC	100(0.0)	100(0.0)	97(1.1)	87(8.4)	100(0.0)	100(0.0)	79(0.8)	100(0.1)
	AUPRC	100(0.0)	100(0.1)	92(3.0)	73(10.5)	100(0.0)	100(0.0)	58(1.2)	100(0.4)
	F1(best)	100(0.7)	99(0.8)	87(5.4)	69(9.0)	100(0.4)	97(1.2)	53(1.4)	94(3.5)
entity (avg)	AUROC	89(1.0)	84(1.3)	83(1.6)	75(1.3)	92(1.0)	93(0.6)	72(0.8)	77(1.0)
	AUPRC	80(1.5)	70(2.0)	69(1.8)	51(2.6)	87(1.2)	89(1.0)	41(1.0)	71(1.2)
	F1(best)	74(1.4)	65(2.0)	63(1.4)	53(2.2)	84(0.8)	84(0.7)	46(0.3)	71(0.7)



(a) Linear VAR (b) Non-linear VAR (c) Lotka-Volterra

Figure 6: Estimated Granger-causal connections using the proposed framework with node-centric decoder (`Multi-node`). Top row shows the estimated $\bar{\mathbf{z}}$ and bottom row shows the estimated $\mathbf{z}^{[1]}$ (as an example). Nonzero entries in $\hat{\mathbf{z}}^{[1]}$ that overlap with those in $\hat{\bar{\mathbf{z}}}$ have been grayed-out so that the idiosyncratic ones stands out. This is based on the same data seed as the ones shown in Figures 3,4,5.

The main findings are as follows: (1) the proposed joint-learning approach clearly outperforms its individual learning counterpart (e.g., `Multi-node` vs. `One-node`), both at the entity level and the group level (i.e., the common graph). The performance is overall on-par with `GVAR`, which is the strongest overall competitor. (2) The node-centric decoder consistently outperforms its edge-centric counterpart (e.g., `Multi-node` vs. `Multi-edge`). (3) If one focuses only on individual learning methods, the ones based on supervised models

tend to exhibit superior performance (e.g., `GVAR`/`NGC` vs. `One-node`). In addition, despite the presence of non-linear dynamics, the regularized linear VAR model exhibits surprisingly good performance, especially for the common structure. (4) For practical purposes, post-hoc averaging of the entity-specific Granger causal graphs is reasonably effective for extracting the common structure.

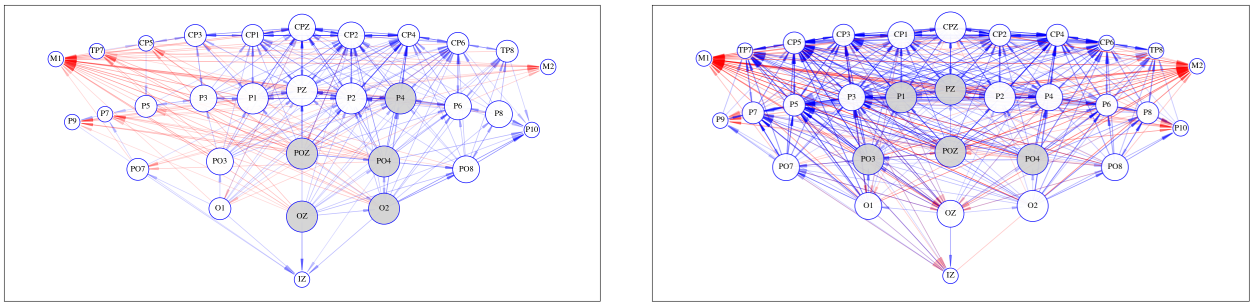
Finally, we remark that despite that `GVAR` exhibits strong performance (as measured primarily by AUROC in Table 1) amongst the methods under consideration, it is observed during evaluation time that given the magnitude of the estimated entries, the quality of the graph skeleton is sensitive to the exact choice of the thresholding level, whereas the proposed framework is more robust. This has implications on the difficulty of choosing a good threshold in practice — see also Table 4 and additional discussion and remarks in Appendix B.2.

5 Application to a Multi-Subject EEG Dataset

The dataset in consideration corresponds to electroencephalogram (EEG) measurements obtained from 72 active electrodes placed on the scalp of 22 subjects (entities), and they are publicly available; see Trujillo et al. (2017). Prior investigation on this dataset primarily centers around understanding the information provided by different connectivity measures that are available in the literature, rather than the connectivity patterns themselves.

The EEG experiment pertains to a stimulus procedure performed on the subjects comprising of 1-min interleaved sessions with eyes open (EO) or closed (EC). Such experiments aim to provide insights into the brain's functional segregation and integration (Barry et al., 2007; Rubinov & Sporns, 2010; Miraglia et al., 2016). Note that the experiment is integrated, but the data are collated separately for the eyes-open and the eyes-closed interleaving sessions, which results in two data sets (EO and EC, respectively); they are then analyzed separately using the joint-learning model (`multi-node`). Further, note that due to the design of the experiment, the dynamics governing the data within the EO sessions (respectively, EC sessions) are stable and stay largely unchanged.

We select to analyze the data from 31 specific EEG channels (and hence $p = 31$) located at the back of the scalp (see Figure 7), where the primary visual cortex is located. For both datasets, we restrict the analysis to entities that have at least 40000 observations (total number of time points)[8], and the whole trajectory is further partitioned into training/validation data, with the latter having 2000 time points. Here the validation data is used to select the best hyperparameters such that the reconstruction error is minimum over the search grid.



(a) Eyes Open (EO)

(b) Eyes Closed (EC)

Figure 7: Estimated **common** Granger-causal connections for EO (left panel) and EC (right panel) after normalization and subsequent thresholding at 0.15. Red edges correspond to positive connections and blue edges correspond to negative ones; the transparency of the edges is proportional to the strength of the connection. Larger node sizes correspond to higher connectivity levels (incoming), and the top 5 nodes are colored in gray.

The estimated common Granger-causal connections are depicted in Figure 7, and the patterns resonate with findings in the literature based on previous EEG studies on similar experiments: (1) the overall Granger

[8]this restriction has reduced the number of entities to 21 for the EO dataset while the number of entities for the EC dataset remains at 22.

causal connectivity is markedly higher for the EC session compared to the EO session; see also Barry et al. (2007); Marx et al. (2004); Das et al. (2016); Trujillo et al. (2017), albeit using different connectivity measures. (2) The OZ channel exhibits different connections for the EO and the EC sessions; in particular, it is Granger causal (i.e., being the emitter of edges) for many other channels in the former and becomes significantly less so in the latter; see also Hatton et al. (2023).

Finally, an interesting finding that merits further investigation is that the highly connected nodes (shaded in gray) are located in the center and on the right side of the scalp in the EO sessions, whereas those in EC are more symmetrically distributed across the two sides.

6 Discussion

This paper proposes a multi-layer VAE-based framework for jointly estimating the group and entity-level Granger-causal graphs, in the presence of connectivity heterogeneity across entities. The framework is based on a hierarchical generative structure that couples the group and entity-specific graphs. The model is learned via an end-to-end encoding-decoding procedure that minimizes the ELBO loss. The joint learning paradigm has a clear advantage over its "individual learning" generative model-based counterpart, which then leads to more accurate quantification for both the common connectivity patterns and the idiosyncratic ones. This advantage becomes more pronounced in settings where one has limited sample size and large collections of related systems. In addition, the joint learning paradigm can be useful in situations, where one may be interested in detecting "outlier" dynamical systems in the collection under consideration, or in identifying clusters of such systems. These tasks can be accomplished by close examination and analysis of the entity specific graphs.

Although "supervised models plus post-hoc aggregation" heuristics can sometimes exhibit competitive performance, the embedded common structure across entities is completely neglected at the formulation level. In addition, existing models within this framework are also limited to scalar-valued nodes, partly due to their reliance on performing ad-hoc extraction/aggregation on intermediate quantities (e.g., neural network weight matrices during training) to infer the Granger causality.

In the presence of non-linearity, a key advantage of generative model-based approaches is that the Granger-causal relationships are solely encoded through the latent graph that serves as the gateway for information propagation. This provides a clean way to model relationships between connectivity patterns — either statically or dynamically. The setting considered in this work is a static one, and the type of such relationship manifests as a common-idiosyncratic one. A potential extension to the generative process under consideration, suitable for more complex real-world dynamical systems, is to allow for time-varying connectivity patterns. For example, Graber & Schwing (2020) extends the work in Kipf et al. (2018) to a dynamic setting. With appropriate modifications to the proposed approach, such as expanding the conditional relationship of the graphs dictated in (2) so that they also depend on their past, this modeling task can be handled in a straightforward manner.

Code and Data Availability

The code repository for this work, including the `PyTorch` implementation of the proposed joint-learning framework, its individual learning counterpart as well as the code and configuration files for synthetic data generation/experiments will be made publicly available at official publication time. The real dataset being analyzed is available at `https://dataverse.tdl.org/dataverse/rsed2017`, as provided by Trujillo et al. (2017).

References

Robert J Barry, Adam R Clarke, Stuart J Johnstone, Christopher A Magee, and Jacqueline A Rushby. EEG differences between eyes-closed and eyes-open resting conditions. *Clinical Neurophysiology*, 118(12): 2765–2773, 2007.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015. doi: 10.1214/15-AOS1315.

Sumanta Basu, Ali Shojaie, and George Michailidis. Network granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 16(1):417–453, 2015.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Rainer Dahlhaus and Michael Eichler. Causality and graphical models in time series analysis. *Oxford Statistical Science Series*, pp. 115–137, 2003.

Rig Das, Emanuele Maiorana, and Patrizio Campisi. EEG biometrics using visual stimuli: A longitudinal study. *IEEE Signal Processing Letters*, 23(3):341–345, 2016.

Michael Eichler. Graphical modelling of multivariate time series. *Probability Theory and Related Fields*, 153: 233–268, 2012.

Paul Erdős and Alfréd Rényi. On random graphs I. *Publ. math. debrecen*, 6(290-297):18, 1959.

Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *Advances in Neural Information Processing Systems*, 31, 2018.

John Geweke. Inference and causality in economic time series models. *Handbook of Econometrics*, 2:1101–1144, 1984.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International Conference on Machine Learning*, pp. 1263–1272. PMLR, 2017.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. *Advances in Neural Information Processing Systems*, 35:24991–25004, 2022.

Colin Graber and Alexander Schwing. Dynamic neural relational inference for forecasting trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1018–1019, 2020.

Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pp. 424–438, 1969.

Clive WJ Granger. Testing for causality: A personal viewpoint. *Journal of Economic Dynamics and Control*, 2:329–352, 1980.

Shelby L Hatton, Shubham Rathore, Ilya Vilinsky, and Annette Stowasser. Quantitative and qualitative representation of introductory and advanced EEG concepts: An exploration of different EEG setups. *Journal of Undergraduate Neuroscience Education*, 21(2):A142, 2023.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

Yongmiao Hong, Yanhui Liu, and Shouyang Wang. Granger causality in risk and detection of extreme risk spillover between financial markets. *Journal of Econometrics*, 150(2):271–287, 2009.

Martin Jankowiak and Fritz Obermeyer. Pathwise derivatives beyond the reparameterization trick. In *International Conference on Machine Learning*, pp. 2235–2244. PMLR, 2018.

John Kerin and Hans Engler. On the Lorenz'96 model and some generalizations. *Discrete and Continuous Dynamical Systems - B*, 27(2):769–797, 2022. ISSN 1531-3492. doi: 10.3934/dcdsb.2021064.

Saurabh Khanna and Vincent Y. F. Tan. Economy statistical recurrent units for inferring nonlinear granger causality. In *International Conference on Learning Representations*, 2020.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems*, 27, 2014.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697. PMLR, 2018.

Edward N Lorenz. Predictability: A problem partly solved. In *Proc. Seminar on Predictability*, volume 1. Reading, 1996.

Sindy Löwe, David Madras, Richard Zemel, and Max Welling. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, pp. 509–525. PMLR, 2022.

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

Ričards Marcinkevičs and Julia E Vogt. Interpretable models for Granger causality using self-explaining neural networks. In *International Conference on Learning Representations*, 2021.

Esther Marx, Angela Deutschländer, Thomas Stephan, Marianne Dieterich, Martin Wiesmann, and Thomas Brandt. Eyes open and eyes closed as rest conditions: impact on brain activation patterns. *Neuroimage*, 21(4):1818–1824, 2004.

Francesca Miraglia, Fabrizio Vecchio, Placido Bramanti, and Paolo Maria Rossini. EEG characteristics in "eyes-open" versus "eyes-closed" conditions: Small-world network architecture in healthy aging and age-related brain degeneration. *Clinical Neurophysiology*, 127(2):1261–1268, 2016.

Alessandro Montalto, Sebastiano Stramaglia, Luca Faes, Giovanni Tessitore, Roberto Prevete, and Daniele Marinazzo. Neural networks with non-uniform embedding and explicit validation phase to assess granger causality. *Neural Networks*, 71:159–171, 2015.

Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019.

Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Anil K Seth, Adam B Barrett, and Lionel Barnett. Granger causality analysis in neuroscience and neuroimaging. *Journal of Neuroscience*, 35(8):3293–3297, 2015.

Ali Shojaie and Emily B Fox. Granger causality: A review and recent advances. *Annual Review of Statistics and Its Application*, 9:289–319, 2022.

Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. *Advances in Neural Information Processing Systems*, 29, 2016.

Cornelis J Stam. Nonlinear dynamical analysis of EEG and MEG: review of an emerging field. *Clinical Neurophysiology*, 116(10):2266–2301, 2005.

James H Stock and Mark W Watson. Vector autoregressions. *Journal of Economic Perspectives*, 15(4): 101–115, 2001.

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B Fox. Neural Granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2021.

Logan T Trujillo, Candice T Stanfield, and Ruben D Vela. The effect of electroencephalogram (EEG) reference choice on information-theoretic measures of the complexity and integration of eeg signals. *Frontiers in Neuroscience*, 11:425, 2017.

Tailin Wu, Thomas Breuel, Michael Skuhersky, and Jan Kautz. Discovering nonlinear relations with minimum predictive information regularization. *arXiv preprint arXiv:2001.01885*, 2020.

A Additional modeling details

In this section, we provide a description for some additional modeling details. In Sections A.1 and A.2, we omit superscript $[m]$ that indexes the entities whenever there is no ambiguity, as the descriptions therein apply to all m's independently unless otherwise specified.

A.1 Encoder

We provide details for the encoder sub-module that is abstracted as $f_{x\to h}$, wherein based on the node trajectories, one obtains the hidden representations for the edges $\{\mathbf{h}_{ij}\} := f_{x\to h}(\mathbb{x})$; see also Section 3.2, module (enc-a).

As the most basic building blocks of message-passing operations, "node2edge" and "edge2node" operate based off a *complete* graph, and can be generically represented as:

$$e_{ij} \leftarrow \text{concat}(x_i, x_j) \quad \text{(node2edge)}; \qquad x_i \leftarrow \sum\nolimits_j e_{ij} \quad \text{(edge2node)},$$

with x_i denoting the node representation and e_{ij} the edge one. $f_{x\to h}$ is then parameterized through the L passes of such operations:

$$\begin{aligned}
\text{(init emb)}: \quad & \check{\mathbb{x}}_i^{(0)} \leftarrow \text{emb}(\mathbb{x}_i), \quad \forall i = 1, \cdots, p \\
\check{\mathbb{x}} \to \mathbf{e}: \quad & e_{ij}^{(l)} \leftarrow \text{MLP}\big(\text{node2edge}(\check{\mathbb{x}}_i^{(l-1)}, \check{\mathbb{x}}_j^{(l-1)})\big); \quad l = 1, \cdots, L \\
\mathbf{e} \to \check{\mathbb{x}}: \quad & \check{\mathbb{x}}_i^{(l-1)} \leftarrow \text{MLP}\big(\text{edge2node}(e_{ij}^{(l)}; j = 1, \cdots, p)\big); \quad l = 2, \cdots, L
\end{aligned}$$

Here $\mathbb{x}_i$ corresponds to the trajectory of node i over time, that is, $\mathbb{x}_i = (x_{i,1}, \cdots, x_{i,T})$ and the final hidden representation is given by $\mathbf{h}_{ij} := e_{ij}^{(L)}$, $i, j = 1, \cdots, p$.

Note that this is effectively the `MLPEncoder` used in Kipf et al. (2018) and the description is given here for the sake of completeness. We refer interested readers to Kipf et al. (2018) for some other encoders considered therein.

A.2 Decoder

We divide this subsection into 2 parts, that respectively (1) discuss how the structure adopted in a node-centric Graph2trajectory module described in Section 3.2 can readily accommodate the presence of dependence on more than 1 lags; and (2) provide a brief discussion on how the original edge-centric decoder adopted in Kipf et al. (2018); Löwe et al. (2022) can be revised to adapt to the case of a numerical graph, and compare it with the node-centric one, although architectural choices are not the focus of this paper.

Extension to multiple lag dependency. The extension of a node-centric decoder to accommodate the presence of more than 1 lags (i.e., $q > 1$) is straightforward, largely due to the fact that the node value at time $t-1$, denoted by $x_{j,t-1}$ is not limited to be scalar-valued in the first place. In the case of q-lag

dependency, one can simply replace $x_{j,t-1}$ by $\text{concat}(x_{j,t-1}, \cdots, x_{j,t-q})$ and proceed with the remainder of the operations as outlined in (9) and (10). In particular, with the presence of more lags, as an alternative to a (optional) numerical embedding step, one can instead consider 1D-CNN as a preprocessing module on the "new" $x_{j,t-1}$, before an element-wise gate represented by z_{ij} is applied to control the information flow.

Adaptation of the edge-centric decoder. The original edge-centric decoder adopted in Kipf et al. (2018) handles the case where each entry in z_{ij} corresponds to an edge type (categorical), and it entails the following operations:

1. node2edge for each time step, that is $e_{ij,t-1} := \text{concat}(x_{i,t-1}, x_{j,t-1})$ to arrive at the edge representation at time $t-1$;
2. for *each* edge type of interest, run $e_{ij,t-1}$'s through its corresponding edge type-specific function (e.g., MLP) to get the "enriched" representation $\check{e}_{ij,t-1}$;
3. aggregate the enriched edge representations back to nodes via an edge2node operation, giving rise to $\mathbf{v}_{i,t-1}$'s, $i = 1, \cdots, p$; $\mathbf{v}_{i,t-1}$ then serves as the predictor for time-t response $x_{i,t}$.

In order for the above module to accommodate the case of a numeric z_{ij}, the following simple modification to step 2 is introduced:

2' run $e_{ij,t-1}$'s through some function (e.g., MLP) to get the "enriched" representation $\check{e}_{ij,t-1}$, and further update it through a gating mechanism as dictated by z_{ij}, that is, $\check{e}_{ij,t-1} \leftarrow \check{e}_{ij,t-1} \circ z_{ij}$.

The information propagation path from node j to i can be represented as:

$$x_{j,t-1} \overset{\text{node2edge}}{\to} e_{ij,t-1} \overset{\text{MLP}}{\to} \check{e}_{ij,t-1} \overset{\text{gating}}{\to} \check{e}_{ij,t-1} \circ z_{ij} \overset{\text{edge2node}}{\to} \mathbf{v}_{i,t-1} \to x_{i,t};$$

one can easily verify that for $z_{ij} = 0$, there is no path from $x_{j,t-1}$ to $x_{i,t}$.

As a final remark, to contrast it with the node-centric decoder where the gating through z_{ij} directly operates on the node representation, the path is given by

$$x_{j,t-1} \overset{\text{emb}}{\to} \check{x}_{j,t-1} \overset{\text{gating}}{\to} \check{x}_{j,t-1} \circ z_{ij} \overset{\text{element of}}{\to} \mathbf{u}_{i,t-1} \to x_{i,t};$$

for the edge-centric decoder, entries in z_{ij} determine the lead-lag information passing from $j \to i$ via $e_{ij,t-1}$ and therefore is somewhat circumstantial.

A.3 Loss calculation

A derivation of (12) is given next.

$$\begin{aligned}
\text{KL}\Big(q_\phi(\mathcal{Z}|\mathcal{X}) \,\Big\|\, p_\theta(\mathcal{Z})\Big) &= \mathbb{E}_{q_\phi(\mathcal{Z}|\mathcal{X})} \log\Big[\frac{q_\phi(\mathcal{Z}|\mathcal{X})}{p_\theta(\mathcal{Z})}\Big] = \mathbb{E}_{q_\phi(\mathcal{Z}|\mathcal{X})}\Big[\log\frac{q_\phi(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\})}{p_\theta(\bar{\mathbf{z}})} + \log\frac{q_\phi(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\})}{p_\theta(\{\mathbf{z}^{[m]}\}|\bar{\mathbf{z}})}\Big] \\
&= \iint q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}\big) q_\phi\big(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}\big) \log\Big[\frac{q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}\big)}{p_\theta\big(\bar{\mathbf{z}}\big)}\Big] \mathrm{d}\bar{\mathbf{z}} \mathrm{d}\{\mathbf{z}^{[m]}\} \\
&\quad + \iint q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}\big) q_\phi\big(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}\big) \log\Big[\frac{q_\phi\big(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}\big)}{p_\theta\big(\{\mathbf{z}^{[m]}\}|\bar{\mathbf{z}}\big)}\Big] \mathrm{d}\bar{\mathbf{z}} \mathrm{d}\{\mathbf{z}^{[m]}\} \\
&= \int q_\phi(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}) \underbrace{\Big\{\int q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}\big) \log\Big[\frac{q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}\big)}{p_\theta\big(\bar{\mathbf{z}}\big)}\Big] \mathrm{d}\bar{\mathbf{z}}\Big\}}_{\text{KL}\left(q_\phi(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}) \,\middle\|\, p_\theta(\bar{\mathbf{z}})\right)} \mathrm{d}\{\mathbf{z}^{[m]}\} \\
&\quad + \int q_\phi\big(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}, \{\mathbb{x}^{[m]}\}\big) \underbrace{\Big\{\int q_\phi\big(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}\big) \log\Big[\frac{q_\phi\big(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}\big)}{p_\theta\big(\{\mathbf{z}^{[m]}\}|\bar{\mathbf{z}}\big)}\Big] \mathrm{d}\{\mathbf{z}^{[m]}\}\Big\}}_{\text{KL}\left(q_\phi(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}) \,\middle\|\, p_\theta(\{\mathbf{z}^{[m]}\}|\bar{\mathbf{z}})\right)} \mathrm{d}\bar{\mathbf{z}} \\
&\overset{(a)}{=} \mathbb{E}_{q_\phi(\{\mathbf{z}^{[m]}\}|\mathcal{X})}\Big[\text{KL}\Big(q_\phi(\bar{\mathbf{z}}|\{\mathbf{z}^{[m]}\}) \,\Big\|\, p_\theta(\bar{\mathbf{z}})\Big)\Big] + \mathbb{E}_{q_\phi(\bar{\mathbf{z}}|\mathcal{X})}\Big[\text{KL}\Big(q_\phi(\{\mathbf{z}^{[m]}\}|\{\mathbb{x}^{[m]}\}) \,\Big\|\, p_\theta(\{\mathbf{z}^{[m]}\}|\bar{\mathbf{z}})\Big)\Big].
\end{aligned}$$

For (a), the first term is straightforward, the second term goes through since

$$\int p(x|y,z)\Big\{\int p(y|z)\log\frac{p(y|z)}{q(y|x)}\mathrm{d}y\Big\}\mathrm{d}x = \iint p(y|z)p(x|y,z)\log\frac{p(y|z)}{q(y|x)}\mathrm{d}x\mathrm{d}y$$
$$= \mathbb{E}_{Y|Z}\mathbb{E}_{X|Z,Y}\log\frac{p(y|z)}{q(y|x)} = \mathbb{E}_{Y|Z}\mathbb{E}_{X|Z}\log\frac{p(y|z)}{q(y|x)} = \mathbb{E}_{X|Z}\Big[\mathbb{E}_{Y|Z}\log\frac{p(y|z)}{q(y|x)}\Big];$$

and the last equality holds as a result of the Fubini-Tonelli theorem.

A.4 Evaluating the predictive strength of Granger causal relationships

Next, we briefly discuss how the trained decoder can be used to measure the predictive strength of the Granger causal connections.

Once the model is trained, using the inference procedure described in Section 3.3, one obtains estimates $\hat{\mathbf{z}}^{[m]}$ for all entity-specific graphs. Further, a trained Graph2Trajectory module, abstracted as $\hat{g}_{z\to x}$, also becomes available. The predictive strength of any connection entry (i,j) — corresponding to the lead-lag relationship from j to i — can then be assessed by *nullifying* the corresponding entry. Throughout the remainder of the discussion, we omit superscript $[m]$ for ease of presentation, as the procedure is applicable to an arbitrary entity of interest.

Let $\tilde{\mathbf{z}}^{(ij)}$ be identical to $\hat{\mathbf{z}}$ except that the (i,j) entry is set to zero (nullified). The reconstructed trajectories, based on the estimated and the nullified graphs are given by $\hat{\mathbb{x}} = \hat{g}_{z\to x}(\hat{\mathbf{z}}, \mathbf{x}_1)$[9] and $\tilde{\mathbb{x}}^{(ij)} = \hat{g}_{z\to x}(\tilde{\mathbf{z}}^{(ij)}, \mathbf{x}_1)$, respectively. The predictive strength can then be evaluated based on the difference in the residual-sum-of-squares (RSS), with the latter obtained by evaluating the reconstructed trajectory against the observed values. Concretely, $\mathrm{RSS}(\hat{\mathbb{x}})$ can be obtained by $\frac{1}{T-1}\sum_{t=2}^{T}\|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2$ and that for $\tilde{\mathbb{x}}^{(ij)}$ can be analogously obtained; the predictive strength of the (i,j) connection can then be calculated as $\mathrm{RSS}(\hat{\mathbb{x}}) - \mathrm{RSS}(\tilde{\mathbb{x}}^{(ij)})$. This procedure can be generalized to a set of connections, where instead of nullifying a single entry, multiple entries are nullified simultaneously and the remainder of the evaluation follows. Note that the proposed procedure resembles that of testing for the presence/absence of Granger causality in linear VAR models, where an F-test is used (Geweke, 1984). The calculated difference $\mathrm{RSS}(\hat{\mathbb{x}}) - \mathrm{RSS}(\tilde{\mathbb{x}}^{(ij)})$ also appears in the numerator of the aforementioned F-statistic.

B Additional Synthetic Data Experiments and Results

B.1 Lorenz96 and Springs5 experiments

To explore the applicability of the proposed framework to selected special cases, there are two other settings considered in our synthetic data experiments: the Lorenz96 and the Springs5 systems. Unlike the settings presented in the numerical experiments in Section 4 wherein the entity-level heterogeneity manifests itself primarily in the form of perturbations to the skeleton of the shared common graph, for these two systems, the entity-specific skeletons are either identical across all M entities and only the magnitude of the entries changes (Lorenz96), or they manifest their heterogeneity through a probablistic mechanism (Springs), as explained in the sequel.

Similar to those presented earlier, for both settings, we run the experiments on 5 data replicates and report the metrics after averaging across the 5 runs, with their respective standard deviation included in the parentheses.

[9] Recall that throughout the main sections, we use $\mathbb{x} := \{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ to denote the trajectory; here $\hat{\mathbb{x}}$ is effectively its "reconstructed" couterpart.

B.1.1 The Lorenz96 system

The Lorenz96 system (Lorenz, 1996) has been previously investigated in Tank et al. (2021); Marcinkevičs & Vogt (2021). The dynamics for a p-variable system evolve according to the following ODE:

$$\frac{\mathrm{d}\aleph_i}{\mathrm{d}t} = (\aleph_{i+1} - \aleph_{i-2})\aleph_{i-1} - \aleph_i + F, \qquad i = 1, \cdots, p, \tag{14}$$

where $\aleph_i := \{\mathbf{x}_{i,t}\}$ denotes the continuous time trajectory of node i with $\aleph_0 := \aleph_p, \aleph_{-1} := \aleph_{p-1}$ and $\aleph_{p+1} := \aleph_1$. Such a system corresponds to a Granger-causal structure shown in Figure 8 that depicts its skeleton. The representation in (14) can be obtained from Kerin & Engler (2022):

$$\frac{\mathrm{d}\aleph_i}{\mathrm{d}t} = \alpha(\aleph_{i+1} - \aleph_{i-2})\aleph_{i-1} - \beta\aleph_i + \gamma, \tag{15}$$

by reparameterizing $\alpha = \beta, \lambda = \alpha/\beta$ and setting $F = \alpha\gamma/\beta^2$. F is the forcing constant that controls the degree of non-linearity; in particular, given the relationship between (14) and (15), as F varies, the *strength* of the Granger-causality changes despite an invariant skeleton. In other words, to induce heterogeneity across entities, we can only change the parameter F that induces heterogeneity in the magnitudes of the Granger causal connections, while the skeleton of the Granger causal graph remains the same. We consider a setting with $p = 20$ and $M = 5$ entities, with the forces taking the following values: $F \in \{10.0, 17.5, 25.0, 32.5, 40.0\}$.

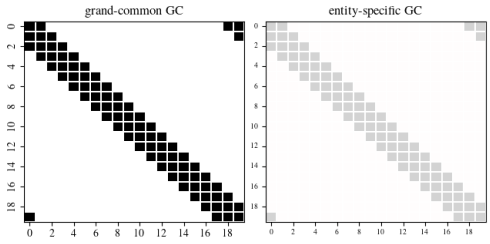


Figure 8: Lorenz96: $\bar{\mathbf{z}}$ and $\mathbf{z}^{[0]}$, showing only the skeleton.

Table 2: Performance evaluation for the estimated $\bar{\mathbf{z}}$ and $\mathbf{z}^{[m]}$'s for setting Lorenz96. Numbers are in % and rounded to integers, and correspond to the mean results based on 5 data replicates

		Generative model-based				Supervised model-based			
		Multi-node	Multi-edge	One-node	One-edge	NGC-cMLP	GVAR	TCDF	Linear
common	AUROC	100(0.1)	100(0.7)	100(0.1)	90(19.7)	97(0.0)	100(0.1)	82(0.9)	99(0.1)
	AUPRC	100(0.4)	99(1.6)	100(0.3)	82(32.5)	87(0.1)	100(0.2)	65(0.9)	97(0.5)
	F1(best)	97(1.5)	96(3.4)	97(1.3)	80(25.7)	87(0.8)	98(1.0)	59(1.3)	89(0.2)
entity (avg)	AUROC	95(1.3)	85(3.7)	96(1.0)	88(1.9)	96(0.1)	97(0.8)	79(0.8)	99(0.1)
	AUPRC	89(2.3)	76(4.6)	91(2.0)	78(2.9)	85(0.3)	90(1.5)	62(0.7)	96(0.3)
	F1(best)	82(3.2)	71(3.5)	84(2.6)	72(3.1)	83(0.4)	83(0.2)	58(0.5)	88(0.3)

The results are shown in Table 2 and the main findings are: (1) consistent with the results in Section 4, the node-centric decoder outperforms the edge-centric one; (2) the proposed joint-learning approach `Multi-node` matches the performance of the supervised `GVAR` and outperforms all other competitors for the common graph; (3) for the entity-specific graphs, interestingly, the linear VAR exhibits a slight edge over all competing methods, while the performance of the proposed model is broadly on-par with the remaining competitors.

Finally, the common and the five entity-specific Granger causal graphs for the `Multi-node` method are depicted in Figure 9. It can be seen that the performance deteriorates for systems with larger external force F.

B.1.2 Springs5 System

This setting is investigated in Kipf et al. (2018); Löwe et al. (2022), and in this work we consider a "multi-entity" version of it. In the original setting, particles (i.e., nodes) are connected (pairwise) by springs at random with probability 0.5; in the case where the connection between particles i and j is present, they interact according to Hooke's law $F_{ij} = -k(r_i - r_j)$, where F_{ij} is the force applied to particle i by particle j, k is the spring constant and r_i is the location vector of particle i in 2-dimensional space. With some initial location and velocity, the trajectories can be simulated by solving Newton's equations

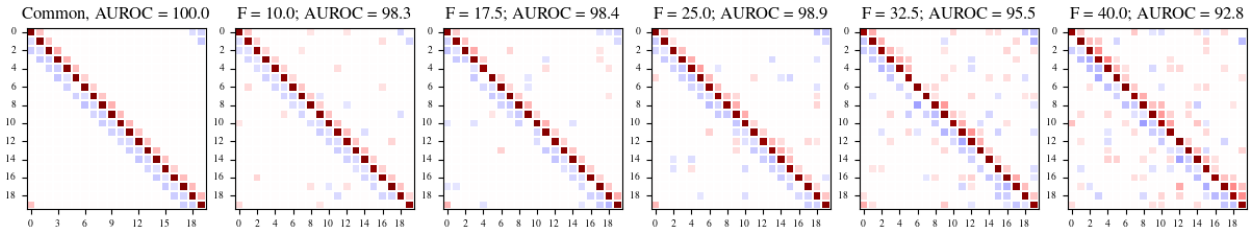


Figure 9: Estimated $\bar{\mathbf{z}}$ and $\mathbf{z}^{[m]}$'s with different F's using the proposed joint-learning framework with a node-centric decoder (`Multi-node`).

of motion (see also Kipf et al. (2018), Appendix B for details). Crucially, (1) the Granger-causal graph is essentially a realization of the homogeneous Erdős-Rényi graph (Erdős & Rényi, 1959) with edge probability being 0.5, and (2) each node's trajectory is multivariate with 4 dimensions, that is, $\mathbf{x}_{i,t} \in \mathbb{R}^d$, $d = 4$; the first 2 dimensions correspond to the velocity and the last 2 to the location in the 2-dimensional space.

The extension to the "multi-entity" case that is suitable for the setup considered in this paper is described next, and it differs primarily from the original one in how the Granger-causal connections across nodes are generated. Specifically, we start from $\bar{\mathbf{z}}$, whose entries (i, j) in its upper-triangular part are generated independently from Beta$(1, 1)$; then set $\bar{\mathbf{z}}_{ji} \equiv \bar{\mathbf{z}}_{ij}, i < j$ so that it's symmetric. For the $\mathbf{z}^{[m]}$'s, let $\mathbf{z}_{ij}^{[m]} \sim \text{Ber}(\bar{\mathbf{z}}_{ij}), i < j$, and then set $\mathbf{z}_{ji}^{[m]} \equiv \mathbf{z}_{ij}^{[m]}$, $\forall\ m = 1, \cdots, M$. Once $\mathbf{z}^{[m]}$'s are generated, they dictate the connections between nodes in their respective systems, and one can proceed with the same procedure as in the original setting to simulate the trajectories. Note that (1) each entity's Granger-causal graph corresponds to a realization of a *heterogeneous* Erdős-Rényi graph; the edge probability differs across node pairs and depends on the corresponding entry in $\bar{\mathbf{z}}$ that is a realization from the Beta distribution, and (2) the grand common structure possesses a "probabilistic" interpretation, in that it effectively captures the *expectation* of an edge being present/absent across all entities. In this experiment, we set $p = 5$ and $M = 10$.

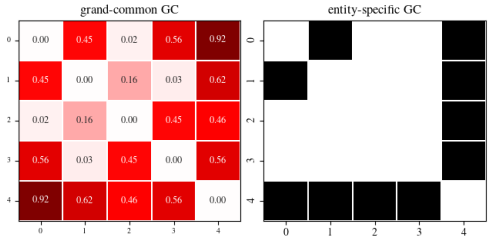


Figure 10: Springs5: $\bar{\mathbf{z}}$ and $\mathbf{z}^{[0]}$. $\mathbf{z}^{[0]}$ is binary (and symmetric) with entries generated according to Bernoulli distributions.

None of the competitors based on the supervised learning models can readily handle this setting[10], and therefore we only present results for those based on generative models. Note that in this experiment, despite that the underlying true graphs are symmetric, we do *not* use this information during our estimation.

Table 3 shows the results for the above-mentioned systems, using both the node- and the edge-centric decoders. A visualization of the estimates is provided in Figure 11. Overall, the proposed joint learning framework outperforms individual learning for entity-level graphs, while the performance is largely comparable for the common graph estimate. Given the physics system nature of this dataset (vis-a-vis time series signals), the edge-centric decoder has a small advantage over the node-centric one; this is manifested by the fact that under the joint learning framework, the two decoders show comparable performance, whereas the edge-centric decoder is clearly superior in the case of single-entity separate learning. Note that this points to another potential advantage of the joint-learning framework, in that it is more robust and exhbits less volatility than individual learning.

[10]There are two issues that the supervised competitors can not readily handle and would require major changes: (1) all of them assume that the Granger-causality to be estimated is numeric and therefore does not naturally handle the binary case, and (2) at any point in time, each node is assumed to have a scalar value, akin to classical time-series settings, whereas here each node is vector valued; consequently, the existing code does not readily handle it.

Table 3: Performance evaluation for the estimated $\bar{\mathbf{z}}$ (error in Frobenious norm) and $\mathbf{z}^{[m]}$'s (accuracy and F1 score after thresholding at 0.5, averaged across all entities) for the Springs5 system.

quantity	metric	Multi-node	Multi-edge	One-node	One-edge
common	ERR-fnorm	1.00(0.259)	0.92(0.294)	1.30(0.412)	0.79(0.217)
entity(avg)	ACC%	99.3(0.84)	99.3(0.76)	87.5(6.45)	96.30(3.99)
entity(avg)	F1Score%	99.5(0.79)	99.4(0.73)	88.2(7.45)	96.27(4.78)

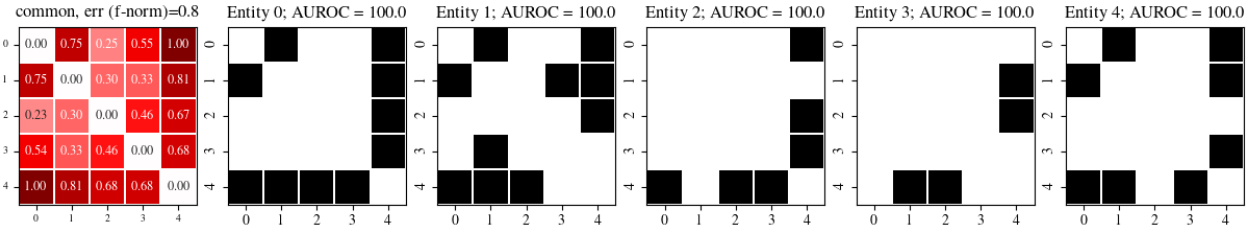


Figure 11: Estimated $\bar{\mathbf{z}}$ and $\mathbf{z}^{[m]}$'s (showing the first five) using the proposed framework with node-centric decoder (`Multi-node`).

B.2 Additional performance evaluation results and their visualization

Table 4 presents additional evaluation metrics (TPR, TNR and ACC) for the proposed method and its strong competitors, after the estimates of the Granger causal graphs are thresholded at various levels no greater than 0.5 (after normalization). We only show the results for the estimated common graph $\bar{\mathbf{z}}$, since the results for the entity-level ones exhibit similar patterns.

As briefly mentioned in Section 4, supervised model-based methods (`NGC`/`GVAR`) are more sensitive to the value of the threshold, manifested by a sudden jump in accuracy once the threshold exceeds a certain level. On the other hand, the change in ACC for the ones based on generative models is more gradual. Given that in practice it is common to use a moderate threshold to eliminate small entries of the initial estimates of the Granger causal graphs to determine their skeleton, the above-mentioned susceptibility can adversely impact the quality of the final estimate used for interpretation purposes and in downstream analytical tasks.

	Multi-node			One-node			NGC-cMLP			GVAR			Linear		
	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC	TPR	TNR	ACC
Linear VAR															
0.10	100	92.1	92.9	98.1	50.3	55.1	100	0.0	10.0	100	0.0	10.0	100	99.9	99.9
0.20	100	99.9	99.9	95.8	78.9	80.6	100	0.0	10.0	100	0.0	10.0	100	100	100
0.30	100	100	100	91.2	90.9	91.0	100	0.0	10.0	100	2.9	12.7	100	100	100
0.40	99.6	100	100	81.8	96.0	94.6	100	48.9	54.2	100	57.4	61.9	100	100	100
0.50	92.7	100	99.3	67.6	98.5	95.4	79.4	99.9	97.9	96.9	100	99.7	98.7	100	99.9
Non-linear VAR															
0.10	100	74.2	76.7	100	59.3	63.1	100	0.0	9.5	100	0.0	9.5	99.5	57.1	61.1
0.20	98.4	89.2	90.0	100	82.9	84.5	100	0.0	9.5	100	0.0	9.5	97.4	99.8	99.5
0.30	94.7	91.7	92.0	96.3	89.3	90.0	100	0.0	9.5	100	85.4	86.8	92.1	100	99.2
0.40	89.5	99.4	98.5	72.1	91.8	89.9	99.5	47.9	52.8	71.1	100	97.2	68.9	100	97.0
0.50	73.2	100	97.5	60.5	95.6	92.2	47.4	95.7	91.2	61.1	100	96.3	60.5	100	96.2
Lotka-Volterra															
0.05	100	72.8	76.8	99.0	40.5	49.3	100	58.4	64.7	34.0	100	90.1	33.3	100	90.0
0.10	100	97.4	97.8	96.3	73.9	77.3	99.7	100	100	33.3	100	90.0	33.3	100	90.0
0.15	99.3	99.8	99.8	90.0	92.4	92.0	90.7	100	98.6	33.3	100	90.0	33.3	100	90.0
0.30	67.0	100	95.0	50.3	100	92.5	33.7	100	90.0	33.3	100	90.0	33.3	100	90.0
0.50	33.3	100	90.0	33.3	100	90.0	33.3	100	90.0	33.3	100	90.0	33.3	100	90.0
Lorenz96															
0.05	95.2	99.5	98.7	93.8	100	98.8	100	0.0	20.0	100	99.8	99.8	95.8	94.1	94.5
0.10	58.8	100	91.8	39.5	100	87.9	100	0.0	20.0	96.8	100	97.0	50.0	100	90.0
0.15	27.2	100	85.5	25.0	100	85.0	100	0.0	20.0	72.8	100	94.5	25.0	100	85.0
0.30	25.0	100	85.0	25.0	100	85.0	100	79.2	83.4	25.0	100	85.0	25.0	100	85.0
0.50	25.0	100	85.0	25.0	100	85.0	93.0	93.4	93.3	25.0	100	85.0	25.0	100	85.0

Table 4: Performance evaluation for the support set of the estimated common graph $\bar{\mathbf{z}}$ at various threshold levels (left-most column). Numbers are in %, and correspond to the mean results based on 5 data replicates.

An illustration of the recovered Granger-causal connections (after "optimal" thresholding) is shown in Figure 12. Note that `NGC` can only produce the "unsigned" version of the connections and hence all its estimates are shown as positive, whereas for other methods, the entries are "signed" with red denoting the positive and blue the negative ones.

One interesting observation is that for the Lotka-Volterra system, all methods have incorrectly estimated the signs of the diagonals, in that the underlying true dependencies on their own lags are positive for the preys and negative for the predators, whereas all methods fail to identify such discrepancy — although for the supervised model-based ones all dependencies show as positive and generative model-based ones have the opposite sign. This could be partially driven by the fact that during trajectory generation, the Runge–Kutta method (specifically, RK4) has been used and thus it renders the presence of a self-lag linear term with coefficient 1 in the recursion; in addition, a small noise term has also been injected.

For this setting, given that the estimated diagonals have dominating magnitude for `GVAR` and `Linear`, we also provide a visual display of the estimates with the diagonals suppressed.

Remark 3. A dichotomous behavior is observed between the unsigned and the signed estimates obtained from the code implementation of `GVAR`[11], with the former typically being 5-10% better (in absolute values, for reported metrics such as AUC, ACC that are between 0-100%). In all the tables, we have reported the performance of the superior one (unsigned), whereas Figure 12 is produced based on the signed estimate to show the positive/negative recovery. The best attainable F1 scores after thresholding (corresponding to the result of the specific data replicate being displayed) for these signed estimates are labeled in the title of the figures; e.g., 0.75 for the non-linear VAR setting, 0.95 and 0.86 for the Lotka-Volterra and the Lorenz96 setting, respectively.

B.3 Lotka-Volterra with perturbation: some characterization

We provide a characterization/justification for the "perturned" Lotka-Volterra system, pertaining to how to validate a Lotka-Volterra system based on the "perturbed" interaction matrix being stable.

The general form of p-multi-species Lotka-Velterra equations are given by

$$\frac{\mathrm{d}\mathbb{x}_i}{\mathrm{d}t} = r_i \mathbb{x}_i \big(1 + \sum_{j=1}^{p} A_{ij} \mathbb{x}_j\big), \tag{16}$$

where $r_i > 0$ is the *inherent per-capita growth rate* of species $\mathbb{x}_i, i = 1, \cdots, p$ and $A \in \mathbb{R}^{p \times p}$ the species interaction matrix. The system considered in (13) can then be put in this canonical form, by assuming that the first $p/2$ species are preys and the last $p/2$ species predators.

Specifically, for the preys the corresponding equation in the canonical form becomes

$$\frac{\mathrm{d}\mathbb{x}^i}{\mathrm{d}t} = \alpha \mathbb{x}_i \Bigg[\big(1 - \frac{1}{\eta^2} \mathbb{x}_i\big) - \beta/\alpha \sum_{j \in \mathcal{P}_i^{\text{prey}}} \mathbb{x}_j \Bigg], \qquad i = 1, \cdots, p/2$$

where $r_i = \alpha$, $A_{ii} = -\frac{1}{\eta^2}$, $A_{ij} = -\beta/\alpha$ for all $j \in \mathcal{P}_i^{\text{prey}}$ otherwise 0; $\mathcal{P}_i^{\text{prey}}$ denotes the support set of the prey indexed by i. Analogously, for the predators the corresponding equation in the canonical form becomes

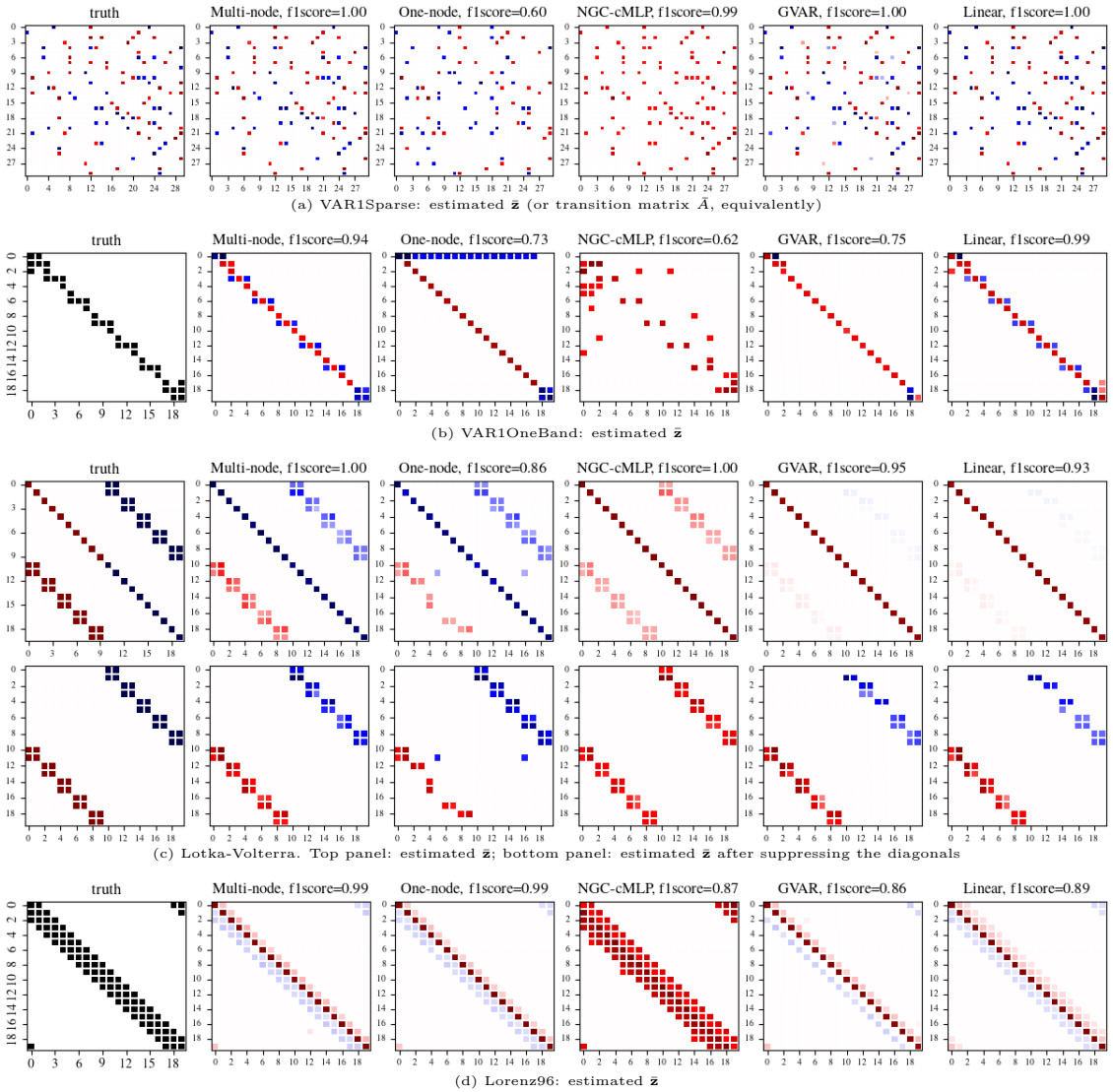
$$\frac{\mathrm{d}\mathbb{x}_i}{\mathrm{d}t} = -\gamma \mathbb{x}_i \big(1 - \delta/\gamma \sum_{j \in \mathcal{P}_i^{\text{predator}}} \mathbb{x}_j\big), \qquad i = p/2 + 1, \cdots, p$$

where $r_i = -\gamma$, $A_{ii} = 0$, $A_{ij} = -\delta/\gamma$ for all $j \in \mathcal{P}_i^{\text{predator}}$ otherwise 0; $\mathcal{P}_i^{\text{predator}}$ denotes the support set of the predator indexed by i.

It can be seen that fixed points of the set of equations in (16) can be found by setting $\mathrm{d}\mathbb{x}_i/\mathrm{d}t = 0$ for all i, which translates to the vector equation

$$\mathbf{r} + A\mathbb{x} = 0, \qquad \mathbf{r} \in \mathbb{R}^p, \mathbb{x} \in \mathbb{R}^p, A \in \mathbb{R}^{p \times p}.$$

[11] Repository for GVAR: `https://github.com/i6092467/GVAR`



(a) VAR1Sparse: estimated $\bar{\mathbf{z}}$ (or transition matrix $\bar{A}$, equivalently)

(b) VAR1OneBand: estimated $\bar{\mathbf{z}}$

(c) Lotka-Volterra. Top panel: estimated $\bar{\mathbf{z}}$; bottom panel: estimated $\bar{\mathbf{z}}$ after suppressing the diagonals

(d) Lorenz96: estimated $\bar{\mathbf{z}}$

Figure 12: Estimated $\bar{\mathbf{z}}$ (after normalization) for various methods. The displayed f1score corresponds to the best attainable one (after thresholding) for each method. Red:(+); blue:(−).

Consequently, fixed points exist if A is invertible and are given by $\mathbb{x} = -A^{-1}\mathbf{r}$. Note that $\mathbb{x}_i = 0$ is a trivial fixed point. Further, the fixed point may contain both positive and negative values, which implies that there is no stable attractor for which the populations of all species are positive. The eigenvalues of A determine the *stability* of the fixed point. By the stable manifold theorem, if its eigenvalues are less than 1, then the fixed point is stable. This can be easily verified once the "perturbed" Granger-causal matrix $\mathbf{z}$'s (which determines the $\mathcal{P}_i$'s and hence the corresponding A) are generated.

C Generalization to multiple levels of grouping

We discuss the generalization of the proposed framework to the case where multiple levels of grouping are present and the corresponding group-common graphs at different levels of the hierarchy are of interest.

Consider L-levels of *nested* grouping where the group assignments become increasingly granular as the level index increases. Specifically, there is a single level-0 group that encompasses all entities, and M (degenerate) level-L groups, with each group m having a singleton member being the entity m; all other levels are cases in between – see also Figure 13 for a pictorial illustration. Note that the case discussed in the main manuscript

corresponds to the special case with $L = 1$. As an example for the case of $L = 2$ levels, consider the data analyzed in Section 5. Suppose that the subjects can be partitioned into 3 groups according to their ages — e.g., less than 30 years old, 30-60 years old, over 60. In such a setting, the single level-0 group comprises of all subjects; the level-1 groups correspond to subjects falling into different age strata; the level-2 groups are the subjects themselves. The quantities of interest are the connectivity patterns shared by subjects within their respective groups at all levels.

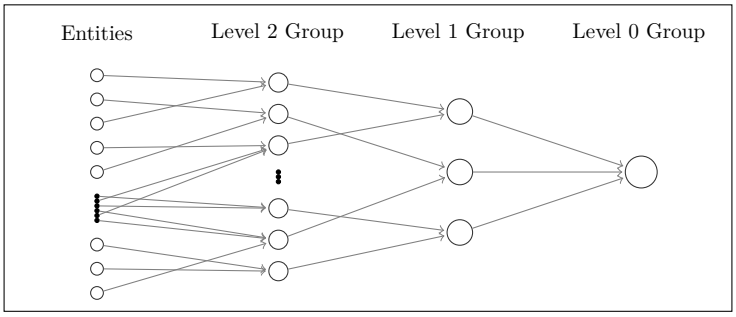


Figure 13: Diagram for a 3-level grouping. Neurons corresponds to G_k^l's that collects the indices of the entities belonging to that group. Solid lines with arrows indicate how small groups from an upper level form larger groups at a lower level.

Let $\mathcal{G}^l := \{G_1^l, \cdots, G_{|\mathcal{G}^l|}^l\}$ denote the collection of groups of level l; each G_k^l is the index set for the entities belonging to group k at level l and the group membership is non-overlapping, that is, $G_{k_1}^l \cap G_{k_2}^l = \emptyset, \forall\, k_1, k_2 \in \{1, \cdots, |\mathcal{G}^l|\}$. The quantities of interest are the entity-specific graphs $\mathbf{z}^{[m]}$, as well as the group-level common structure for all groups at all levels, that is $\bar{\mathbf{z}}^{G_k^l}$, denoting the group-common structure amongst all entities that belong to the kth group, with level-l grouping; $l = 0, \cdots, L-1$ indexes the group level; $k = 1, \cdots, |\mathcal{G}_l|$ indexes the group id within each level. Finally, we let $\bar{\mathbf{z}} \equiv \bar{\mathbf{z}}^{G^0}$, which is consistent with its definition in the main text and it corresponds to the grand-common structure across all entities.

Without getting into the details of each step, the end-to-end learning procedure can be summarized in Figure 14. Compared with the two-level case, the generalization amounts to additional intermediate encoded/decoded distributions in the form of $q_\phi(\mathbf{z}^{[G_k^{l-1}]}|\mathbf{z}^{[G_k^l]})$, $p_\theta(\mathbf{z}^{[G_k^l]}|\mathbf{z}^{[G_k^{l-1}]})$ and $p_\theta(\mathbf{z}^{[G_k^l]}|\cdot)$ (post conjugacy adjustment/merging information); $l = 2, \cdots, L; k = 1, \cdots, |\mathcal{G}_l|$.

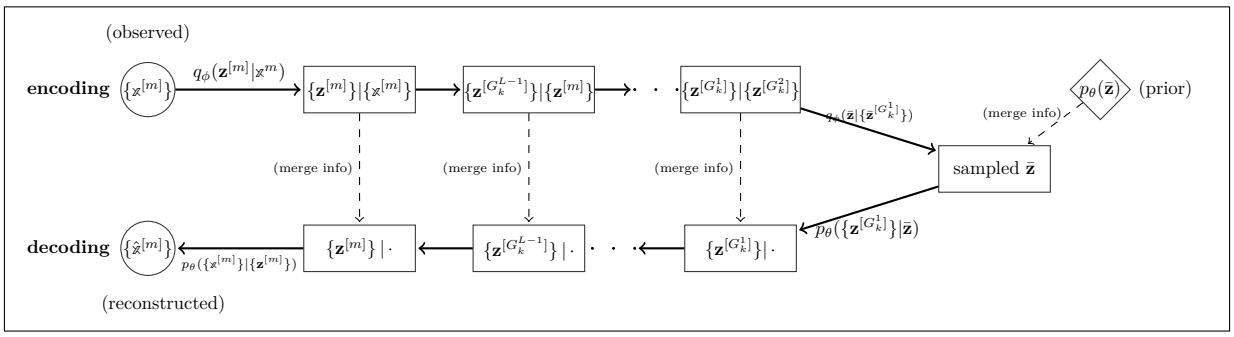


Figure 14: Diagram for the end-to-end encoding-decoding procedure in the presence of multiple levels of grouping.