KarmaTS: A Universal Simulation Platform for Multivariate Time Series with Functional Causal Dynamics

Haixin Li[*] PETER.LI@TUM.DE
School of Computation, Information and Technology (CIT)
Technical University of Munich, Germany
SCAI Lab, D-HEST, ETH Zurich, Switzerland

Yanke Li[*] YANKE.LI@HEST.ETHZ.CH
SCAI Lab, D-HEST, ETH Zurich, Switzerland
Digital Health Care and Rehabilitation, Swiss Paraplegic Research (SPF), Switzerland

Diego Paez-Granados DIEGO.PAEZ@HEST.ETHZ.CH
SCAI Lab, D-HEST, ETH Zurich, Switzerland
Digital Health Care and Rehabilitation, Swiss Paraplegic Research (SPF), Switzerland

Abstract

We introduce KarmaTS, an interactive framework for constructing lag-indexed, executable spatiotemporal causal graphical models for multivariate time series (MTS) simulation. Motivated by the challenge of access-restricted physiological data, KarmaTS generates synthetic MTS with known causal dynamics and augments real-world datasets with expert knowledge. The system constructs a discrete-time structural causal process (DSCP) by combining expert knowledge and algorithmic proposals in a mixed-initiative, human-in-the-loop workflow. The resulting DSCP supports simulation and causal interventions, including those under user-specified distribution shifts. KarmaTS handles mixed variable types, contemporaneous and lagged edges, and modular edge functionals ranging from parameterizable templates to neural network models. Together, these features enable flexible validation and benchmarking of causal discovery algorithms through expert-informed simulation.

Keywords: Clinical Data Simulation, Spatiotemporal Causal Models, Privacy-Preserving Synthetic Data, Multivariate Time Series

Data and Code Availability The code for KarmaTS and the interface mentioned in this paper are available at the GitHub repositories KarmaTS and KarmaTS-HCI, respectively.

Institutional Review Board (IRB) This research does not involve human subjects and therefore did not require Institutional Review Board (IRB) approval.

1. Introduction

1.1. Motivation

Multivariate time series (MTS) arise naturally in domains as diverse as healthcare (Che et al., 2018), earth science (Angryk et al., 2020), transportation (Ghosh et al., 2009), and finance (Tsay, 2013). A hallmark of MTS is the richly interwoven network of temporal and contemporaneous interactions among variables, which causal directed graphs (DAGs), where an edge between two variables signifies a direct causal effect can be represented in an interpretable way. For instance, causal models built on electronic health records have improved early sepsis prediction (Valik et al., 2023) and characterized recovery trajectories after spinal cord injury (Ehrmann et al., 2020), while in the geosciences, they have quantified nonlinear dependencies in large observational series (Runge et al., 2019a). In engineered systems (e.g., monitoring pin voltages on a circuit board), physical connectivity directly defines the causal graph.

By contrast, most real-world applications must infer structure from data, often supplementing partial expert knowledge with *causal discovery* algorithms. These range from constraint-based methods (Spirtes et al., 2000) (PC, FCI) to score-based

[*] These authors contributed equally

searches (GES (Chickering, 2002)) and continuous-optimization approaches (NOTEARS (Zheng et al., 2018)). However, the observed statistical distribution is often a noisy or confounded projection of the underlying physiological mechanism—shaped by latent variables, measurement error, and heterogeneous subpopulations—so purely data-driven discovery may miss dependencies that are weak in the data but structurally inevitable from anatomical or mechanistic constraints that domain experts consider unequivocal (Pearl, 2009; Runge et al., 2019a).

1.2. Contribution

We introduce KarmaTS, a universal simulation framework that unifies expert knowledge elicitation and algorithmic causal discovery for MTS. Our key contributions are:

- **Interactive, lag-indexed graph editor** A user-friendly interface for defining, visualizing, and refining DSCPs (see Definition 1).
- **Executable DSCP runtime** Simulation of MTS from DSCPs, enabling systematic benchmarking and robustness testing of discovery algorithms.
- **Mixed-initiative human–machine loop** An iterative workflow in which experts and causal discovery algorithms inform one another, leading to progressively more accurate graphical models and higher-fidelity synthetic MTS data.

1.3. Paper Roadmap

The remainder of this paper is organized as follows. Section 2 reviews related work on MTS simulation and causal discovery. Sections 3 and C formally introduce KarmaTS and the accompanying user interface. Section 5 demonstrates a human–machine loop application of KarmaTS on real-world fMRI data. Sections 6 and 7 report and analyze empirical evaluations on benchmarks with systematically varied topology and temporal dependence. Finally, Section 8 discusses limitations and future directions.

2. Related Work

Researchers often resort to simulating MTS data to evaluate their algorithms, since high-quality, publicly available datasets remain scarce. While such simulations—built on user-specified dynamics—grant fine-grained control over experimental conditions, they also impose a heavy burden: each new study requires bespoke construction of graphical models and functional mappings. We argue that this redundancy can be greatly reduced by assigning roles more strategically: domain experts should define and validate the causal relationships underlying the data, while algorithm developers concentrate on advancing and benchmarking their methods. The core aim of this work is to bridge these complementary strengths, uniting expert-driven causal specification with streamlined, reusable simulation tools to accelerate and enhance causal discovery research.

2.1. Synthetic MTS Generation

The **CausalTime** pipeline (Cheng et al., 2024b) learns a causal graph from real-world time series using a nonlinear autoregressive model to generate synthetic data. While it uses discriminative scores for fidelity, these aggregate metrics can overlook local patterns and nuances critical in expert-driven fields like healthcare.

2.2. Interactive Causal Modeling Interfaces

Several tools support visual or interactive work with causal graphs. **DAGitty** (Textor et al., 2016) provides a browser-based editor for static DAGs with adjustment-set analysis and bias diagnostics, and **PyRCA** (Liu et al., 2023) offers an interactive environment for root-cause analysis in AIOps, combining causal and dependency graphs with diagnostic workflows.

The classic **Tetrad** system (Scheines et al., 1994) combines a GUI for editing and visualizing graphs with a large collection of causal discovery algorithms and basic simulation tools, including time-indexed variables. However, it is organized around generic DAG/SEM and Bayesian network workflows, and time dependence is typically handled by duplicating variables across time slices rather than via a time-series–native notion of lagged and contemporaneous edges with associated functionals.

2.3. Benchmarking Platforms

CAUSEME (Runge et al., 2019a) is a web-based benchmarking platform with curated ground-truth datasets from diverse domains, featuring challenges like non-stationarity and high dimensionality. Its interface allows experts to upload data, select methods, and compare performance metrics (e.g., precision, recall), standardizing offline evaluation to rigorously assess algorithmic robustness.

2.4. Complementary Roles of KarmaTS and CAUSEME

Whereas CAUSEME focuses on *offline* benchmarking against static and time-varying ground truths, KarmaTS offers an *interactive* simulation environment for *real-time* definition, modification, and refinement of spatio-temporal causal graphical models. Together, these tools form a comprehensive ecosystem. CAUSEME provides community-driven standardized benchmarks for validating causal discovery methods, while KarmaTS empowers experts and developers to iteratively design and fine-tune synthetic MTS datasets tailored to specific research questions. This synergy accelerates both methodological innovation and reliable evaluation in multivariate time series causal discovery. Table 1 provides a summary of the comparison.

	KarmaTS	CausalTime	CAUSEME
Expert Engagement	✓	✗[1]	✗
Functional Definition	✓	✗	✗
Realistic	✗[2]	✓	✓
Real-world Integration	✓[2]	✗	✗

Table 1: Feature comparison between KarmaTS, CausalTime, and CAUSEME.

Notes about Table 1: 1. While CausalTime can incorporate expert knowledge as a prior graph, this input becomes diluted through the pipeline. 2. Via the human-in-a-loop iteration, KarmaTS allows continuous refinement through real-world data fusion to increase the realism.

2.5. Causal Discovery in Multivariate Time Series

Causal discovery in multivariate time series (MTS) addresses temporal and contemporaneous effects, often amid challenges like high dimensionality and irregular sampling.

Constraint-based methods like **PCMCI** (Runge et al., 2019b) and its variants (**PCMCI+** (Runge, 2020), **LPCMCI** (Gerhardus and Runge, 2020)) use conditional independence (CI) tests to filter variables and manage confounders. Structural approaches such as VarLiNGAM (Hyvärinen et al., 2010) combine vector autoregression with LiNGAM assumptions to find effects in non-Gaussian data.

More recent scalable methods include **DYNOTEARS** (Pamfil et al., 2020), which extends the continuous-optimization framework of **NOTEARS** (Zheng et al., 2018) to time series. Deep learning approaches have also emerged: **NGM** (Bellot et al., 2021) uses Neural ODEs to learn causal graphs from irregularly sampled data, while **TCDF** (Nauta et al., 2019) applies attention-based networks to find time delays and detect confounders. Graph-neural methods like **CUTS+** (Cheng et al., 2024a) employ GNNs to scale discovery for high-dimensional series.

These diverse approaches offer complementary strengths. For a comprehensive survey of the field, including datasets and evaluation metrics, see Gong et al. (Gong et al., 2024).

3. KarmaTS: An Overview

KarmaTS is a Python library for MTS generation with expert knowledge integration. In this section, we will first introduce the mathematical model along with the assumptions. Following that, we will explain the implementation details.

3.1. The Mathematical Model

We used a *discrete-time structural causal process* (Runge, 2020) (DSCP), defined in 1. All notations are described in Appendix A

Definition 1 *A* discrete-time structural causal process *is a multivariate dynamical system determined by two components:*

1. A vector process of N variables

$$\mathbf{X}_t = (X_t^1, \dots, X_t^N)$$

2. A set of mappings

$$X_t^j = f^j\left(\mathrm{Pa}(X_t^j), \eta_t^j\right), \quad j = 1, \cdots, N$$

where $\mathrm{Pa}(X_t^j) = \{X \in \mathbf{X} \mid \mathbf{X} \in \{\mathbf{X}_t, \mathbf{X}_{t-1}, \dots\}\} \setminus X_t^j$, *and* η_t^j *is a random variable from an uncertainty process* η^j.

Remark 2

- *In other parts of this paper, we will refer to the set* $\boldsymbol{f} = \{f^j \mid j \in [N]\}$ *as the functional form of the process.*

- *When not specified otherwise, "parents" indicates the union of the upstream nodes from both the contemporaneous and lagged edges.*

Definition 3 *We call an edge* $(X^i_{t-\tau}, X^j_t)$ *to be*

- contemporaneous *if* $\tau = 0$
- lagged *if* $\tau > 0$

The DSCP defined is node-oriented, given the time and index of a variable, as depicted in Figure 1. Rolling through time, the generation dynamics illustrate a spatial-temporal process whose realization is an MTS. The spatial perspective is due to the mappings across different variables despite their progress in time, and the temporal perspective is due to the existence of lagged edges. Taking clinical monitoring as an example, $\mathbf{X}_t = (X^1_t, \ldots, X^N_t)$ could record the collection of values of human temperature, ECG (electrocardiogram), blood pressure, and so on, at the time step t.

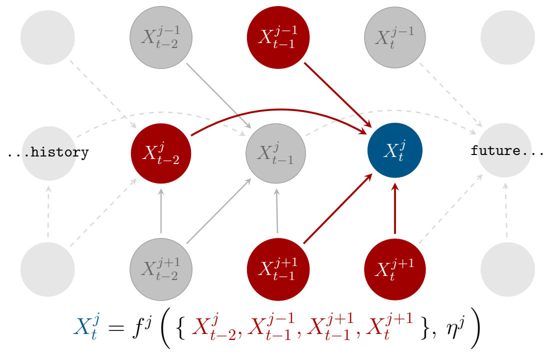


Figure 1: An illustration of Definition 1.

Figure 1 exemplifies the computation of X^j_t (blue). Its parents are colored red. The explicit mapping from a set of parent nodes to their common child node must be defined for the simulation. Generally, a node can have multiple parents; therefore, the effects of those parents can be considered together or separately (see Section 3.2). Ideally, users should specify this N-to-1 mapping function f^j for each node. Although this is sometimes difficult to achieve, we use a default mapping for each edge if none is specified. In addition, users may define mappings corresponding to non-intersecting subsets of a node's parents; in that case, the effects from different parent groups are summed.

A Note on the Functionals The nature of $\boldsymbol{f}$ varies widely across different research domains, taking forms from linear to non-linear, and from simple to intricate. This diversity makes it challenging to emphasize any specific paradigm of $\boldsymbol{f}$ as universally representative, which, thankfully, is mitigated by advances in neural networks (Lu et al., 2021). DSCP, albeit complicated by the functionals, is mathematically powerful enough for modeling real-world processes (Bongers and Mooij, 2018). In Figure 5, 3 and 13, we exemplify with synthetic MTS generated by the DSCP.

3.2. Assumptions

Uncertainty Processes For simplification, we assume the functional f^j is additive with respect to $\text{Pa}(X^j_t)$ and the uncertainty process η^j_t, which means $X^j_t = f^j\left(\text{Pa}(X^j_t), \eta^j_t\right) = f^j\left(\text{Pa}(X^j_t)\right) + f^j\left(\eta^j_t\right)$. Noting that $f^j\left(\eta^j_t\right)$ is a new uncertainty process, we can further define $g^j_t = f^j\left(\eta^j_t\right)$. As a result, we land on the model:

$$X^j_t = f^j\left(\text{Pa}(X^j_t)\right) + g^j_t, \quad j = 1, \cdots, N$$

Additive Decomposition of Edge Functionals The edge function is a mapping from multiple nodes to their common child, i.e., an N-to-1 mapping. It is not always realistic to specify the joint causal effect in a single attempt. Usually, it is more natural to progressively input 1-to-1 mappings, or k-to-1 mappings, where k is a considerably smaller number than N. Therefore, we assume f^j is additive with respect to the local causal effects of subsets of parents. To formally model this, we first partition $\text{Pa}(X^j_t)$:

$$\text{Pa}(X^j_t) = \bigcup_{k=1}^{m} \text{Pa}_k(X^j_t)$$

where $m \leq N$ and $N = |\text{Pa}(X^j_t)|$. In practice, the partition depends on real-world human interaction. Then, we define the local causal effects as:

$$f^j_k : \text{Pa}_k(X^j_t) \mapsto X^j_t$$

Finally, the additive assumption can be written as:

$$f^j(\cdot) = \sum_{k=1}^{m} f^j_k(\cdot)$$

It is useful to assume f^j_k to be non-linear for all k because otherwise we can decompose f^j further with a partition of smaller granularity. Additionally, one may notice that when $m = 1$, it is equivalent to possessing the N-to-1 joint functionals.

3.3. Anatomy of the Framework

In this section, we will introduce how we programmatically construct the process in the definition 1.

a. Graphical Model

Graphical models encode the structure of the DSCP: edges represent the causal relationships among variables. These edges can be (i) derived from empirical expert experience and input via KarmaTS's interface, (ii) learned from data using causal discovery algorithms, or (iii) based on research-backed domain knowledge.

The implementation of the graphical model is built upon NetworkX (Hagberg et al., 2008), which is a major Python package in network analysis.

b. Functional Mappings

Once the graphical model is defined, a naive mapping is assigned to every edge. The naive mapping can either be an identical mapping, which copies the same value to the successor, or a null mapping, which does nothing to the successor.

With the additive functional edges assumption mentioned in 3.2, the human experts will incrementally improve those edges. We provide several options:

The following options are available for constructing functional maps, each offering distinct advantages and applications:

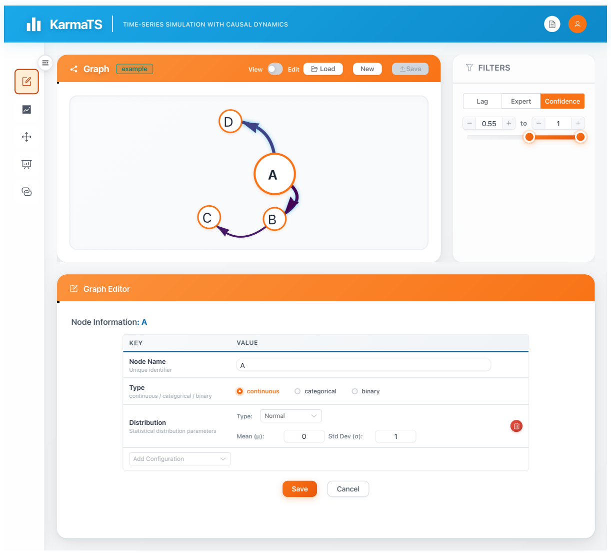
i) Parameterized Templates These are simple, expert-specified rules for ad-hoc mappings between variable types (e.g., thresholding for a continuous-to-binary edge). Their main advantages are rapid implementation and adaptability to changing requirements.

ii) Neural Network Templates This approach uses sophisticated, data-driven models (e.g., ML algorithms, validated equations) to define complex relationships. It offers high reliability and allows for a direct comparison between empirical models and expert knowledge.

4. User Interface for Expert Knowledge Integration

To facilitate the process of creating, editing, and refining causal graphs, we have developed a user interface, shown in Figure 2. This interface is designed to support researchers and practitioners in the iterative process of causal discovery and model refinement.

The user interface is a comprehensive tool that enables users to construct, visualize, and interact with graphical models for time series data. A more comprehensive introduction of the interface is provided in Section C.



0.55 to 1

Node Information: A

KEY	VALUE
Node Name Unique identifier	A
Type continuous / categorical / binary	[✓ continuous] categorical binary
Distribution Statistical distribution parameters	Type: Normal Mean (μ): 0 Std Dev (σ): 1
Add Configuration	

Figure 2: The user interface for expert knowledge input.

In Figure 2, the graph view (upper left) is designed to display and edit the graphical model. The image shows a graph created from scratch by an expert, but in practice, it is possible to load graphs from causal discovery algorithms or saved graphs in another workflow. At the bottom is a dialog for the user to specify the details of the graph elements (nodes and edges).

Figure 3(*a*) shows a graphical model created by an expert interacting with the user interface, representing causal relationships between mixed-type variables, including continuous, binary, and categorical variables. The edges indicate the direction and lag of causal influences among the variables. Figure 3(*b*) shows the simulated time series generated based on the graphical model, showing the dynamics of the mixed-type variables. Variable B is binary (red), and variable E is categorical (multi-colored segments), while the remaining variables exhibit continuous values. The time

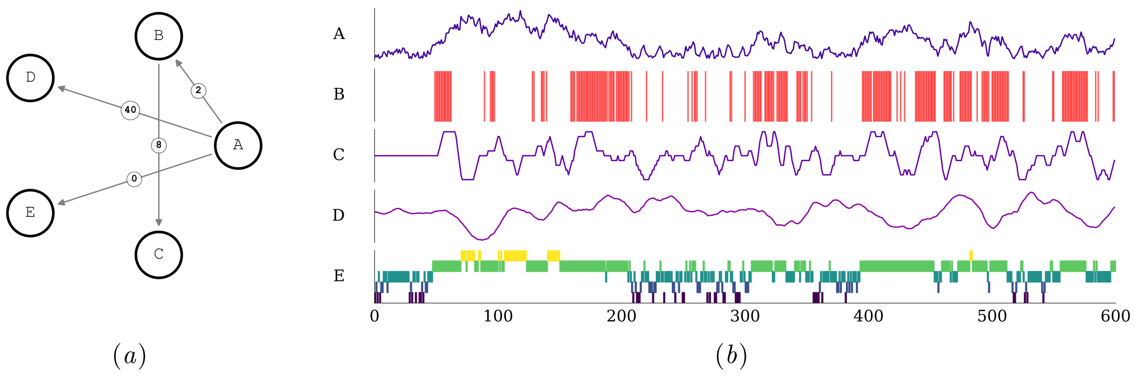


Figure 3: An example of a mixed-type graphical model and the simulated MTS.

series captures the causal dependencies as defined by the expert-generated model. More interface details can be found in the Appendix C.

4.1. The Assumed Role of Human Experts

The interface functions as a labeling system that integrates expert insights, blending theoretical knowledge with empirical observations from complex, real-world conditions. For example, clinical experts can use the system to map the nuanced interactions between physiological parameters subject to unpredictable factors like stress or medication.

KarmaTS mitigates potential individual bias by valuing the collective knowledge of multiple contributors. The underlying graphical model is refined through a collaborative, iterative process that balances human input with algorithmic adjustments, ensuring a more accurate representation of causal structures.

4.2. Human-in-the-loop with KarmaTS

Figure 4 summarizes the mixed-initiative workflow in KarmaTS. Experts provide node and edge information, along with initial functional templates, through the interactive editor. In parallel, real-world time series are analyzed by causal discovery algorithms, whose outputs are treated as structural priors, and by statistical learners that fit edge-wise functionals (potentially with losses tied to downstream objectives). Together, the edited graph and learned functionals define a DSCP (Definition 1), which KarmaTS then uses to simulate synthetic multivariate time series.

Synthetic and real-time series can be fused for downstream tasks such as prediction, robustness analysis, or model selection. At the same time, simulated trajectories and diagnostic plots provide feedback to the experts, who may iteratively refine both structure and functionals.

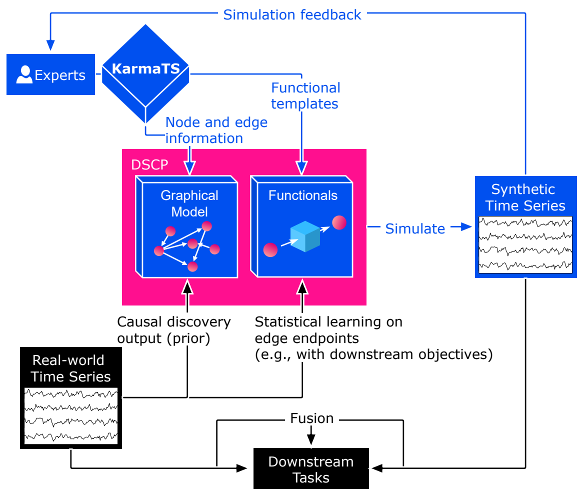


Figure 4: Overview of the human-in-the-loop pipeline with KarmaTS.

5. Example: Privacy-Conscious fMRI Synthesis

Figure 4 illustrates how a DSCP can act as a summary of both expert knowledge and real-world time series. In settings where raw time series are access-restricted (e.g., clinical or industrial logs), sharing DSCP parameters or synthetic outputs can offer a more privacy-conscious alternative to distributing the original data.

In this section, we instantiate the workflow in Figure 4 with a small, illustrative, privacy-conscious use case of fMRI data simulation. However, we do not claim any formal privacy guarantees (e.g., differential privacy Dwork et al. (2006)), and the actual privacy–utility trade-off will depend on the chosen learning objectives and fusion strategy.

5.1. Dataset and Workflow Overview

We consider a movie-watching resting-state functional Magnetic Resonance Imaging (fMRI) dataset, using the MSDL (Multi-Subject Dictionary Learning) atlas as implemented in `Nilearn` (Richardson et al., 2018; Abraham et al., 2014).

To build the DSCP, we follow the expert-edit route in Figure 4: the graphical model is constructed by an "expert" who relies purely on correlation estimates, and the functionals are modeled by statistical learners (detailed in the next subsection). Concretely, the workflow consists of three components, corresponding to Figure 4: (i) *real-world time series* given by MSDL-derived fMRI signals loaded from `Nilearn`, (ii) an *expert knowledge graph* represented by a correlation-thresholded connectivity graph (details in the next subsection), and (iii) *functionals* instantiated as GRU–VAE statistical learners (also detailed below).

5.2. Instantiating the DSCP for fMRI

Below, we specify how the two components of the DSCP from Section 3 are instantiated in this example.

a) Expert Knowledge (Graphical Model) We encode expert knowledge as a correlation-thresholded connectivity graph: nodes are brain regions and an edge is included between regions i and j whenever their functional connectivity $|\mathbf{C}_{ij}|$ exceeds a sparsity threshold τ following Zalesky et al. (2012). For each such pair, we add a lag-1 edge $X^i_{t-1} \to X^j_t$, motivated by the low temporal resolution of fMRI and fast neural dynamics, which makes a fixed single-step lag a common simplification (Smith et al., 2011; Seth et al., 2015). Self-loops with lag 1 are also included, yielding a DAG $\tilde{G} = \big(V, E\big),\ E \subseteq \{(X^i_{t-1}, X^j_t) : i, j \in [N]\}$. For each node X^j_t, representing a brain region at time t, the parent set is $\mathrm{Pa}(X^j_t) = \{X^i_{t-1} : (X^i_{t-1}, X^j_t) \in E\}$, as in Definition 1. This simple, fixed-lag construction does not identify within-lag causal ordering, but provides a concrete anatomy-informed prior for our demonstration.

b) Functionals As introduced in Section 3.3, we model the edge-wise functionals with neural networks, using either fidelity-based losses (e.g., reconstruction error) or task-oriented objectives depending on the downstream goal. For each $j \in [N]$ we learn

$$f^j : \mathrm{Pa}(X^j_t) \longrightarrow X^j_t, \quad X^j_t = f^j\big(\mathrm{Pa}(X^j_t), \eta^j_t\big),$$

where η^j_t denotes stochastic innovations. In this example each f^j is implemented as a *variational autoencoder* (VAE) Kingma and Welling (2013) with a *gated recurrent unit* (GRU) Cho et al. (2014) encoder for sequence modeling: a GRU encodes the parent trajectories into a latent distribution, from which the VAE samples during training and whose mean is used at inference, and a decoder maps latent codes back to the target outputs. Model structure and training details are given in Appendix B.1 and B.2.

5.3. Empirical Behavior and Limitations

In Figure 5(a), the synthetic MTS (blue) is initialized with a short segment of real fMRI data (gray) to match the initial state, but subsequently diverges from the real events, thereby reducing the risk of reproducing subject-specific temporal patterns. Figures 5(b) and 5(c) compare the functional connectivity networks derived from the real and synthetic data, respectively.

Qualitatively, the synthetic network preserves the global organization of the real map: (i) strong left–right mirror–pair edges remain dominant over other cross-hemispheric links, consistent with homotopic interhemispheric functional connectivity (Zuo et al., 2010); (ii) the large-scale community structure resembles canonical resting-state networks reported by Yeo et al. (2011); Power et al. (2011); (iii) key long-range hub pathways align with known hub/rich-club architecture (van den Heuvel and Sporns, 2011); and (iv) although local intra-hemispheric links differ and some weights are redistributed (e.g., a somewhat stronger midline hub, slightly weaker ipsilateral columns), bilateral mirror cores remain salient. This suggests that the synthetic data can maintain key network-level properties of the original dataset while obfuscating fine-grained temporal details. We additionally report a quantitative comparison in Appendix B.3.

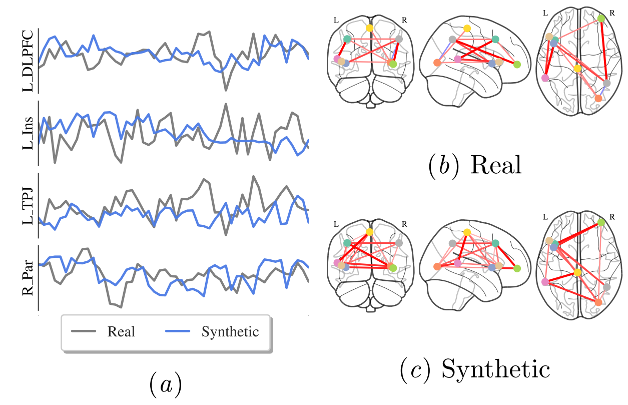


Figure 5: The outcome of privacy-conscious synthetic data generation. We also provide a quantitative comparison in Appendix B.3.

We emphasize that this example is an illustrative, semi-synthetic use case, not a state-of-the-art privacy mechanism: we do not claim formal privacy guarantees such as differential privacy, and the fidelity–privacy trade-off will depend on the choice of functionals, loss, and fusion strategy.

6. Benchmarking Causal Discovery Algorithms on KarmaTS

Standard benchmarks in causal discovery often rely on generic, algorithmically generated graphs that lack domain-specific realism. Involving domain experts is crucial, as their labeled models capture important empirical nuances and can serve as privacy-preserving summaries of sensitive data. This enables the creation of safer, more meaningful benchmarks that reflect real-world dynamics.

Causal Discovery for Time Series Data Causal discovery algorithms have varied assumptions (e.g., constraint- vs. gradient-based), leading to divergent results with no standard evaluation metric. The best measure of performance, therefore, is an algorithm's ability to reconstruct a known ground truth graph. We evaluate how well different methodologies recover template-based, user-specified ground-truth graphs, which serve as proxies for expert-defined structures. We will evaluate the algorithms mentioned in Section 2 using mainstream metrics, i.e., the F1-score and Structural Intervention Distance (SID), on a simulated dataset generated with a specific configuration. Details of the metrics and the simulated dataset are described in Appendix D.

7. Results and Discussion

We report empirical findings in three stages. First, Section 7.1 summarizes *overall accuracy* of six state-of-the-art (SOTA) causal discovery algorithms across the full suite of KarmaTS benchmarks. Section 7.2 disentangles how specific data-generating factors—series length, graph topology, edge density, maximum lag, and latent variable rate—drive performance differences.

Demonstration Protocol Metrics are plotted as line charts against contextually chosen variables (e.g., time horizon and node count). F1-score is presented with an error band denoting the standard deviation across repeated runs with different random seeds. These error bands are symmetric.

7.1. Overall Performance Analysis

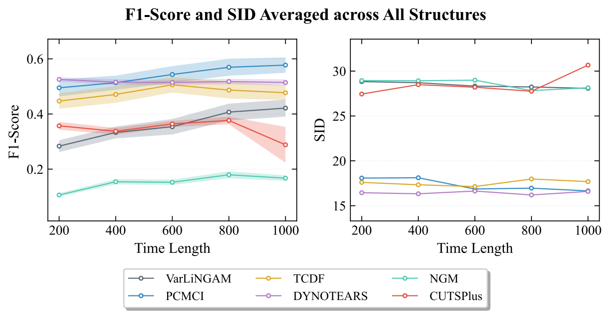


Figure 6: Performance of six causal discovery methods as time length increases (200–1000), averaged over all structures (see D.2).

Figure 6 reveals a clear distinction in model performance across both metrics. Based on the F1-Score, **PCMCI** emerges as the top performer at longer time lengths (F1-Score ∼0.58), while **DYNOTEARS** demonstrates the most stability (∼0.52 across all lengths). This is reinforced by the SID analysis, where **DYNOTEARS** is the clear winner with the lowest and most stable error rate (SID ∼16.5). In contrast, **NGM** is consistently the weakest performer in terms of accuracy (F1-Score < 0.2), and **CUTS+** shows degraded performance with high error rates, especially with longer time series.

External calibration to prior benchmarks This relative ordering is broadly consistent with established time-series causal discovery benchmarks such as CAUSEME and related studies (Runge et al., 2019a; Gong et al., 2024; Runge et al., 2019b; Pamfil et al., 2020; Nauta et al., 2019; Bellot et al., 2021; Cheng et al., 2024a): constraint-based methods like **PCMCI** and continuous-optimization approaches like **DYNOTEARS** typically achieve the strongest structural accuracy, while deep models such as **TCDF** and **NGM** exhibit more variable performance. Our KarmaTS results align with these trends at a coarse level and further clarify how performance shifts with lag, density, and latent-variable rate; Appendix F provides a brief mapping to prior benchmark reports.

7.2. Factor–wise Performance Analysis

Although certain models achieve strong overall performance, some algorithms demonstrate clear advantages in specific datasets. KarmaTS enables researchers to evaluate performance under more granular and adaptable configurations, offering deeper and more targeted insights.

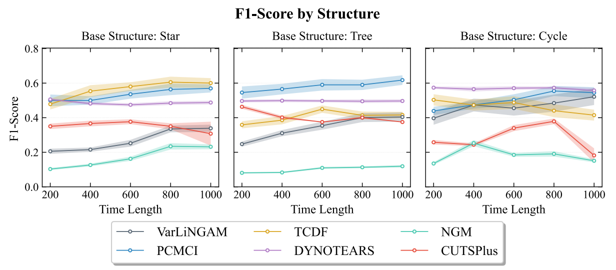


Figure 7: Average F1-scores for three prototypical graph structures (Star, Tree, Cycle). Results have been averaged over all edge configurations (see Section D.2).

As an example, Figure 7 demonstrates that no single model is universally superior. Instead, algorithms exhibit clear, context-specific advantages. For the **Star** structure, **TCDF** shows a distinct advantage, peaking with the highest F1-Score of approximately 0.62. On **Tree** structures, **PCMCI** is the dominant performer, maintaining a stable F1-Score of $\sim$0.6. In the more complex **Cycle** configuration, both **PCMCI** and **DYNOTEARS** prove most robust, achieving stable F1-Scores around 0.55. This targeted evaluation allows researchers to gain finer insights beyond aggregate performance metrics and understand the precise conditions under which a specific algorithm will succeed or fail. More factor-wise analysis is organized in Appendix E.

8. Conclusion

KarmaTS offers a useful tool for integrating human expertise with algorithmic processes for causal discovery in MTS data. By supporting collaboration between domain experts and computational models, the system enhances the representation of causal dynamics. With its flexible interface, customizable uncertainty features, and the ability to simulate datasets with known ground truth, KarmaTS provides a valuable resource for research in various fields, including physiology. Additionally, the system contributes to the iterative refinement of causal models and the evaluation of causal discovery algorithms.

Limitations The system lacks comprehensive validation of its effectiveness across user groups and use cases, including medical researchers, data scientists, and domain experts, and the benchmarks used in this paper currently rely on template-based, user-specified proxy graphs rather than fully elicited expert structures. More extensive testing is needed to evaluate how different users interact with and benefit from the system's features, particularly in real-world analytical scenarios.

Future Work Future research directions include conducting user studies with domain experts to validate the system's effectiveness and usability, as well as refining the expert-elicitation protocol used to construct and revise DSCP graphs. Additionally, we aim to incorporate more sophisticated statistical methods for uncertainty estimation and develop robust quantification approaches for learned causal relationships. These improvements will strengthen the system's support of reliable causal inference in time series analysis.

Acknowledgment

This research project was partially supported by the JST Moonshot R&D Program, Grant Number JPMJMS2034-18.

References

Alexandre Abraham, Fabian Pedregosa, et al. Nilearn: Machine learning for neuroimaging in Python. *Journal of Machine Learning Research*, 17:1–5, 2014.

Rafal A Angryk, Petrus C Martens, Berkay Aydin, Dustin Kempton, Sushant S Mahajan, Sunitha Basodi, Azim Ahmadzadeh, Xumin Cai, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, et al. Multivariate time series dataset for space weather data analytics. *Scientific data*, 7(1):227, 2020.

Alexis Bellot, Kim Branson, and Mihaela van der Schaar. Neural graphical modelling in continuous-time: consistency guarantees and algorithms. In *International Conference on Learning Representations*, 2021.

Stephan Bongers and Joris M. Mooij. From random differential equations to structural causal models: the stochastic case. *CoRR*, abs/1803.08784, 2018. URL http://arxiv.org/abs/1803.08784.

Ruichu Cai, Zhiyi Huang, Wei Chen, Zhifeng Hao, and Kun Zhang. Causal discovery with latent confounders based on higher-order cumulants. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 3380–3407. PMLR, 2023. Shows that, in certain structured settings, latent variables can help identifiability via rank constraints and higher-order cumulants.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085, 2018.

Yuxiao Cheng, Lianglong Li, Tingxiong Xiao, Zongren Li, Jinli Suo, Kunlun He, and Qionghai Dai. Cuts+: High-dimensional causal discovery from irregular time-series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11525–11533, 2024a.

Yuxiao Cheng, Ziqian Wang, Tingxiong Xiao, Qin Zhong, Jinli Suo, and Kunlun He. Causaltime: Realistically generated time-series for benchmarking of causal discovery. In *The Twelfth International Conference on Learning Representations*, 2024b.

David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. URL https://arxiv.org/abs/1406.1078.

Xinshuai Dong, Biwei Huang, Ignavier Ng, Xiangchen Song, Yujia Zheng, Songyao Jin, Roberto Legaspi, Peter Spirtes, and Kun Zhang. A versatile causal discovery framework to allow causally-related hidden variables. In *International Conference on Learning Representations (ICLR) 2024*, 2024. Presents RLCD, a rank-based latent causal discovery algorithm that often benefits from additional latent structure to enhance identifiability.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, pages 265–284. Springer, 2006. doi: 10.1007/11681878_14.

Cristina Ehrmann, Jan D. Reinhardt, Conran Joseph, Nazirah Hasnan, Brigitte Perrouin-Verbe, Piotr Tederko, Mauro Zampolini, InSCI, and Gerold Stucki. Describing Functioning in People Living With Spinal Cord Injury Across 22 Countries: A Graphical Modeling Approach. *Archives of Physical Medicine and Rehabilitation*, 101(12): 2112–2143, December 2020. ISSN 1532-821X. doi: 10.1016/j.apmr.2020.09.374.

Andreas Gerhardus and Jakob Runge. High-recall causal discovery for autocorrelated time series with latent confounders. *Advances in Neural Information Processing Systems*, 33:12615–12625, 2020.

Bidisha Ghosh, Biswajit Basu, and Margaret O'Mahony. Multivariate short-term traffic flow forecasting using time-series analysis. *IEEE transactions on intelligent transportation systems*, 10(2): 246–254, 2009.

Chang Gong, Chuzhe Zhang, Di Yao, Jingping Bi, Wenbin Li, and YongJun Xu. Causal discovery from temporal data: An overview and new perspectives. *ACM Computing Surveys*, 57(4):1–38, 2024.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of* *the 7th Python in Science Conference*, pages 11–15, Pasadena, CA USA, 2008.

Frank Harary. *Graph Theory.* Addison–Wesley, 1969.

Aapo Hyvärinen, Kun Zhang, Shohei Shimizu, and Patrik O Hoyer. Estimation of a structural vector autoregression model using non-gaussianity. *Journal of Machine Learning Research*, 11(5), 2010.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Chunlin Liu, Zhe Zhang, Zheng Wen, et al. PyRCA: A python library for root cause analysis in large-scale systems. *arXiv preprint arXiv:2306.11417*, 2023. Includes an interactive dashboard for causal graph visualization and expert-knowledge injection.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Learning nonlinear operators via deeponet. *Nature Machine Intelligence*, 3(3):218–229, 2021.

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

Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. Dynotears: Structure learning from time-series data. In *International Conference on Artificial Intelligence and Statistics*, pages 1595–1605. PMLR, 2020.

Judea Pearl. *Causality.* Cambridge university press, 2009.

Jonathan D. Power, Alexander L. Cohen, Steven M. Nelson, G. R. Wig, Kelly A. Barnes, Jessica A. Church, Alecia C. Vogel, Timothy O. Laumann, Fran M. Miezin, Bradford L. Schlaggar, and Steven E. Petersen. Functional network organization of the human brain. *Neuron*, 72(4):665–678, 2011. doi: 10.1016/j.neuron.2011.09.006.

Hilary Richardson, Grace Lisandrelli, Alexa Riobueno-Naylor, and Rebecca Saxe. Development of the social brain from age three to twelve years. *Nature Communications*, 9(1):1027, 2018.

Jakob Runge. Discovering contemporaneous and lagged causal relations in autocorrelated nonlinear time series datasets. In *Conference on Uncertainty in Artificial Intelligence*, pages 1388–1397. PMLR, 2020.

Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D. Mahecha, Jordi Muñoz-Marí, Egbert H. Van Nes, Jonas Peters, Rick Quax, Markus Reichstein, Marten Scheffer, Bernhard Schölkopf, Peter Spirtes, George Sugihara, Jie Sun, Kun Zhang, and Jakob Zscheischler. Inferring causation from time series in Earth system sciences. *Nature Communications*, 10(1):2553, June 2019a. ISSN 2041-1723. doi: 10.1038/s41467-019-10105-3.

Jakob Runge, Peer Nowack, Marlene Kretschmer, Seth Flaxman, and Dino Sejdinovic. Detecting and quantifying causal associations in large nonlinear time series datasets. *Science advances*, 5(11):eaau4996, 2019b.

Richard Scheines, Peter Spirtes, Clark Glymour, and Christopher Meek. Tetrad ii: Tools for discovery, 1994.

Anil K. Seth, Patrick Chorley, and Lionel Barnett. Granger causality analysis of fmri bold signals is invariant to hemodynamic convolution but not downsampling. *NeuroImage*, 119:418–426, 2015. doi: 10.1016/j.neuroimage.2015.06.088.

Stephen M. Smith, Mark Jenkinson, Mark W. Woolrich, Christian F. Beckmann, Timothy E. J. Behrens, Heidi Johansen-Berg, Peter R. Bannister, Marilena De Luca, Ivana Drobnjak, David E. Flitney, Rami K. Niazy, James Saunders, John Vickers, Yongyue Zhang, Nicola De Stefano, J. Michael Brady, and Paul M. Matthews. Network modelling methods for fmri. *NeuroImage*, 54(2):875–891, 2011. doi: 10.1016/j.neuroimage.2010.08.063.

Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search.* MIT press, 2000.

Johannes Textor, Benito van der Zander, Mark S. Gilthorpe, Maciej Liśkiewicz, and George T. M. Leyland. Robust causal inference using directed acyclic graphs: the R package `dagitty`. *International Journal of Epidemiology*, 45(6):1887–1894, 2016.

Ruey S Tsay. *Multivariate time series analysis: with R and financial applications.* John Wiley & Sons, 2013.

John Karlsson Valik, Logan Ward, Hideyuki Tanushi, Anders F. Johansson, Anna Färnert, Mads Lause Mogensen, Brian W. Pickering, Vitaly Herasevich, Hercules Dalianis, Aron Henriksson, and Pontus Nauclér. Predicting sepsis onset using a machine learned causal probabilistic network algorithm based on electronic health records data. *Scientific Reports*, 13(1):11760, July 2023. ISSN 2045-2322. doi: 10.1038/s41598-023-38858-4.

Martijn P. van den Heuvel and Olaf Sporns. Rich-club organization of the human connectome. *Journal of Neuroscience*, 31(44):15775–15786, 2011. doi: 10.1523/JNEUROSCI.3539-11.2011.

B. T. Thomas Yeo, Fenna M. Krienen, Jorge Sepulcre, Mert R. Sabuncu, Danial Lashkari, Marisa Hollinshead, Joshua L. Roffman, Jordan W. Smoller, Lilla Zöllei, Jonathan R. Polimeni, Bruce Fischl, Hesheng Liu, and Randy L. Buckner. The organization of the human cerebral cortex estimated by intrinsic functional connectivity. *Journal of Neurophysiology*, 106(3):1125–1165, 2011. doi: 10.1152/jn.00338.2011.

Andrew Zalesky, Alex Fornito, and Ed Bullmore. On the use of correlation as a measure of network connectivity. *NeuroImage*, 60(4):2096–2106, 2012.

Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Xi-Nian Zuo, Clare Kelly, Adriana Di Martino, Maarten Mennes, Daniel S. Margulies, Saroja Bangaru, Rebecca Grzadzinski, Alan C. Evans, Yu-Feng Zang, F. Xavier Castellanos, and Michael P. Milham. Growing together and growing apart: Regional and sex differences in the lifespan developmental trajectories of functional homotopy. *Journal of Neuroscience*, 30(45):15034–15043, 2010. doi: 10.1523/JNEUROSCI.2612-10.2010.

Appendix A. Notations

We denote the set of characteristics by X^j, where $j \in [N]$ and N is the number of characteristics. Each feature X^j also represents the associated random process $X^j : t \mapsto X^j_t$, mapping time steps to random variables. Following standard conventions, we use X^j_t to denote the random variable at time step t, and x^j_t for its observed value (realization).

In our graphical model, the nodes correspond to the random variables X^j_t. Since there is no ambiguity, we use X^j_t to refer to both the random variable and the corresponding node in the graph. The set $\mathrm{Pa}(X^j_t)$ denotes the causal parents of X^j_t within the graph. Extending this notation to realizations, $\mathrm{Pa}(x^j_t)$ represents the collection of observed values corresponding to the parents of X^j_t.

For edge notation, we use (X, Y) to represent a directed edge from X to Y, indicating that X is a direct cause of Y. An undirected edge between X and Y is denoted by $\{X, Y\}$.

Appendix B. Supplementary Information for the fMRI Example

B.1. VAE-based Functional Model

For a given target process X^j with parent set $\mathrm{Pa}(X^j)$, we model the edge functional from parent histories to the next value of X^j using a variational autoencoder (VAE) (Kingma and Welling, 2013) with a GRU encoder.

Fix a history length L and a prediction horizon H. For each time index t, we collect the parent history window

$$\mathrm{Pa}(x^j_{t-L+1:t}) \;\in\; \mathbb{R}^{L \times d_j},$$

where $d_j = |\mathrm{Pa}(X^j)|$ is the number of parents, and the corresponding target sequence for the child

$$y^j_{1:H} \;=\; \big(x^j_{t+1}, \ldots, x^j_{t+H}\big).$$

The encoder is a unidirectional GRU that reads the parent sequence and outputs a hidden state $h^j \in \mathbb{R}^h$ summarizing the history:

$$h^j \;=\; \mathrm{GRU}_\phi\big(\mathrm{Pa}(x^j_{t-L+1:t})\big).$$

From h^j we obtain the mean and (log-)variance of a latent Gaussian,

$$\mu^j \;=\; W_\mu h^j + b_\mu, \qquad \log \sigma^{j2} \;=\; W_\sigma h^j + b_\sigma,$$

and sample a latent code via the reparameterization trick

$$z^j = \mu^j + \sigma^j \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I).$$

In our implementation, the latent code z^j is directly projected to a scalar prediction of the next value of X^j; rolling this one-step predictor forward with teacher forcing on non-target variables yields a predicted sequence

$$\hat{y}^j_{1:H} = (\hat{x}^j_{t+1}, \dots, \widehat{x^j_{t+H}})$$

that is used in the DSCP update for node X^j.

B.2. VAE Mixed Loss

Training uses a composite loss that balances sequence reconstruction, low-order statistics, temporal dependence, cross-variable correlation structure, and a KL regularizer on the latent space. For brevity, we drop the superscript j and write $y_{1:H}$ and $\hat{y}_{1:H}$ for the true and predicted sequences over a batch.

Reconstruction Term We define a time-weighted mean-squared error over the prediction horizon:

$$\mathcal{L}_{\text{rec}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{h=1}^{H} w_h \left(\hat{y}_{b,h} - y_{b,h}\right)^2,$$

where B is the batch size and w_h increases with h to emphasize later predictions.

Marginal Statistics To encourage the synthetic trajectories to match the marginal level and variability of the real series, we match the global mean and standard deviation over batch and time:

$$\mathcal{L}_{\text{stat}} = \left(\hat{\mu} - \mu\right)^2 + \left(\hat{\sigma} - \sigma\right)^2,$$

where $\hat{\mu}, \hat{\sigma}$ are the mean and standard deviation of $\hat{y}_{1:H}$ and μ, σ are those of $y_{1:H}$.

Lag-1 Autocorrelation We also match the lag-1 autocorrelation,

$$r = \mathbb{E}[y_t y_{t+1}], \qquad \hat{r} = \mathbb{E}[\hat{y}_t \hat{y}_{t+1}],$$

via

$$\mathcal{L}_{\text{ac}} = \left(\hat{r} - r\right)^2,$$

computed empirically over batch and time.

Cross-variable Correlation Row Let $X_t^1, \dots, X_t^N$ denote all processes at time t and consider the empirical Pearson correlations between X^j and all other variables over the horizon, arranged in a vector

$$c \in \mathbb{R}^N, \qquad c_k = \operatorname{corr}\left(y_{1:H}, x^k_{1:H}\right).$$

Using the same construction with $\hat{y}_{1:H}$ yields $\hat{c} \in \mathbb{R}^N$. We penalize deviations in this correlation "row" via

$$\mathcal{L}_{\text{corr}} = \|\hat{c} - c\|_2^2.$$

KL Regularization For each latent code, the encoder defines a Gaussian posterior $q_\phi(z \mid \text{Pa}(x^j_{t-L+1:t})) = \mathcal{N}(\mu, \text{diag}(\sigma^2))$. We regularize it toward the standard normal prior $p(z) = \mathcal{N}(0, I)$ using the closed-form KL divergence

$$\mathcal{L}_{\text{KL}} = \mathbb{E}\left[\frac{1}{2} \sum_d \left(\mu_d^2 + \sigma_d^2 - 1 - \log \sigma_d^2\right)\right],$$

averaged over batch and time.

Total Objective The full VAE mixed loss combines these terms with scalar weights

$$\begin{aligned}\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{stat}} \mathcal{L}_{\text{stat}} + \lambda_{\text{ac}} \mathcal{L}_{\text{ac}} \\ + \lambda_{\text{corr}} \mathcal{L}_{\text{corr}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}.\end{aligned}$$

where $\lambda_{\text{stat}}, \lambda_{\text{ac}}, \lambda_{\text{corr}}, \lambda_{\text{KL}}$ control the strength of each regularizer. This objective encourages each learned edge functional to reproduce not only the one-step dynamics of X^j, but also its basic marginal behavior, short-range temporal dependence, and correlation pattern with the rest of the MTS, while maintaining a regularized latent representation.

B.3. Results

To quantify pattern similarity between real and synthetic connectivity, we report a "matrix correlation" (`Matrix_Corr`) score: the Pearson correlation between the vectorized off-diagonal entries of the two Pearson correlation matrices, following common practice in network neuroscience for comparing connectomes (Yeo et al., 2011; Power et al., 2011; van den Heuvel and Sporns, 2011). In Figure B.3, this score is 0.5097, indicating moderate alignment of the edge-strength patterns despite differences in individual weights.

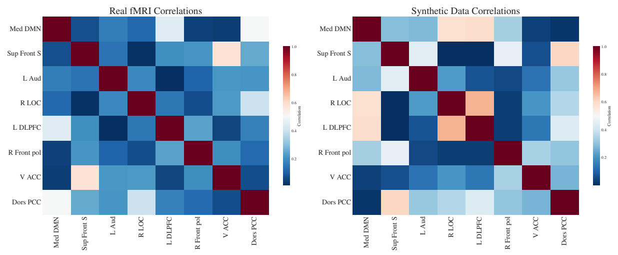


Figure 8: Correlation matrix of the brain connections from real fMRI data (left) and synthetic time series (right).

B.4. Same Expert Graph with Different Functionals

Setup We keep the same expert graph and swap the edge functionals: (i) a VAE-based generator and (ii) a Transformer-based generator. We then compare the synthetic vs. real Pearson correlation matrices in Table 2 using the following complementary metrics to measure the similarity between brain connection graphs from synthetic data and real data.

Metrics	VAE	Transformer
Matrix_Corr (↑)	**0.5097**	-0.0342
MAE (↓)	0.3820	**0.2993**
RMSE (↓)	0.4927	**0.4355**
Frobenius Norm (↓)	3.6871	**3.2592**
Cosine (↑)	**0.5178**	0.1015
Spectral L_2 (↓)	1.6243	**0.6020**

Table 2: Matrix_Corr and Cosine similarity measure the overall pattern and edge directional alignment. MAE, RMSE, and Frobenius Norm showcase the absolute entry-wise distance, while Spectral L_2 tells the global eigen-structure similarity via the L_2 distance between eigenvalue spectra.

Observation There is a clear trade-off: the VAE better preserves the pattern of correlations (relatively strong/weak edges), while the Transformer is closer in absolute error and eigen-structure. For uses that threshold correlations to recover edges/hubs or compare top-k links, the VAE is preferable (higher Matrix_Corr/Cosine). For applications driven by global network statistics (e.g., spectral or diffusion properties), the Transformer is preferable (lower MAE/RMSE/Frobenius Norm and Spectral L_2).

Conclusion This is a small, illustrative experiment, not a state-of-the-art benchmarking effort. The comparison in the table is intended to clarify behavior under a fixed modeling choice, not to rank models.

Within this constrained setup, the table highlights a trade-off: the VAE better preserves pattern alignment, while the Transformer is closer in absolute error and spectral structure. This enriches the example without positioning it as a competitive benchmark.

Appendix C. User-Interface

C.1. User Interface Features

Key functionalities of the interface include:

Graph Creation and Visualization As shown in the upper left panel in Figure 9, users can interactively build, edit, and visualize causal graphs alongside their corresponding time series data, labeling nodes and edges with domain-specific meanings.

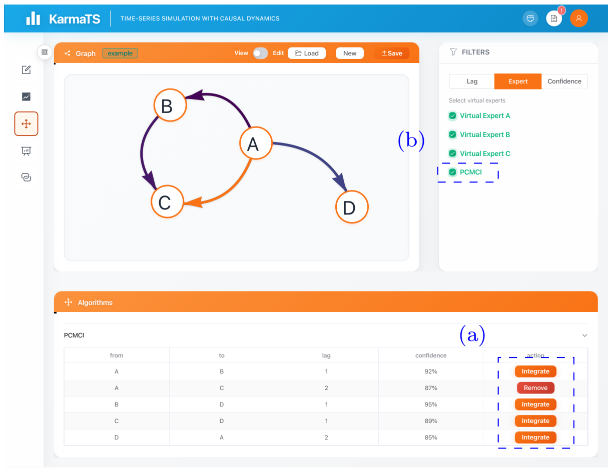
Algorithm Integration The interface integrates with causal discovery algorithms to generate initial graphs, which experts can then iteratively refine based on their domain knowledge (see Figure 9).

Synthetic Data Generation Users can generate configurable synthetic time series data from the specified causal graphs to test hypotheses and validate causal models (see Figure 10).

Collaborative Environment The system supports real-time, multi-user editing of causal graphs, with version control included to track and manage changes.

C.2. Example Editing View

Node Editing Figure 11(*a*) shows the input interface for adding or editing a categorical variable, such as "activities." The user enumerates the possible values of interest by adding items; in the background, the categories are assigned unique integer labels so that the backend can utilize them seamlessly. Similarly, Figure 11(*b*) shows an example of a continuous variable, "temperature." The minimum and maximum fields define the range, and an offset records a typical value of the variable. Additional configurations include a memo field for notes to the algorithm developer, an aggregation mode (averaging, summing,



from	to	lag	confidence	action
A	B	1	92%	Integrate
A	C	2	87%	Remove
B	D	1	95%	Integrate
C	D	1	89%	Integrate
D	A	2	85%	Integrate

Figure 9: Edges suggested by causal discovery algorithms. (a) Experts can decide which ones to integrate. (b) An "expert" named after the algorithm will appear for the integrated edge.

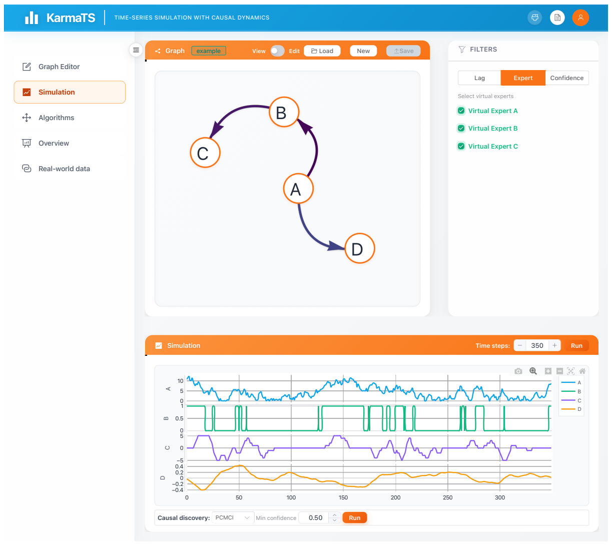
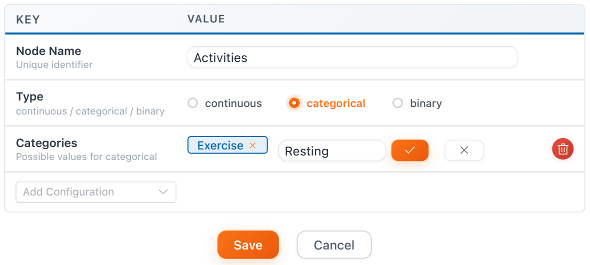


Figure 10: Time series simulation from a defined DSCP over a set of mixed-type variables.

voting, etc., depending on the variable type), and a typical range when the variable is continuous.

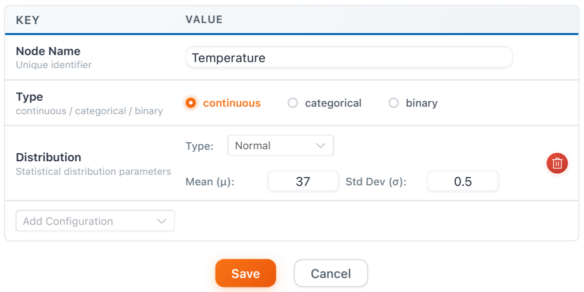
Edge Editing Figure 12 illustrates how an edge can be edited. An important configuration is the choice of a functional, which can be either a statistical learner or a simple template, such as a linear model over a short window of the parent node's history.



KEY	VALUE
Node Name Unique identifier	Activities
Type continuous / categorical / binary	continuous [✓ categorical] binary
Categories Possible values for categorical	Exercise × Resting ✓ ×
Add Configuration	

Save   Cancel

(*a*) Node editor for categorical variables



KEY	VALUE
Node Name Unique identifier	Temperature
Type continuous / categorical / binary	[✓ continuous] categorical binary
Distribution Statistical distribution parameters	Type: Normal Mean (μ): 37 Std Dev (σ): 0.5
Add Configuration	

Save   Cancel

(*b*) Node editor for continuous variables

Figure 11: Dialogues for node editing are illustrated using an example from a physiological scenario.

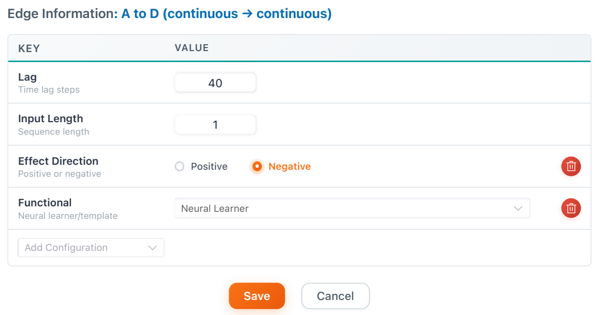
Appendix D. Experiment Details

D.1. Metrics

F1-score We evaluate the performance of causal discovery algorithms by **F1 Score**, which provides a balanced measure of precision and recall, offering a comprehensive assessment of accuracy. The F1 Score is defined as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

where Precision $= \frac{|\text{TP}|}{|\text{TP}|+|\text{FP}|}$ and Recall $= \frac{|\text{TP}|}{|\text{TP}|+|\text{FN}|}$. In our experiment, true positives (**TP**) refer to cases



Edge Information: A to D (continuous → continuous)

KEY	VALUE
Lag Time lag steps	40
Input Length Sequence length	1
Effect Direction Positive or negative	Positive [✓ Negative]
Functional Neural learner/template	Neural Learner

Figure 12: A dialogue for edge editing.

where both the source and target nodes of an edge are correctly identified, and the lag falls within an acceptable window. False positives (**FP**) are incorrectly identified edges, while false negatives (**FN**) represent missing causal edges. For *summary graphs*—in which edges are aggregated irrespective of their lags—an edge is deemed correctly identified if its source and target nodes correspond to those in the ground truth.

Structural Intervention Distance (SID) SID evaluates a learned causal graph, $\hat{G}$, by comparing it to a ground-truth graph, G. Unlike purely structural metrics like the Structural Hamming Distance (SHD), SID measures discrepancies in the causal effects implied by each graph.

Formally, SID is defined as the number of pairs of nodes (i, j) for which the set of causal parents of j under an intervention on i differs between the two graphs. Let $\text{Pa}_G(j|do(i))$ be the set of parents of node j in the manipulated graph that results from an intervention on node i in graph G. The SID is then calculated as:

$$\text{SID}(\hat{G}, G) = \sum_{i \neq j} \mathbb{I}\left(\text{Pa}_{\hat{G}}(j|do(i)) \neq \text{Pa}_G(j|do(i))\right)$$

where $\mathbb{I}(\cdot)$ is the indicator function. In simpler terms, a point is added to the distance for every pair of nodes where an intervention on one node leads to a different causal pathway to the other. This makes SID highly sensitive to errors in the causal hierarchy, providing a more robust measure of causal fidelity than metrics that simply count edge differences.

D.2. Simulated Datasets

We developed multiple simulated datasets featuring graph instances with a consistently fixed number of

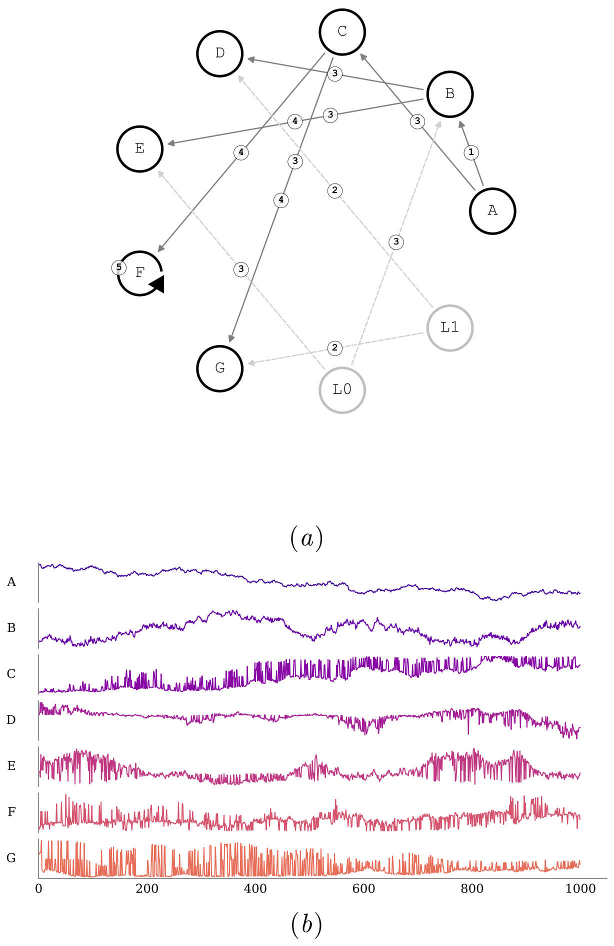


Figure 13: Simulated data with latent variables.

nodes of *five*, allowing for an in-depth exploration of complex structural types while maintaining computational manageability. The ground-truth graphs in these datasets are template-based synthetic motifs (star, tree, cycle) with lagged edges, used as proxies for user- or expert-specified structures rather than graphs directly elicited from human experts; they are schema-compatible with true expert graphs in KarmaTS (same lagged/contemporaneous representation and export format), so real expert graphs can be imported and used interchangeably in the same benchmarking pipeline. The dataset is systematically varied along three major dimensions:

i) *Structural configurations.* We choose three classical graph topologies—*star graphs*, *cycle graphs*, and *(rooted) trees*—as defined in standard graph theory (e.g. Harary (1969)). They represent distinct challenges in causal inference and reflect common patterns observed in real-world networks.

ii) *Maximum lag of edges.* This parameter governs the temporal reach of causal links, allowing us to assess each algorithm's ability to resolve delayed interactions inherent in dynamic systems. Here, we hold the edge–node ratio constant (see iii)). We evaluated two lag settings: (1) *small*, with a maximum lag of 5 time units; and (2) *large*, with a maximum lag of 10 time units.

iii) *Edge–node ratio (ENR).* Defined as $|E|/|V|$, the ENR approximates the average number of direct causes per node. By varying the ENR (including all lagged edges), we probe how algorithms perform under different connectivity and information densities. In this experiment, the maximum lag is fixed (see ii)). We consider two ENR regimes: (1) *sparse*, with ENR ≤ 2; and (2) *dense*, with $2 <$ ENR < 4.

We only considered lagged graphs in our simulation. This allows us to use cycles because a lag in edges prevents cyclicity. While it is possible to input contemporaneous effects for KarmaTS, we consider it a special case of a lag effect with a lower sampling frequency. For the functional maps, random multi-layer perceptrons (MLP) are employed.

We suggest that this paradigm could be enhanced by generating multivariate datasets with configurable spatio-temporal architectures. By allowing users to specify both the graphical model and the functional relationships between variables, we hope to enable the simulation of scenarios that better reflect real-world dynamics. For example, users might define spatial relationships, such as network-based or geographical dependencies, along with their temporal evolution patterns.

With the growing adoption of graphical model-based methodologies, there appears to be an increasing need for comprehensive benchmarking frameworks. Our proposed system aims to contribute to this space by providing a platform for evaluating time series forecasting and imputation algorithms using data that attempts to represent spatio-temporal complexities. Through the generation of data with adjustable characteristics, we hope that this work may help improve the assessment of algorithms across different scenarios, though further research would be needed to validate its effectiveness.

It is also possible to simulate datasets with latent variables. Figure 13(a) shows a simulated graphical model representing the causal structure of seven *observed* variables (A-G) and two *latent variables* (L0 & L1). The dashed arrows indicate the presence of a latent confounder influencing the system. No contemporaneous links are included in the model. Figure 13(b) shows the corresponding simulated MTS, which was generated using a 2-layer multilayer perceptron (MLP) with ReLU activation functions.

Appendix E. Factor-wise Analysis

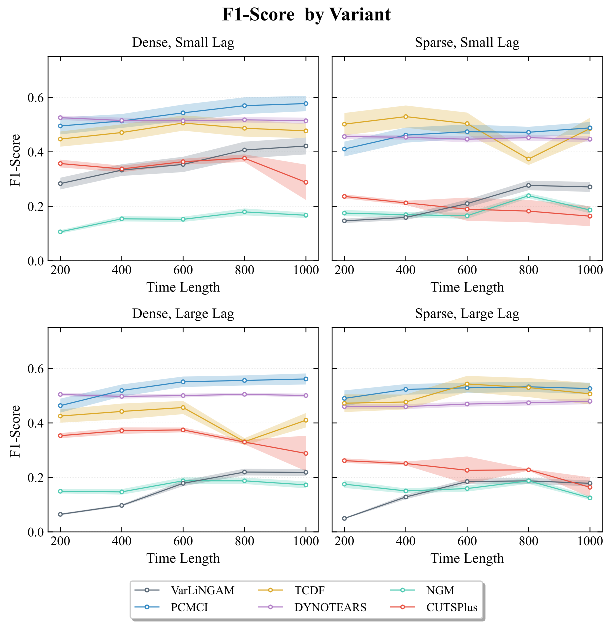


Figure 14: F1-score performance grouped by ENR and maximum lag of edges; results are averaged over all structural configurations (see Section D.2).

Time Series Length In the main text, Figure 6 shows F1 versus sequence length ($100-1000$). Constraint-based methods (**PCMCI**, **VarLiNGAM**) benefit most from longer series, gaining up to +0.25 F1, whereas gradient-based (**DYNOTEARS**) plateaus early once its acyclicity penalty stabilizes.

Benchmark / Setting	Reported strong methods (coarse ordering)
CAUSEME (stationary, moderate lag)	**PCMCI**, **VAR**-based and **NOTEARS**-style methods near top; deep models more variable (Runge et al., 2019a,b; Pamfil et al., 2020)
Synthetic irregular MTS (Neural ODE / **NGM**)	**NGM** competitive in highly nonlinear continuous-time settings, but less stable on simpler graphs (Bellot et al., 2021)
Attention-based **TCDF** studies	**TCDF** strong on some nonlinear benchmarks but not consistently dominant across all structures (Nauta et al., 2019)
High-dimensional graph benchmarks (**CUTS+**)	**CUTS+** improves scalability but may sacrifice edge-wise accuracy in dense or long-lag regimes (Cheng et al., 2024a)
KarmaTS summary (this work)	**PCMCI** and **DYNOTEARS** typically dominate F1/SID; **TCDF** excels in some star-like structures; **NGM** and **CUTS+** lag on most settings but remain competitive in selected regimes (e.g., sparse or highly nonlinear)

Table 3: Coarse external calibration of method rankings from KarmaTS to published time-series causal discovery benchmarks.

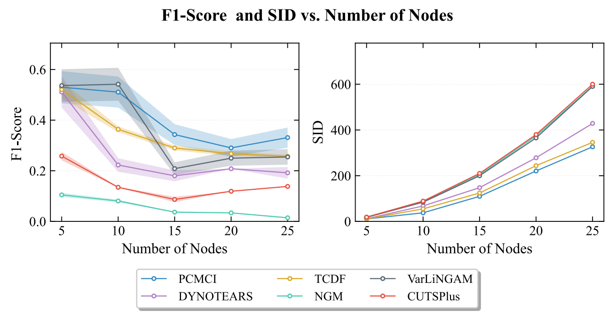


Figure 15: F1-score and SID for causal discovery methods, as a function of **the number of nodes** at a fixed time series length of 600. Formatting conventions are detailed in Section 7.

Edge Density and Maximum Lag Model performance also varies significantly with data density and time lag, as shown in Figure 14. For dense data, **PCMCI** consistently holds an advantage, achieving the highest F1-Score of ∼0.6 with a small lag and ∼0.55 with a large lag. In sparse conditions, however, the best-performing model changes. With a small lag, **TCDF** demonstrates a clear advantage with a peak F1-score of ∼0.55. With a large lag under sparse conditions, both **PCMCI** and **TCDF** emerge as the most robust choices. This again highlights that performance is not universal, but rather a function of specific data characteristics.

Number of Nodes The result is shown in Figure 15. As the graph size increases, F1-scores decline and SID rises precipitously. **PCMCI** preserves the highest F1 on smaller graphs, while **CUTS+** and **TCDF** exhibit comparatively greater robustness in SID.

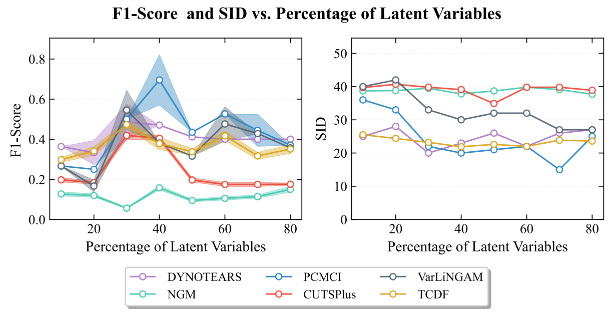


Figure 16: F1-score and SID for causal discovery methods. The time series length is fixed to be 600. Plot formatting details are given in Section 7.

Latent Variables Figure 16 plots F1 as the latent-variable fraction rises from 0% to 80%. The figure shows that the effect of latent variables on performance does not follow a simple monotonic trend: both F1-score and SID fluctuate as the percentage of latent variables increases. This behavior is consistent with prior findings that latent variables do not necessarily degrade or improve causal discovery, but can either obscure or reveal structure depending on the setting and assumptions (Gerhardus and Runge, 2020; Dong et al., 2024; Cai et al., 2023).

Appendix F. External Calibration to Prior Benchmarks

Table 3 provides a coarse alignment between the relative performance we observe on KarmaTS and method orderings reported in prior time-series causal discovery benchmarks, such as CAUSEME and follow-up studies (Runge et al., 2019a; Gong et al., 2024; Runge et al., 2019b; Pamfil et al., 2020; Nauta et al., 2019; Bellot et al., 2021; Cheng et al., 2024a). The goal is not to re-benchmark these methods, but to indicate which patterns appear consistent across settings versus those that appear specific to our stress tests.