LEARNING NETWORKED DYNAMICAL SYSTEMS VIA NEURAL ODE-BASED GRAPH STRUCTURE INFERENCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Inferring the latent interaction graph from multivariate time series is fundamental to understanding complex dynamical systems in neuroscience, biology, and physics. Despite recent progress in neural relational inference, most methods rely on discrete-time modeling, which degrades under sparse and irregular observations and tends to entangle multi-hop dependencies into spurious direct edges. We show that a continuous-time formulation naturally decouples inference from any fixed observation schedule, allowing gradient signals to be integrated continuously between observations for both dynamics and structure learning. We propose NOGI (**N**eural **O**DE-based **G**raph **I**nference), a framework that couples a deterministic graph learner with an MPNN-based Neural ODE. The two modules are trained end-to-end via the adjoint method with $\mathcal{O}(1)$ memory cost. Across four simulated dynamical systems at two scales, NOGI achieves the best structure recovery in 30 out of 32 evaluated cases with highly competitive forecasting performance, remaining robust under observation degradation. On the realistic Netsim fMRI benchmark, it attains an average AUPRC of 0.90 (AUROC 0.96). These results suggest that explicitly decoupling relational interactions from self-dynamics within a continuous-time vector field provides a powerful mechanism for graph structure learning under realistic observation conditions.

1 INTRODUCTION

Inferring the latent interaction graph from multivariate time series is a fundamental problem in network science, with direct applications in neuroscience (Bassett & Sporns, 2017), systems biology (Marbach et al., 2012), and physics (Timme & Casadiego, 2014). Relying solely on observed trajectories of coupled objects, the goal is to recover the underlying network topology that governs their interactions, without prior knowledge of the governing equations or the dynamical rules. While discrete-time models dominate the literature on neural relational inference, the dynamical systems they aim to model are almost always continuous.

In practice, observations are far from ideal. In fMRI-based brain network discovery (Tagliazucchi et al., 2012), recordings can be sparse and irregularly sampled due to sensor delays; in gene regulatory network reconstruction (Greenfield et al., 2010), expression profiles are often contaminated by measurement noise and available at only a handful of time points. Recovering the interaction graph under such conditions is critical for downstream scientific discovery, yet remains largely unsolved.

Existing approaches fall into three categories. *Classical statistical methods* such as Granger causality (Granger, 1969) and its nonlinear extensions (Tank et al., 2021) test whether one variable's history improves prediction of another, but multivariate conditioning is expensive and scales poorly. *Discrete-time neural methods*, including NRI (Kipf et al., 2018) and AIDD (Zhang et al., 2022), jointly learn graph structure and dynamics via differentiable graph sampling, but degrade under sparse and irregular sampling; moreover, their step-wise modeling often introduces spurious edges by mistakenly capturing multi-hop dependencies as direct connections. *Continuous-time neural methods* such as NGM (Bellot et al., 2022) and SCOTCH (Wang et al., 2024) leverage Neural ODEs or SDEs to model continuous dynamics with principled theoretical foundations, but either tightly couple graph topology with dynamics representation, which limits interpretability, or rely on computationally intensive variational inference whose training cost grows with the number of observed time points

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and solver steps. Across all three categories, no existing method jointly achieves continuous-time dynamics modeling, explicit graph inference, and scalability to long sequences and large networks.

Our key insight stems from a general form of coupled dynamical system (Eq. 1), which naturally isolates self-dynamics from topology-driven interactions, making the graph an explicit, learnable component of the continuous-time vector field. An ODE formulation further decouples the model from any fixed observation schedule, and the adjoint method accumulates gradient signal for each candidate edge continuously between consecutive observations, yielding structural supervision at $\mathcal{O}(1)$ memory cost. Building on this, we propose Neural ODE-based Graph Inference (NOGI), which couples two jointly trained modules: (i) a deterministic graph learner that directly optimizes a continuous edge-weight matrix, and (ii) a graph-conditioned Neural ODE whose vector field is parameterized by a message passing neural network, explicitly encoding the learned topology into the dynamics. (Fig. 1).

Our contributions are as follows:

- We propose NOGI, a framework that jointly infers latent interaction graphs and continuous-time dynamics by coupling a deterministic graph learner with a graph-conditioned Neural ODE, trained end-to-end via the adjoint method with $\mathcal{O}(1)$ memory cost.
- Across four simulated dynamical systems at two scales (N=10 and N=100), NOGI achieves superior structure recovery in 30 out of 32 comparisons. Furthermore, on the Netsim fMRI benchmark, it attains an average AUROC of 0.96 and AUPRC of 0.90 across 27 simulations.
- Evaluated under extreme observational constraints (including observation noise, sparse sampling, and severe data missingness), NOGI outperforms both discrete-time and continuous-time baselines across most tested conditions.
- The learned dynamics module successfully transfers across different network topologies and scales, demonstrating that NOGI effectively decouples intrinsic dynamical rules from structural inference.

2 METHOD

2.1 PROBLEM STATEMENT

Consider a coupled dynamic system with N objects, where the underlying connections can be represented by a directed graph $G = (V, E)$. V denotes the set of nodes, and E denotes the set of edges. The adjacency matrix $A \in \{0, 1\}^{N \times N}$ encodes the interaction between nodes, where $A_{ji} = 1$ indicates a directed edge from v_j to v_i. The time-varying feature of node v_i at time t is defined as $\mathbf{x}_i^t \in \mathbb{R}^d$ where d is the feature dimension.

System dynamics govern the temporal evolution of node features and can often be formally expressed as a system of Ordinary Differential Equations (ODEs). For instance, the classic Kuramoto system (Kuramoto, 1984) describes the synchronization of coupled oscillators via $d\theta_i/dt = \omega_i + K \sum_{j=1}^{N} A_{ji} \sin(\theta_j - \theta_i)$, $i = 1, \ldots, N$, where θ_i and ω_i are the phase and natural frequency of node v_i, respectively, and K denotes the constant coupling strength. The Kuramoto model indicates that the evolution of phase θ_i is not only driven by the node's own natural frequency ω_i, but also by accumulated phase differences from its neighbouring nodes. Following the same logic, a generalized formulation of complex network dynamics (Gao & Yan, 2022) can be derived as:

$$\frac{d\mathbf{x}_i}{dt} \;=\; \mathbf{F}(\mathbf{x}_i) + \sum_{j=1}^{N} A_{ji}\mathbf{G}(\mathbf{x}_i, \mathbf{x}_j), \quad i = 1, \ldots, N \tag{1}$$

$\mathbf{F}$ represents the self-dynamics, while $\mathbf{G}$ captures the interaction dynamics, which explicitly incorporates the influence of the network topology A. This formulation highlights that system dynamics can be decoupled into the underlying network topology and dynamical rules. Notably, the dynamical rules in many real-world systems may not strictly follow the form of Eq. 1, or may not even be expressed as ordinary differential equations (ODEs). More generally, we abstract the continuous system dynamics as:

$$\mathbf{X}^{t+\phi} \;=\; \Phi_\theta\left(\mathbf{X}^t, A; \phi\right) \tag{2}$$

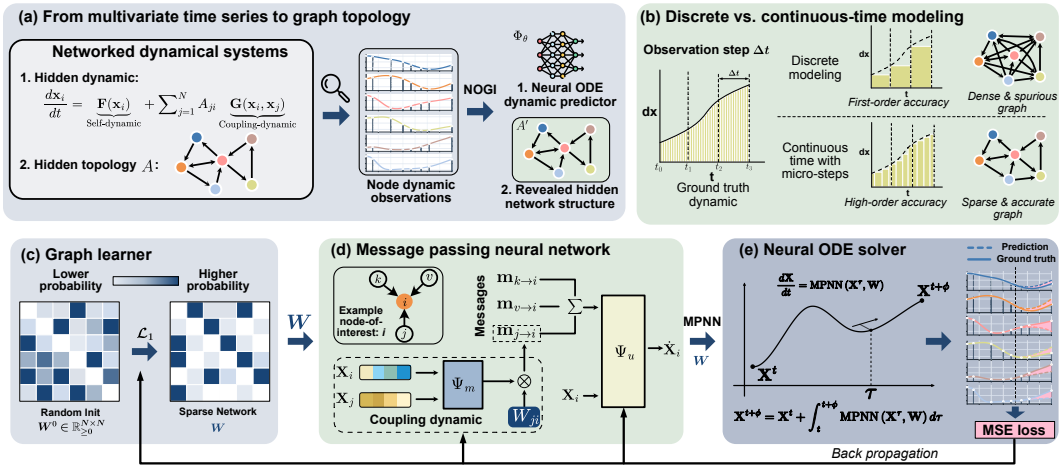

Figure 1: **Overview of the NOGI framework. (a) Problem formulation:** NOGI aims to infer the unknown graph topology and continuous dynamics from multivariate time-series observations of coupled dynamical systems. **(b) Motivation:** Continuous-time modeling ensures high-order accuracy and recovers sparse, accurate structures. **(c)-(e) Model pipeline:** The framework consists of a coupled graph learner and a dynamic learner. **(c) Graph Learner** infers a weighted adjacency matrix W, regularized by an $\mathcal{L}_1$ penalty. **(d) MPNN** parameterizes the vector field of node states over W, decoupling self- and coupling-dynamics. **(e) Neural ODE Solver** integrates states forward in time, trained end-to-end via adjoint method.

In this equation, the state of the entire system at time t is represented by the feature matrix $\mathbf{X}^t = [\mathbf{x}_1^t, \mathbf{x}_2^t, \ldots, \mathbf{x}_N^t]$. Φ_θ is an unknown function parameterized by θ, and $\phi > 0$ is the prediction horizon. In practice, we only have access to discrete-time observations. **Our goal is to jointly learn dynamical patterns Φ_θ and network topology A solely from observed trajectories $\mathbf{X}_{\text{hist}} = \{\mathbf{X}^{t_1}, \mathbf{X}^{t_2}, \ldots, \mathbf{X}^{t_T}\}$ at times $t_1 < t_2 < \cdots < t_T$.**

To achieve unsupervised learning of the network topology, our method is inspired by the principle of *Granger causality* (Granger, 1969): v_j exerts a directional influence on v_i if the history of v_j improves the prediction of v_i's future state. Under this interpretation, $A_{ji} = 1$ implies that v_j contributes to predicting v_i, and minimizing prediction error over observed trajectories naturally recovers the underlying graph structure. Formally, we optimize:

$$\min_{\theta,\, A} \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{X}^t, \mathbf{X}^{t+\phi}) \in \mathcal{D}} \left\| \mathbf{X}^{t+\phi} - \Phi_\theta(\mathbf{X}^t, A;\, \phi) \right\|^2 \tag{3}$$

where $\mathcal{D}$ denotes the set of training observation pairs. Notably, directly optimizing Eq. 3 is challenging. First, the binary constraint $A \in \{0,1\}^{N \times N}$ makes the objective non-differentiable. Second, the coupling between θ and A in $\Phi_\theta(\mathbf{X}^t, A)$ creates a challenging interdependence: accurate dynamics modeling requires knowing the true graph structure, yet inferring the correct structure requires an accurate dynamics model. Our method addresses these challenges through: (1) continuous relaxation of the discrete adjacency matrix, and (2) a Graph Neural ODE architecture that naturally decouples topology learning from dynamics modeling through message passing.

2.2 Model

Our model consists of two components that are jointly trained end-to-end (Fig. 1): a **Graph Learner** that infers the weighted adjacency matrix, and a **Dynamic Learner** that models the temporal evolution using Neural ODEs. NOGI accepts the snapshot of the dynamic system $\mathbf{X}^t$ as input. The key design principle is that the learned graph structure explicitly participates in the dynamics modeling through message passing, creating a direct gradient path from prediction error to graph structure.

2.2.1 GRAPH LEARNER

Theoretically, for a complex system with N nodes, the search space of discrete directed graph structures is $2^{N\times(N-1)}$, making heuristic optimization practically infeasible for large-scale networks. A common differentiable learning approach is to relax the binary entries in the adjacency matrix via the Straight-Through Estimator (Franceschi et al., 2019) or Gumbel-Softmax sampling (Kipf et al., 2018; Zhang et al., 2022), allowing for gradient-based optimization. While effective, they inject stochasticity into the forward pass or require a temperature schedule, which we find to interact poorly with the iterative ODE solver downstream.

We adopt a deterministic continuous relaxation, parameterizing the graph with a learnable matrix $\mathbf{W} \in \mathbb{R}_{\geq 0}^{N\times N}$ initialized from $\mathcal{U}(0,1)$. Entries W_{ji} denote continuous edge strengths rather than probabilities. Non-negativity is enforced via projected gradient descent ($W_{ji} \leftarrow \max(W_{ji}, 0)$). We impose no upper bound on $\mathbf{W}$ to help the optimizer escape poor local optima in the non-convex objective, relying instead on a sparsity regularizer to bound the magnitudes.

2.2.2 DYNAMIC LEARNER

Given the relaxed graph $\mathbf{W}$ produced by the graph learner, the dynamic learner models the unknown evolution operator Φ_θ in Eq. 2 that maps $\mathbf{X}^t$ to $\mathbf{X}^{t+\phi}$. Instead of directly regressing the future state, we parameterize the underlying topology-dependent *vector field* that governs the latent dynamics of nodes and obtain predictions by integrating it in continuous time:

$$\frac{d\mathbf{X}}{dt} \;=\; g_\theta\big(\mathbf{X}^t, \mathbf{W}\big)\,, \qquad \hat{\mathbf{X}}^{t+\phi} \;=\; \mathbf{X}^t + \int_t^{t+\phi} g_\theta(\mathbf{X}^s, \mathbf{W})\; ds. \tag{4}$$

The integral in Eq. 4 is computed via a numerical ODE solver, so the prediction horizon ϕ becomes a continuous variable that decouples the architecture from any fixed observation interval and naturally accommodates irregular sampling.

Message Passing Vector Field. The decomposition in Eq. 1 suggests that the vector field at each node should consist of a self-dynamics term $\mathbf{F}(\mathbf{x}_i)$ plus interaction dynamics $A_{ji}\,\mathbf{G}(\mathbf{x}_i, \mathbf{x}_j)$. We apply this inductive bias to a message passing neural network (MPNN), defining g_θ as the composition of a learnable message function Ψ_m and update function Ψ_u:

$$\mathbf{m}_{j\to i}^t \;=\; \Psi_m\big([\,\mathbf{x}_i^t \,\|\, \mathbf{x}_j^t\,]\big)\,, \qquad \mathbf{m}_i^t \;=\; \sum_{j=1}^{N} W_{ji}\,\mathbf{m}_{j\to i}^t, \tag{5}$$

$$\big[\,g_\theta\big(\mathbf{X}^t, \mathbf{W}\big)\big]_i \;=\; \Psi_u\big([\,\mathbf{x}_i^t \,\|\, \mathbf{m}_i^t\,]\big)\,, \tag{6}$$

where $\|$ denotes vector concatenation, and Ψ_m, Ψ_u are both four-layer multi-layer perceptrons (MLPs) with ReLU activations and a shared hidden width h. Here W_{ji} plays the role of A_{ji} in Eq. 1: it scales the message from v_j to v_i, and when $W_{ji} = 0$, v_j has no effect on the dynamics of v_i. The learned $\mathbf{W}$ can therefore be read out directly as the inferred graph. Meanwhile, by retaining $\mathbf{x}_i^t$ as a direct input to Ψ_u, the update step can model the self-dynamics even when no self-loop is present in $\mathbf{W}$.

Notably, because Ψ_m and Ψ_u are shared across all nodes and integration steps, the resulting vector field (learned system dynamics) is permutation-equivariant in the node ordering and applicable to networks of arbitrary size.

Adjoint Method for Joint Optimization. Equations 4–6 couple the graph $\mathbf{W}$ with the dynamics parameters θ inside an iterative ODE solver. Direct backpropagation through the solver requires storing every intermediate state, leading to a memory cost that scales linearly with the number of solver steps (Wang et al., 2024). We instead derive gradients with respect to both θ and $\mathbf{W}$ via the adjoint sensitivity method (Chen et al., 2018), reducing memory to $\mathcal{O}(1)$. The key insight is to treat $\mathbf{W}$ as an explicit differentiable parameter of the vector field g_θ, so that the graph gradient is integrated continuously over the prediction interval $[t, t+\phi]$ through the adjoint dynamics (full derivation in Appendix A.3):

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} = -\int_{t+\phi}^{t} (\mathbf{a}^s)^\top \frac{\partial\, g_\theta(\mathbf{X}^s, \mathbf{W})}{\partial \mathbf{W}}\, ds. \tag{7}$$

This allows a single prediction loss to jointly supervise both the dynamics parameters θ and the graph $\mathbf{W}$ through one shared backward pass, without requiring any multi-stage training procedure.

2.3 Training

Loss Function. Our training objective combines prediction accuracy and graph sparsity:

$$\mathcal{L}_{\text{total}} = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{X}^t, \mathbf{X}^{t+\phi}) \in \mathcal{D}} \left\| \mathbf{X}^{t+\phi} - \hat{\mathbf{X}}^{t+\phi} \right\|^2 + \lambda \|\mathbf{W}\|_1 \tag{8}$$

The first term, the prediction loss $\mathcal{L}_{\text{pred}}$, measures forecasting accuracy, reflecting the model's predictive capacity given the learned graph structure. The second term, the sparsity regularization $\mathcal{L}_{\text{sparse}} = \|\mathbf{W}\|_1$, penalizes unnecessary edges, biasing the model toward the essential graph structure that still explains the observed dynamics. The hyperparameter λ controls this trade-off. Implementation details, together with a sensitivity analysis of λ and a stability-based criterion for selecting it, are provided in Appendix A.4.

3 Experiments

Our experiments are designed to answer the following questions:

- **Q1**: Can NOGI recover ground-truth topology from diverse dynamical systems? (§3.2)
- **Q2**: Does NOGI learn the true underlying dynamics of the system? (§3.3, §3.5)
- **Q3**: Is NOGI robust to observation degradation? (§3.4)
- **Q4**: Are the learned dynamics transferable across different network topologies? (§3.5)
- **Q5**: Can NOGI generalize to realistic benchmarks beyond synthetic systems? (§3.6)

3.1 Experimental Setup

Dynamical systems. We evaluate on four synthetic coupled network systems that span a range of complexity: **Spring** (coupled oscillators, d=4), **Kuramoto** (phase oscillators, d=1), **FHN** (FitzHugh–Nagumo neurons, d=2), and **MM** (Michaelis–Menten gene regulation, d=1), where d denotes the per-node state dimension. For each system we generate trajectories on two scales (N=10 and N=100) over Erdős–Rényi (ER) and Barabási–Albert (BA) random graphs. Details of simulation settings are provided in Appendix A.2.

Baselines. NOGI is compared against baselines from three categories: (i) **Model-free Causal Measures**: Mutual Information (MI) and Partial Correlation (PC), non-parametric, learning-free metrics that infer causality based on statistical dependencies; (ii) **Universal Time Series Models**: MTGNN (Wu et al., 2020) and iTransformer (Liu et al., 2024), both general-purpose forecasting models; only MTGNN learns an adaptive graph and is thus also evaluated on structure inference; and (iii) **Neural Relation Inference Methods**, which jointly learn graph structure and dynamics: the discrete-time NRI (Kipf et al., 2018) and AIDD (Zhang et al., 2022), and the continuous-time NGM (Bellot et al., 2022) and SCOTCH (Wang et al., 2024), built on Neural ODEs/SDEs. Among these, SCOTCH is additionally evaluated as a state-of-the-art baseline on the realistic Netsim benchmark (§3.6).

Metrics. To bridge the gap between the continuous dataset-level learned weights $\mathbf{W}$ and binary ground-truth $\mathbf{A}$, we prioritize relative ranking and evaluate structure inference as a binary classification task using Area Under the ROC Curve (AUROC) and Area Under the Precision–Recall Curve (AUPRC). For forecasting dynamics, we report RMSE and MAE on the held-out test split.

Table 1: Structure inference performance (AUROC / AUPRC) on ER and BA graphs with N=10 and N=100 nodes. Values are means over three independent runs; standard deviations for the stochastic methods are reported in Table 7. **Bold**: best; underline: runner-up.

	$N = 10$								$N = 100$							
	Spring		Kuramoto		MM		FHN		Spring		Kuramoto		MM		FHN	
Method	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA
AUROC↑																
MI	0.95	0.94	0.94	0.80	**1.00**	0.94	0.76	0.34	0.86	0.59	0.91	0.90	0.99	0.98	0.52	0.62
PC	0.87	0.79	0.40	0.78	0.83	0.78	0.74	0.56	0.81	0.67	0.63	0.64	0.69	0.84	0.64	0.73
NRI	0.80	0.86	0.56	0.51	0.65	0.68	0.50	0.61	–	–	–	–	–	–	–	–
MTGNN	0.54	0.63	0.48	0.58	0.55	0.59	0.53	0.57	0.50	0.52	0.51	0.51	0.50	0.53	0.50	0.50
AIDD	**1.00**	**1.00**	0.78	0.80	**1.00**	**1.00**	0.96	0.61	**1.00**	**1.00**	0.93	0.83	**1.00**	**1.00**	0.58	0.33
NGM	**1.00**	**1.00**	0.53	0.60	0.98	0.88	0.87	**0.98**	0.70	0.95	0.57	0.63	0.94	0.98	0.54	0.82
SCOTCH	0.98	0.82	0.68	0.54	0.61	0.65	0.95	0.87	0.49	0.48	0.61	0.61	0.71	0.81	0.63	0.77
NOGI	**1.00**	**1.00**	**1.00**	**0.96**	**1.00**	**1.00**	**0.98**	0.79	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.98**
AUPRC↑																
MI	0.57	0.79	0.59	0.32	**1.00**	0.80	0.34	0.28	0.24	0.03	0.38	0.13	0.85	0.56	0.04	0.03
PC	0.23	0.32	0.40	0.31	0.23	0.29	0.31	0.20	0.12	0.06	0.07	0.06	0.09	0.14	0.07	0.08
NRI	0.61	0.71	0.16	0.23	0.14	0.35	0.21	0.40	–	–	–	–	–	–	–	–
MTGNN	0.28	0.41	0.29	0.40	0.26	0.35	0.40	0.41	0.12	0.13	0.14	0.13	0.12	0.13	0.13	0.11
AIDD	0.98	**1.00**	0.39	0.61	**1.00**	**1.00**	0.79	0.46	0.99	0.96	0.74	0.59	**1.00**	0.93	0.18	0.01
NGM	**1.00**	**1.00**	0.14	0.26	0.89	0.70	0.62	**0.94**	0.11	0.44	0.10	0.16	0.84	0.97	0.06	0.30
SCOTCH	0.75	0.46	0.19	0.31	0.25	0.33	0.90	0.70	0.04	0.02	0.08	0.03	0.16	0.37	0.14	0.22
NOGI	**1.00**	**1.00**	**1.00**	**0.94**	**1.00**	**1.00**	**0.91**	0.71	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.89**

3.2 Structure Learning Results on Synthetic Dynamical Systems

Table 1 summarizes unsupervised structure inference performance for all four synthetic dynamical systems across two scales (10/100) under both ER and BA network configurations. NOGI achieves the highest or jointly highest scores in **30 out of 32 comparisons**; at the more challenging N=100, it ranks first in all eight settings. The only two exceptions arise on FHN–BA at N=10, where NGM attains 0.98/0.94 AUROC/AUPRC versus 0.79/0.71 for NOGI. The evolution of network topologies during NOGI's training is visualized in Fig. 2.

Turning to the baselines, MI and PC often attain high AUROC but low AUPRC, especially at N=100: they separate true edges from non-edges on average but include many false positives among the top-ranked edges. MI is the exception on MM, where pairwise dependence closely reflects the topology. MTGNN learns an adaptive graph, but only as a latent dependency structure serving prediction, with no mechanism tying it to the underlying interactions; its recovery remains near-random. Among the discrete-time methods, AIDD is the strongest: it is nearly perfect on Spring and MM, yet underperforms on Kuramoto at both scales and collapses on FHN at N=100, where its AUPRC falls to 0.18 and 0.01. The continuous-time baselines are competitive on some systems at N=10, but degrade markedly at N=100: NGM's AUPRC exceeds 0.50 only on MM, and SCOTCH's falls to 0.02–0.37, while NOGI still holds up at this scale. The difference lies in how structure is represented: NGM encodes the graph implicitly in its network weights and SCOTCH infers it variationally, whereas NOGI learns an explicit graph decoupled from the dynamics. We attribute its scalable structure recovery to this decoupled representation.

Finally, NRI's low scores reflect a task mismatch. As a variational autoencoder, it infers a separate latent graph for each input trajectory. Our benchmark instead follows the protocol of NGM and SCOTCH, where all trajectories share a single sparse graph. Under this setting, NRI's VAE architecture offers no advantage, and NOGI outperforms it in all 16 comparisons at N=10; NRI does not scale to N=100 owing to its $\mathcal{O}(N^2)$ encoder cost.

3.3 Forecasting Performance

Beyond structure recovery, we ask whether the learned graph facilitates dynamics learning. Table 2 reports one-step-ahead forecasting errors on N=100 ER graphs against general-purpose forecasting

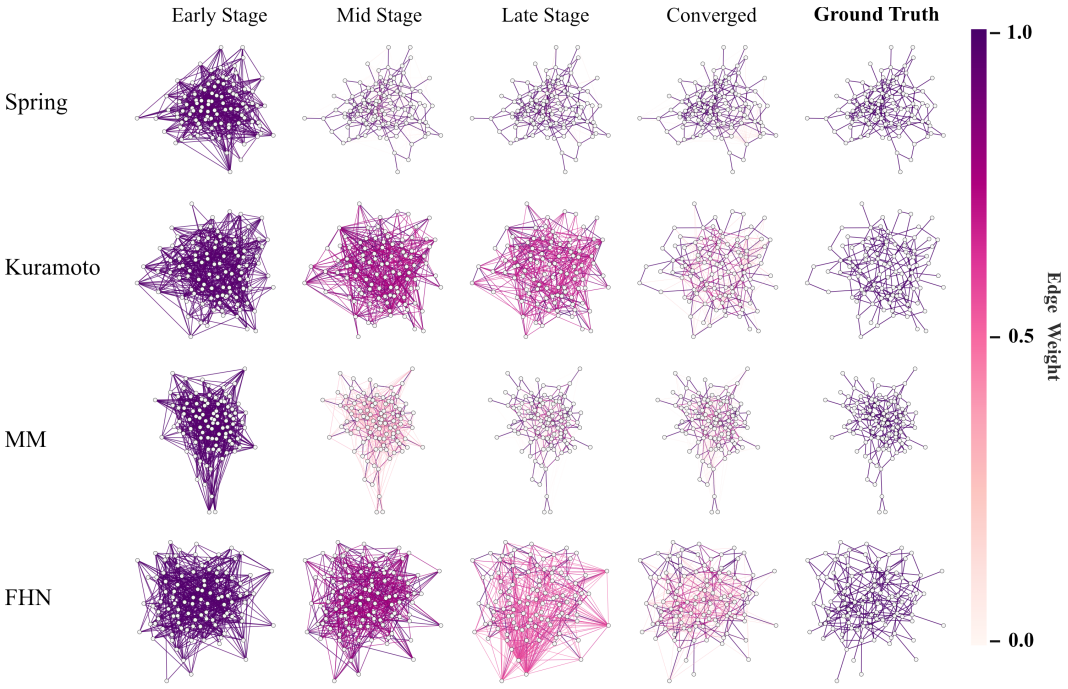

Figure 2: Evolution of learned network topologies during NOGI training (100-node ER graphs). The learned patterns converge toward the ground-truth graph, visually supporting the structure recovery performance reported in Table 1. Only the top 10% weighted connections are shown for clarity. Appendix A.6 provides the corresponding adjacency matrix heatmaps and further details on the learned topologies.

Table 2: Time series forecasting accuracy on ER graphs with $N{=}100$ nodes.

Dynamics	Metric	GRU	MTGNN	iTrans	AIDD	NODE	NGM	SCOTCH	**NOGI**	*NOGI w/ GT*
Spring	*RMSE* ↓	0.0245	0.0369	0.0127	**0.0055**	0.0203	0.0262	0.0241	0.0056	0.0073
	MAE ↓	0.0155	0.0223	0.0091	**0.0003**	0.0150	0.0192	0.0174	0.0010	0.0004
Kuramoto	*RMSE* ↓	0.7941	0.3046	0.0034	0.0027	0.0054	0.0056	0.0057	**0.0013**	0.0033
	MAE ↓	0.6594	0.2772	0.0020	0.0019	0.0025	0.0027	0.0024	**0.0008**	0.0018
MM	*RMSE* ↓	0.0945	**0.0027**	0.0044	0.0049	0.0635	0.0282	0.0324	0.0036	0.0105
	MAE ↓	0.0286	0.0017	0.0038	**0.0014**	0.0284	0.0212	0.0096	0.0017	0.0037
FHN	*RMSE* ↓	0.0663	0.0088	0.0047	0.0008	0.0036	0.0049	0.0032	**0.0004**	0.0005
	MAE ↓	0.0275	0.0075	0.0031	0.0005	0.0017	0.0022	0.0010	**0.0002**	0.0004

models (GRU, MTGNN, iTransformer) and relation-inference methods (AIDD, NGM, SCOTCH), plus two ablations: an unstructured Neural ODE (NODE) that replaces graph message passing with a global MLP shared across all nodes, and an oracle *NOGI w/ GT* that fixes the ground-truth adjacency $\mathbf{A}$ in place of the learned $\mathbf{W}$. NOGI attains the lowest RMSE and MAE on Kuramoto and FHN, ranks second on Spring and MM, and outperforms NGM, SCOTCH, GRU, and iTransformer on all four systems. MTGNN attains the lowest RMSE on MM despite weak structure recovery, showing that prediction accuracy alone does not guarantee a correct graph.

3.4 Robustness to Observation Degradation

Real-world multivariate time series often suffer from noise, low resolution, and irregular sampling. NOGI's continuous-time ODE addresses these challenges by inferring latent trajectories at arbitrary time points, requiring no architectural changes. We evaluate this robustness on $N{=}100$ ER graphs against the discrete-time AIDD and the continuous-time NGM under three controlled degradation scenarios (Fig. 3). **(a) Observation noise.** We inject additive Gaussian noise at five signal-to-noise ratios (SNR $\in \{30, 20, 15, 10, 5\}$ dB). On Spring and MM, NOGI degrades most gracefully and

consistently outperforms both baselines, maintaining 0.84 AUPRC on Spring even at SNR=5 dB, whereas AIDD fluctuates drastically across SNR levels. On Kuramoto, NOGI leads at 30 dB but falls slightly below AIDD at 20 dB, and at 15 dB and below no method recovers the structure. On FHN, recovery fails at every noise level for all three methods. **(b) Temporal down-sampling.** We uniformly retain every K-th observation ($K \in \{1, 2, 10, 20, 50\}$); MM is excluded because its trajectories contain only ten time points. NOGI attains the best or jointly best AUPRC at every K on all three systems, and the margin is most pronounced on Kuramoto and FHN. **(c) Irregular sampling.** We randomly drop 90–95% of the time points on Spring. NOGI preserves 0.87 AUPRC and RMSE < 0.22 even at 95% missingness, while AIDD's AUPRC drops to 0.39 and its RMSE rises to 0.44. NGM is omitted from this scenario because its training is numerically unstable under irregular sampling and fails in most runs.

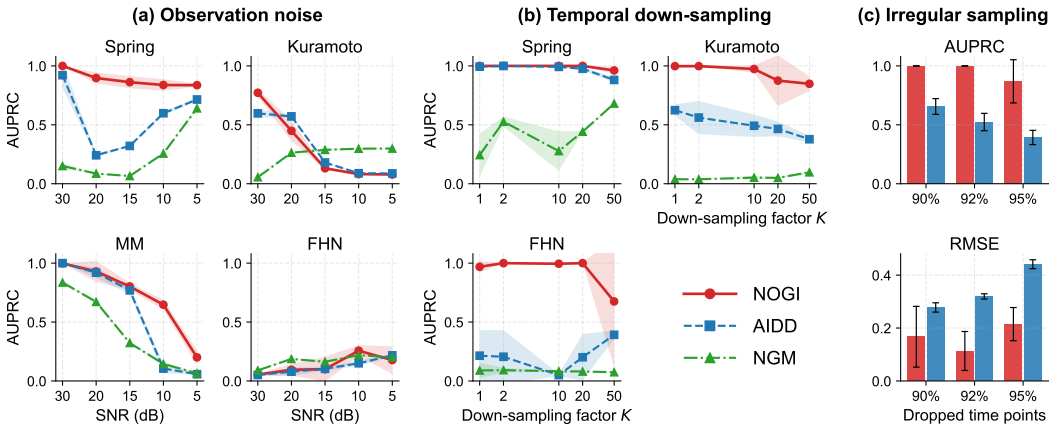

Figure 3: Model robustness under three types of observation degradation on N=100 ER graphs: **(a)** observation noise, **(b)** temporal down-sampling, and **(c)** irregular sampling. Shaded bands and error bars denote ± 1 s.d. over three seeds.

3.5 Transferability of Learned Dynamics

Table 3: Transferability of learned dynamics (FHN) from source ER graphs to target BA graphs. *None* (scratch) denotes training from scratch on the target, i.e., the NOGI FHN–BA results in Table 1.

		Source of g_θ		
Target Graph	Metric	None (scratch)	ER-10	ER-100
BA-10	*AUROC* ↑	0.7870	1.0000	0.9993
	AUPRC ↑	0.7104	1.0000	0.9970
BA-100	*AUROC* ↑	0.9797	0.9973	0.9806
	AUPRC ↑	0.8942	0.9639	0.8870

Drawing inspiration from Eq. 1, our proposed method naturally decouples the network topology from the system dynamics, ensuring that the learned dynamics are invariant to node permutation and network size. To demonstrate this capability, we further validate the transferability of the dynamics module in NOGI across diverse topologies (Table 3).

As noted in §3.2, FHN–BA remains the most challenging setting for NOGI at both scales. We freeze g_θ learned from FHN–ER time series and fine-tune only the graph learner on FHN–BA observations. Transfer matches or outperforms training from scratch in all settings. The results indicate that the learned dynamics module in NOGI transfers effectively across graphs of diverse topologies and network sizes, not only confirming that the model has captured the true underlying dynamics, but also suggesting a transfer-based route to structure recovery enabled by NOGI's decoupling of dynamics and topology.

Table 4: Structure Inference on the Netsim fMRI benchmark across all 28 simulations. Sim-13 is excluded from evaluation because its topology varies across subjects. NOGI scores are means over three independent runs; the corresponding standard deviations are reported in Table 8.

Method	Metric	Sim.													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
NOGI	*AUROC*↑	**1.00**	**0.99**	**0.99**	**0.97**	**1.00**	**1.00**	**1.00**	**0.98**	**1.00**	**1.00**	**0.98**	**0.99**	–	**0.99**
	AUPRC↑	**1.00**	**0.93**	**0.93**	**0.79**	**1.00**	**1.00**	**1.00**	**0.93**	**1.00**	**1.00**	**0.90**	**0.91**	–	**0.95**
SCOTCH	*AUROC*↑	0.67	0.92	0.92	0.93	0.81	0.89	0.93	0.73	0.81	0.91	0.77	0.87	–	0.67
	AUPRC↑	0.26	0.38	0.33	0.17	0.35	0.34	0.77	0.46	0.59	0.87	0.21	0.28	–	0.25

Method	Metric	Sim.													
		15	16	17	18	19	20	21	22	23	24	25	26	27	28
NOGI	*AUROC*↑	**0.96**	**0.88**	**1.00**	**1.00**	**0.98**	**1.00**	**1.00**	**0.87**	**0.84**	**0.90**	**0.99**	**0.81**	**0.89**	**1.00**
	AUPRC↑	**0.86**	**0.78**	**1.00**	**1.00**	**0.94**	**0.99**	**1.00**	**0.57**	**0.70**	**0.71**	**0.95**	**0.61**	**0.79**	**1.00**
SCOTCH	*AUROC*↑	0.41	0.60	0.89	0.77	0.74	0.67	0.75	0.56	0.32	0.63	0.69	0.42	0.47	0.70
	AUPRC↑	0.16	0.56	0.34	0.59	0.30	0.32	0.30	0.20	0.15	0.26	0.27	0.16	0.20	0.27

3.6 Realistic Neural Relation Inference Benchmark

We further evaluate NOGI on the Netsim benchmark (Smith et al., 2011), a widely used dataset for neural connectivity inference and causal discovery, comprising realistic fMRI simulations on synthetic brain networks with ground-truth **directed** connectivity. Netsim consists of 28 simulations spanning network sizes from $N{=}5$ to $N{=}50$ with diverse coupling configurations. The goal is to infer the underlying brain network from blood-oxygen-level-dependent (BOLD) time series.

As detailed in Table 4, NOGI delivers highly accurate connectivity recovery across the Netsim dataset, yielding an average AUROC of 0.96 and AUPRC of 0.90. Previous studies largely focus on the Sim3 subset, where NOGI achieves an AUROC of 0.99, providing a substantial margin over the recent continuous-time SCOTCH (Wang et al., 2024) (0.92). Beyond these quantitative gains, our approach demonstrates a clear qualitative advantage in recovering directed connections (Appendix A.6.3). By expanding the evaluation to all Netsim simulations, we further establish NOGI's strong generalization across varied network configurations, confirming its utility for realistic graph structure learning tasks.

4 Conclusion

We presented NOGI, a framework for joint graph inference and continuous-time dynamics learning that couples a deterministic graph learner with a graph-conditioned Neural ODE. Across synthetic and realistic benchmarks, NOGI improves structure recovery and remains robust under sparse, irregular, and noisy observations. These results suggest that separating self-dynamics from topology-driven interactions is an effective way to make the graph an explicit, learnable component of the vector field. Formulating this vector field in continuous time further decouples inference from any fixed observation schedule and accumulates gradient signal for each candidate edge continuously between observations.

Four structural limitations point to future work. First, NOGI assumes a static graph, which does not hold in evolving systems. Second, its deterministic ODE formulation cannot capture intrinsic stochasticity; incorporating latent SDEs (Wang et al., 2024) could model process noise and improve structural identifiability. Third, the dense $N \times N$ edge representation incurs $\mathcal{O}(N^2)$ cost, which limits scalability to very large networks; restricting the graph learner to a sparse set of candidate edges could lift this bottleneck. Finally, because all edges share a single message function Ψ_m and $\mathbf{W}$ is non-negative, NOGI assumes that every edge follows the same interaction mechanism, and thus cannot distinguish an edge that promotes the activity of its target from one that suppresses it. Learning a sign or a type label for each edge would relax this assumption.

AI USE STATEMENT

In this work, we used generative AI tools to assist with research execution, serving as a coding assistant during the implementation and debugging of our experiment code, and to assist with the derivation of the adjoint-based gradients in Appendix A.3. Additionally, we used generative AI tools to aid and polish the writing of the manuscript, and for literature retrieval to identify related work. We have not used generative AI tools for research ideation, experimental design, or the interpretation of results. We have reviewed all AI-assisted work: all AI-assisted code was reviewed and tested by the authors, all reported results were obtained by running the verified code, the derivations were checked step by step by the authors, and every cited work was verified against its original publication. We take responsibility for the final content of this work.

REFERENCES

Lei Bai, Lina Yao, Can Li, Xianzhi Wang, and Can Wang. Adaptive graph convolutional recurrent network for traffic forecasting. *Advances in neural information processing systems*, 33:17804–17815, 2020.

Danielle S Bassett and Olaf Sporns. Network neuroscience. *Nature neuroscience*, 20(3):353–364, 2017.

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*, 2022.

Zachary C Brown and David Carlson. Generating hypotheses of dynamic causal graphs in neuroscience: Leveraging generative factor models of observed time series. In *International Conference on Machine Learning*, pp. 5615–5648. PMLR, 2025.

Ben Chamberlain, James Rowbottom, Maria I Gorinova, Michael Bronstein, Stefan Webb, and Emanuele Rossi. Grand: Graph neural diffusion. In *International conference on machine learning*, pp. 1407–1418. PMLR, 2021.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. Spatial-temporal graph ode networks for traffic flow forecasting. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 364–373, 2021.

Luca Franceschi, Mathias Niepert, Massimiliano Pontil, and Xiao He. Learning discrete structures for graph neural networks. In *International conference on machine learning*, pp. 1972–1982. PMLR, 2019.

Ting-Ting Gao and Gang Yan. Autonomous inference of complex network dynamics from incomplete and noisy data. *Nature Computational Science*, 2(3):160–168, 2022.

Thomas Gaskin, Grigorios A Pavliotis, and Mark Girolami. Inferring networks from time series: A neural approach. *PNAS nexus*, 3(4):pgae063, 2024.

Colin Graber and Alexander G Schwing. Dynamic neural relational inference. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8513–8522, 2020.

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

Alex Greenfield, Aviv Madar, Harry Ostrer, and Richard Bonneau. Dream4: Combining genetic and dynamic information to identify biological networks and dynamical models. *PloS one*, 5(10): e13397, 2010.

Song Jiang, Zijie Huang, Xiao Luo, and Yizhou Sun. Cf-gode: Continuous-time causal inference for multi-agent dynamical systems. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 997–1009, 2023.

Wei Jin, Yao Ma, Xiaorui Liu, Xianfeng Tang, Suhang Wang, and Jiliang Tang. Graph structure learning for robust graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 66–74, 2020.

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.

Yoshiki Kuramoto. Chemical turbulence. In *Chemical oscillations, waves, and turbulence*, pp. 111–140. Springer, 1984.

Kai Lagemann, Christian Lagemann, Bernd Taschler, and Sach Mukherjee. Deep learning of causal structures in high dimensions under data limitations. *Nature Machine Intelligence*, 5(11):1306–1316, 2023.

Zhe Li, Andreas S. Tolias, and Xaq Pitkow. Learning dynamics and structure of complex systems using graph neural networks. In *A causal view on dynamical systems, NeurIPS 2022 workshop*, 2022. URL `https://openreview.net/forum?id=cmDMXmnykbo`.

Zhixun Li, Liang Wang, Xin Sun, Yifan Luo, Yanqiao Zhu, Dingshuo Chen, Yingtao Luo, Xiangxin Zhou, Qiang Liu, Shu Wu, et al. Gslb: The graph structure learning benchmark. *Advances in Neural Information Processing Systems*, 36:30306–30318, 2023.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

Daniel Marbach, James C Costello, Robert Küffner, Nicole M Vega, Robert J Prill, Diogo M Camacho, Kyle R Allison, Manolis Kellis, James J Collins, et al. Wisdom of crowds for robust gene network inference. *Nature methods*, 9(8):796–804, 2012.

Michael Poli, Stefano Massaroli, Junyoung Park, Atsushi Yamashita, Hajime Asama, and Jinkyoo Park. Graph neural ordinary differential equations. *arXiv preprint arXiv:1911.07532*, 2019.

Jakob Runge, Andreas Gerhardus, Gherardo Varando, Veronika Eyring, and Gustau Camps-Valls. Causal inference for time series. *Nature Reviews Earth & Environment*, 4(7):487–505, 2023.

Chao Shang and Jie Chen. Discrete graph structure learning for forecasting multiple time series. In *Proceedings of International Conference on Learning Representations*, 2021.

Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

Enzo Tagliazucchi, Pablo Balenzuela, Daniel Fraiman, and Dante R Chialvo. Criticality in large-scale brain fmri dynamics unveiled by a novel point process analysis. *Frontiers in physiology*, 3:15, 2012.

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.

Marc Timme and Jose Casadiego. Revealing networks from dynamics: an introduction. *Journal of Physics A: Mathematical and Theoretical*, 47(34):343001, 2014.

Benjie Wang, Joel Jennings, and Wenbo Gong. Neural structure learning with stochastic differential equations. In *The Twelfth International Conference on Learning Representations*, 2024.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 753–763, 2020.

Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In *International conference on machine learning*, pp. 10432–10441. PMLR, 2020.

Bocheng Zeng, Qi Wang, Mengtao Yan, Yang Liu, Ruizhi Chengze, Yi Zhang, Hongsheng Liu, Zidong Wang, and Hao Sun. Phympgn: Physics-encoded message passing graph network for spatiotemporal pde systems. In *The Thirteenth International Conference on Learning Representations*, 2025.

Yan Zhang, Yu Guo, Zhang Zhang, Mengyuan Chen, Shuo Wang, and Jiang Zhang. Universal framework for reconstructing complex networks and node dynamics from discrete or continuous dynamics data. *Physical Review E*, 106(3):034315, 2022.

A Technical appendices and supplementary material

A.1 Extended Related Work

Graph Structure Learning from Time Series. Inferring the underlying network topology from observed time series of coupled dynamical systems is a core problem in network science. Classical approaches based on Granger causality (Granger, 1969) and their nonlinear extensions (Tank et al., 2021) examine whether the predictability of one variable can be enhanced by other variables. Applying these methods to identify true direct causalities requires multivariate conditioning, which incurs a prohibitive computational complexity growing up to the fourth power of the network size and triggers a severe curse of dimensionality.

Some approaches incorporate stronger dynamical priors but impose restrictive assumptions: BP-PGM (Li et al., 2022) is limited to belief propagation dynamics on probabilistic graphical models, and Gaskin et al. (2024) requires the governing ODE to be known.

Other works aim to learn the underlying dynamics and network structure jointly. Neural Relational Inference (NRI) (Kipf et al., 2018) introduced a variational autoencoder (VAE) framework, demonstrating strong performance on simulated physical systems. Subsequent work extends this paradigm along several axes: DNRI (Graber & Schwing, 2020) allows the inferred graph to vary over time, ACD (Löwe et al., 2022) amortizes the inference to generalize across unseen systems, and AIDD (Zhang et al., 2022) proposes a more general framework with differentiable Gumbel-softmax sampling. However, these methods invariably rely on *discrete-time* dynamics models, introducing a fundamental mismatch with the continuous-time nature of the underlying interactions, which can bias the learned network structure.

NGM (Bellot et al., 2022) addresses this mismatch by leveraging Neural ODEs to model continuous system dynamics, naturally supporting irregular time series observations. It discovers sparse structures by applying adaptive group Lasso to per-node MLPs, thereby producing a representation in which dynamics and graph topology are tightly entangled. Leveraging latent Stochastic Differential Equations (SDEs), Wang et al. (2024) proposed SCOTCH, a Bayesian structure learning method that offers robust theoretical guarantees for structural identifiability under intrinsic stochasticity. Nevertheless, SCOTCH relies on variational inference over both graph structures and latent SDE trajectories; each training step solves an augmented SDE to estimate the path-wise KL term, making the training cost grow with the number of observed time points and solver steps. This limits its scalability to long sequential data in practice.

In contrast, our proposed NOGI takes a much simpler and more direct approach: it optimizes the adjacency matrix and models the system dynamics using a Neural ODE powered by a Message Passing Neural Network. Furthermore, by using the adjoint method for backpropagation, NOGI requires only $\mathcal{O}(1)$ memory in the number of solver steps. This design keeps training stable and allows NOGI to scale to large networks and long time series that typically bottleneck existing continuous-time methods.

Related efforts in spatio-temporal forecasting (Wu et al., 2019; 2020; Bai et al., 2020; Shang & Chen, 2021) learn soft adjacency matrices for dynamic prediction rather than recovering discrete ground-truth topology. Causal discovery from time series (Lagemann et al., 2023; Runge et al., 2023; Brown & Carlson, 2025) also infers inter-variable relationships from observations, but focuses on causal relations rather than coupling topologies of dynamical systems. Graph structure learning for node classification (Franceschi et al., 2019; Jin et al., 2020; Li et al., 2023) operates on static features without dynamics modeling.

Neural Ordinary Differential Equations with Graph Structure. Neural ODEs (Chen et al., 2018) parameterize continuous-time hidden state dynamics with a neural network, with the latent ODE formulation naturally suited for time series modeling. Several works have combined Neural ODEs with graph neural networks, broadly falling into two categories.

The first treats graph structure as given and focuses on improving GNN expressiveness through continuous-depth propagation: GODE (Poli et al., 2019) generalizes discrete GNN layers to continuous dynamics; CGNN (Xhonneux et al., 2020) designs continuous message passing from the perspective of graph diffusion equations; and GRAND (Chamberlain et al., 2021) further extends this to graph diffusion PDEs with learnable, time-varying edge weights via attention. The second category applies Graph Neural ODEs to spatio-temporal systems, e.g., traffic forecasting (Fang et al., 2021) and physical simulation (Zeng et al., 2025). CF-GODE (Jiang et al., 2023) extends GraphODE to continuous-time counterfactual inference in multi-agent systems via adversarial balancing, but assumes the interaction graph is known a priori.

Across both categories, the graph structure is assumed to be known. Our method removes this assumption by integrating differentiable graph structure learning into a Graph Neural ODE, jointly recovering both the interaction topology and the system dynamics.

A.2 Synthetic Dynamical System Details

This section details the synthetic data-generation protocol used in §3.1. We generate coupled multivariate time series on graphs with $N \in \{10, 100\}$ nodes. For each dynamical system and graph family, the sampled adjacency matrix is saved and used as the ground-truth topology for the evaluation of structure inference.

Graph generation. We use two random graph families in the main synthetic experiments: Erdős–Rényi (ER) and Barabási–Albert (BA). For all four synthetic systems, ER graphs are sampled with edge probability $p = 0.1$ for $N = 10$ and $p = 0.04$ for $N = 100$. BA graphs are generated with one edge attached per newly added node ($m = 1$). The resulting adjacency matrix is denoted by $\mathbf{A}$, and the degree used for normalized coupling is $k_i = \sum_j A_{ji}$, with isolated-node degrees clipped to one in the simulator to avoid division by zero.

Table 5: Synthetic data-generation settings. S denotes the number of simulated trajectories, T the number of saved time points per trajectory, and d the per-node state dimension used by the learner.

System	d	S	T	Solver	Step / sampling	Coupling
Spring	4	1	10000	leapfrog	$\Delta t = 10^{-3}$, save every 100 steps	$k = 0.1$
Kuramoto	1	200	1000	`odeint`	$\Delta t = 0.01$, horizon 10	$K = 5$
FHN	2	20	1000	`odeint`	$\Delta t = 0.01$, horizon 10	$K = 1$
MM	1	2000	10	`odeint`	$\Delta t = 1.0$, horizon 10	degree-normalized

Spring coupled oscillators. Each node represents a two-dimensional particle with position $\mathbf{r}_i \in \mathbb{R}^2$ and velocity $\mathbf{v}_i \in \mathbb{R}^2$, giving a per-node state dimension $d = 4$ after concatenating position and velocity. The dynamics follow Hooke-type pairwise interactions:

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{v}_i, \tag{9}$$

$$\frac{d\mathbf{v}_i}{dt} = -k \sum_{j=1}^{N} A_{ji}(\mathbf{r}_i - \mathbf{r}_j), \tag{10}$$

where $k = 0.1$. Initial positions are sampled from a Gaussian distribution with standard deviation 0.5, and initial velocities are normalized to norm 0.5. Particles evolve in a box of size 5 with elastic wall reflections. We integrate with a leapfrog scheme using internal step size 10^{-3} and retain one observation every 100 integration steps, resulting in 10000 saved time points. No observation noise is added during generation.

Kuramoto phase oscillators. Each node has scalar phase θ_i, so $d = 1$. The networked Kuramoto dynamics are

$$\frac{d\theta_i}{dt} = \omega_i + \frac{K}{k_i} \sum_{j=1}^{N} A_{ji} \sin(\theta_j - \theta_i), \tag{11}$$

with coupling strength $K = 5$. Natural frequencies are sampled as $\omega_i \sim \mathcal{N}(1.0, 0.3^2)$ and clipped to $[0.1, 2.0]$. Initial phases are sampled uniformly from $[0, 2\pi]$. For each graph, we generate 200 trajectories over horizon 10 with time step 0.01, yielding 1000 saved time points per trajectory.

FitzHugh–Nagumo neurons. Each node contains membrane potential v_i and recovery variable w_i, hence $d = 2$. The coupled FHN system is

$$\frac{dv_i}{dt} = v_i - \frac{v_i^3}{3} - w_i + I_i + \frac{K}{k_i} \sum_{j=1}^{N} A_{ji}(v_i - v_j), \tag{12}$$

$$\frac{dw_i}{dt} = \epsilon \left(v_i + a - b w_i\right), \tag{13}$$

where $K = 1$, $\epsilon = 0.08$, $a = 0.7$, and $b = 0.8$. The external input is heterogeneous across nodes: $I_i \sim \mathcal{N}(0.5, 0.1^2)$ clipped to $[0, 1]$, and is shared across trajectories generated on the same graph. Initial conditions are sampled as $v_i(0) \sim \mathrm{Unif}[-1, 1]$ and $w_i(0) \sim \mathrm{Unif}[-0.5, 0.5]$. We generate 20 trajectories over horizon 10 with time step 0.01, giving 1000 saved time points per trajectory.

Michaelis–Menten gene regulation. The MM system uses a scalar gene-expression state x_i at each node ($d = 1$). For node i, let $\mathcal{N}^-(i) = \{j : A_{ji} = 1\}$ denote its incoming regulators and $k_i = |\mathcal{N}^-(i)|$. The dynamics are

$$\frac{dx_i}{dt} = -x_i + \frac{1}{k_i} \sum_{j=1}^{N} A_{ji} \frac{x_j}{1 + x_j}, \tag{14}$$

with the interaction term omitted when $k_i = 0$. The initial state is sampled as $x_i(0) = 1 + u_i$, $u_i \sim \mathrm{Unif}[0, 1]$. We generate 2000 trajectories over horizon 10 with time step 1.0, giving 10 saved time points per trajectory.

A.3 Adjoint Derivation for Joint Graph and Dynamics Optimization

This appendix provides the complete derivation of the adjoint-based gradient computation for both the dynamics parameters θ and the graph $\mathbf{W}$.

Recall that the forward dynamics are given by:

$$\hat{\mathbf{X}}^{t+\phi} \;=\; \mathbf{X}^t + \int_t^{t+\phi} g_\theta(\mathbf{X}^s, \mathbf{W})\; ds. \tag{15}$$

Let $\mathcal{L}\left(\hat{\mathbf{X}}^{t+\phi}\right)$ denote the prediction loss. We define the adjoint state as the gradient of the loss with respect to the system state at time s:

$$\mathbf{a}^s \;=\; \frac{\partial \mathcal{L}}{\partial \mathbf{X}^s}. \tag{16}$$

The adjoint state satisfies a linear ODE that runs backward in time:

$$\frac{d\mathbf{a}^s}{ds} \;=\; -\,(\mathbf{a}^s)^\top \frac{\partial\, g_\theta(\mathbf{X}^s,\,\mathbf{W})}{\partial \mathbf{X}^s}. \tag{17}$$

This can be verified by differentiating the constraint $\mathbf{X}^{s+\epsilon} = \mathbf{X}^s + \epsilon\, g_\theta(\mathbf{X}^s, \mathbf{W})$ and applying the chain rule:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{X}^s} \;=\; \frac{\partial \mathcal{L}}{\partial \mathbf{X}^{s+\epsilon}} \left(\mathbf{I} + \epsilon\, \frac{\partial\, g_\theta}{\partial \mathbf{X}^s}\right), \tag{18}$$

which in the limit $\epsilon \to 0$ yields Eq. 17.

The key insight that enables joint learning is to treat $\mathbf{W}$ as an explicit differentiable parameter of the vector field g_θ. The gradients of θ and $\mathbf{W}$ are then obtained by integrating along the same backward trajectory:

$$\frac{\partial \mathcal{L}}{\partial \theta} \;=\; -\int_{t+\phi}^{t} (\mathbf{a}^s)^\top \frac{\partial\, g_\theta(\mathbf{X}^s,\,\mathbf{W})}{\partial \theta}\, ds, \tag{19}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}} \;=\; -\int_{t+\phi}^{t} (\mathbf{a}^s)^\top \frac{\partial\, g_\theta(\mathbf{X}^s,\,\mathbf{W})}{\partial \mathbf{W}}\, ds. \tag{20}$$

To see why Eq. 20 holds, note that $\mathbf{W}$ is constant across the integration interval (it does not evolve with s). At each instant s, the vector field $g_\theta(\mathbf{X}^s, \mathbf{W})$ depends on $\mathbf{W}$ both directly (through the weighted message aggregation $\sum_j W_{ji}\, \mathbf{m}_{j\to i}$) and indirectly (through the state $\mathbf{X}^s$, which itself was shaped by $\mathbf{W}$ at earlier times). The adjoint method accounts for both pathways: the indirect effect is captured by the adjoint state $\mathbf{a}^s$ (which propagates sensitivity through the state trajectory), while the direct effect is captured by $\partial g_\theta / \partial \mathbf{W}$.

In practice, the three ODEs — the forward state, the backward adjoint, and the parameter gradients — are solved jointly in a single backward pass using an ODE solver, with $\mathcal{O}(1)$ memory cost regardless of the number of solver steps.

A.4 Implementation and Training Details

NOGI uses adjoint sensitivity with either the `dopri5` or the RK4 solver. The L_1 sparsity weight λ is tuned per system over $\{10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}\}$, optionally following a decay schedule during training. We use two Adam optimizers: one for the graph learner parameters ($\mathbf{W}$) with learning rate $\alpha_g = 10^{-2}$, and one for the dynamic learner parameters (θ) with learning rate α_d set per system. All neural network parameters use Kaiming uniform initialization, and $\mathbf{W}$ is initialized from $\mathcal{U}(0, 1)$. After each gradient step, we clamp $W_{ji} \geq 0$ to maintain the non-negativity constraint. The exact settings for each system are provided in our released code.

The dataset is divided into training (60%), validation (20%), and testing (20%) subsets. Forecasting metrics are computed on the held-out test subset. Structure metrics instead compare the learned $\mathbf{W}$ against the ground-truth adjacency; since $\mathbf{W}$ is a single dataset-level parameter shared by all trajectories rather than a per-sample prediction, these scores are not tied to any split. All reported results are averaged over 3 independent runs with different random seeds. Models are trained on NVIDIA RTX 4090 GPUs with 24GB memory.

Sensitivity and ground-truth-free selection of λ. In practice, the underlying graph of a new dataset is unknown, so λ cannot be selected using structure-recovery metrics. We therefore evaluate a stability-based criterion that selects λ from cross-seed graph stability alone. For each seed, we retain the top 4% of directed off-diagonal entries of $|\mathbf{W}|$ as that seed's edge set; the stability score of a candidate λ is the mean pairwise Jaccard similarity between these edge sets across seeds, and the λ with the highest stability is selected. To validate this method, we apply it as a single fixed rule over the same grid $\lambda \in \{10^{-9}, 10^{-8}, 10^{-7}, 10^{-6}, 10^{-5}\}$ to the four synthetic systems on ER graphs with N=100 and to Sim3 of the Netsim benchmark, with three seeds per configuration. We load the ground-truth adjacency only after all stability scores have been computed and the selection has been frozen, solely to report the AUPRC of the selected λ. As shown in Table 6, sensitivity to λ is system-dependent; nevertheless, the blindly selected λ attains an AUPRC comparable to the best

value in the sweep in all five settings. These results provide empirical support for the criterion, but not a theoretical guarantee of its reliability on new datasets.

Table 6: Ground-truth-free selection of the sparsity weight λ via cross-seed top-4% Jaccard stability. **Bold**: the λ selected by the criterion.

Setting	Metric	10^{-9}	10^{-8}	10^{-7}	10^{-6}	10^{-5}
Spring	Top-4% Jaccard	0.93	0.97	**0.97**	0.17	0.93
	AUPRC	1.00±0.00	1.00±0.00	1.00±0.00	0.39±0.53	0.25±0.23
Kuramoto	Top-4% Jaccard	0.33	**0.95**	0.33	0.07	0.66
	AUPRC	0.69±0.43	1.00±0.00	0.69±0.53	0.45±0.48	0.22±0.26
MM	Top-4% Jaccard	0.66	0.55	0.58	**0.91**	0.18
	AUPRC	0.88±0.21	0.85±0.26	0.83±0.29	1.00±0.00	0.54±0.42
FHN	Top-4% Jaccard	**0.96**	0.89	0.33	0.13	0.75
	AUPRC	1.00±0.00	0.99±0.01	0.68±0.55	0.36±0.26	0.08±0.02
Netsim (Sim3)	Top-4% Jaccard	0.51	0.42	0.51	0.55	**0.55**
	AUPRC	0.89±0.03	0.88±0.02	0.87±0.03	0.88±0.03	0.89±0.01

A.5 Variability of Graph Structure Learning Results

Table 7 complements Table 1 by reporting the sample standard deviation of AUROC and AUPRC over three independent runs for all stochastic methods. MI and PC are deterministic and are therefore omitted.

Table 7: Standard deviations (AUROC / AUPRC) over three independent runs for the stochastic methods in Table 1, rounded to two decimals. Corresponding means are reported in Table 1. NRI is evaluated only at N=10.

	$N = 10$								$N = 100$							
	Spring		Kuramoto		MM		FHN		Spring		Kuramoto		MM		FHN	
Method	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA	ER	BA
AUROC																
NRI	0.21	0.07	0.03	0.01	0.01	0.08	0.04	0.12	–	–	–	–	–	–	–	–
MTGNN	0.04	0.01	0.07	0.06	0.12	0.03	0.06	0.14	0.01	0.01	0.00	0.02	0.00	0.03	0.02	0.02
AIDD	0.00	0.00	0.10	0.06	0.00	0.00	0.03	0.28	0.00	0.00	0.03	0.03	0.00	0.01	0.27	0.06
NGM	0.00	0.00	0.14	0.05	0.03	0.01	0.03	0.01	0.01	0.02	0.01	0.01	0.02	0.00	0.00	0.01
SCOTCH	0.00	0.02	0.01	0.02	0.10	0.04	0.01	0.00	0.02	0.01	0.01	0.01	0.01	0.04	0.01	0.02
NOGI	0.00	0.00	0.00	0.07	0.00	0.00	0.03	0.37	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
AUPRC																
NRI	0.24	0.07	0.02	0.01	0.00	0.06	0.01	0.12	–	–	–	–	–	–	–	–
MTGNN	0.04	0.05	0.09	0.03	0.09	0.02	0.09	0.14	0.01	0.01	0.00	0.02	0.00	0.03	0.02	0.01
AIDD	0.04	0.00	0.19	0.03	0.00	0.00	0.17	0.30	0.00	0.01	0.09	0.05	0.00	0.12	0.24	0.00
NGM	0.00	0.00	0.05	0.09	0.15	0.03	0.08	0.01	0.01	0.09	0.01	0.02	0.03	0.01	0.00	0.01
SCOTCH	0.07	0.03	0.01	0.04	0.09	0.06	0.01	0.03	0.00	0.00	0.01	0.00	0.01	0.18	0.02	0.08
NOGI	0.00	0.00	0.00	0.11	0.00	0.00	0.16	0.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.07

Table 8 reports the corresponding standard deviations for NOGI on the Netsim benchmark, over the three independent runs whose means are given in Table 4.

Table 8: Standard deviations of AUROC and AUPRC for NOGI over three independent runs on the Netsim benchmark, rounded to two decimals. The corresponding means are reported in Table 4. Sim-13 is excluded, as in Table 4.

Method	Metric	Sim.													
		1	2	3	4	5	6	7	8	9	10	11	12	13	14
NOGI	*AUROC*	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.02	0.00	0.00	0.02	0.00	–	0.02
	AUPRC	0.00	0.01	0.01	0.04	0.00	0.00	0.00	0.06	0.00	0.00	0.06	0.02	–	0.08
Method	Metric	Sim.													
		15	16	17	18	19	20	21	22	23	24	25	26	27	28
NOGI	*AUROC*	0.06	0.05	0.00	0.00	0.02	0.01	0.00	0.06	0.02	0.04	0.02	0.04	0.10	0.00
	AUPRC	0.21	0.11	0.00	0.00	0.05	0.02	0.00	0.10	0.02	0.01	0.05	0.03	0.18	0.00

A.6 Additional Structure Inference Visualizations

This section provides supplementary visualizations of the learned graph structures, complementing the network topology evolution shown in Fig. 2 and results reported in Tables 1 and 4.

A.6.1 Adjacency Matrix Heatmaps (NOGI, ER, N=100)

Fig. 4 displays the learned edge-weight matrices on ER graphs at selected training epochs of NOGI. As training progresses, the off-diagonal structure increasingly concentrates on the ground-truth edges, consistent with the near-perfect AUROC/AUPRC scores in Table 1.

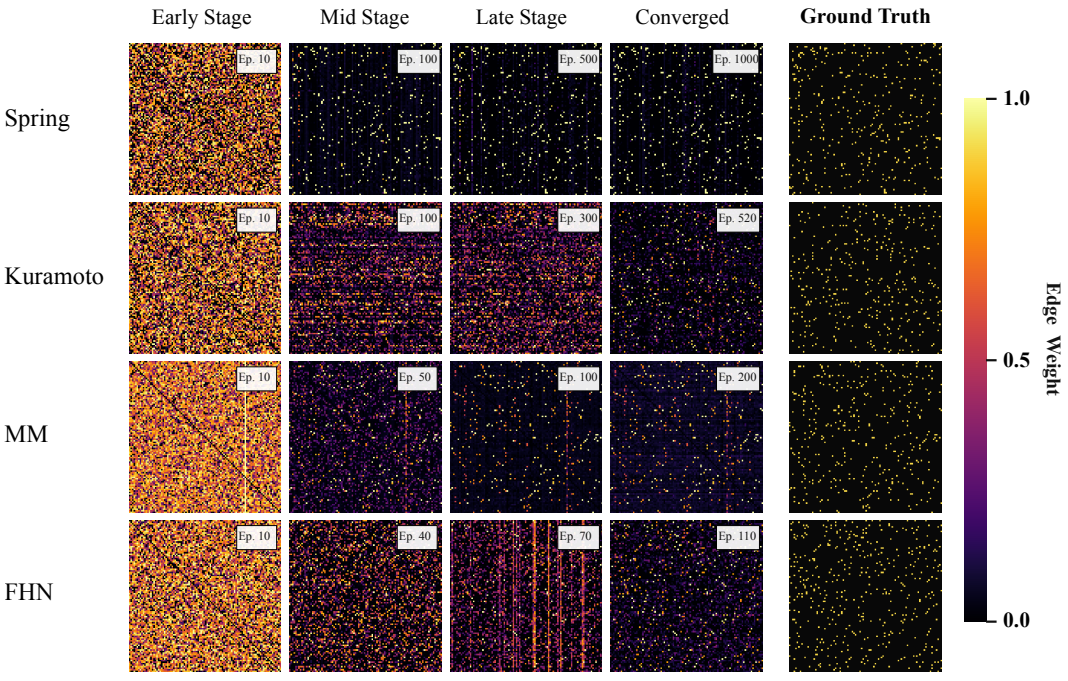

Figure 4: Learned edge weights on ER graphs with N=100. Weights are clamped at the 99th percentile and rendered using power normalization (γ=0.5). *Ep* stands for Epoch.

A.6.2 Network Topologies Inferred in Synthetic Systems

We visualize the recovered network structures behind Table 1. Figs. 5 – 8 compare AIDD, NGM, SCOTCH, and NOGI against the ground-truth graph across Spring, Kuramoto, MM, and FHN dynamics. Edge color and width indicate inferred connection strength. Each panel shows the K

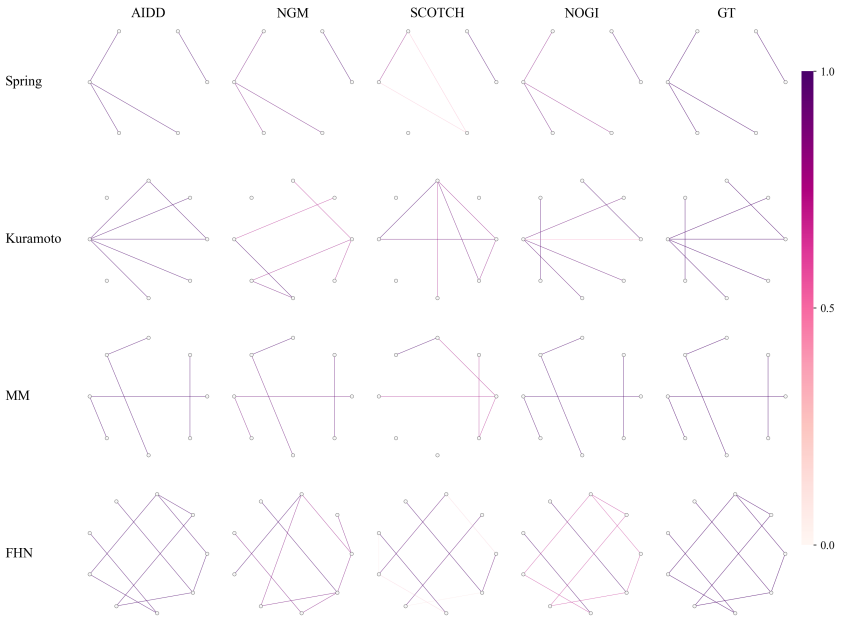

Figure 5: Recovered network structures on ER graphs with $N = 10$.

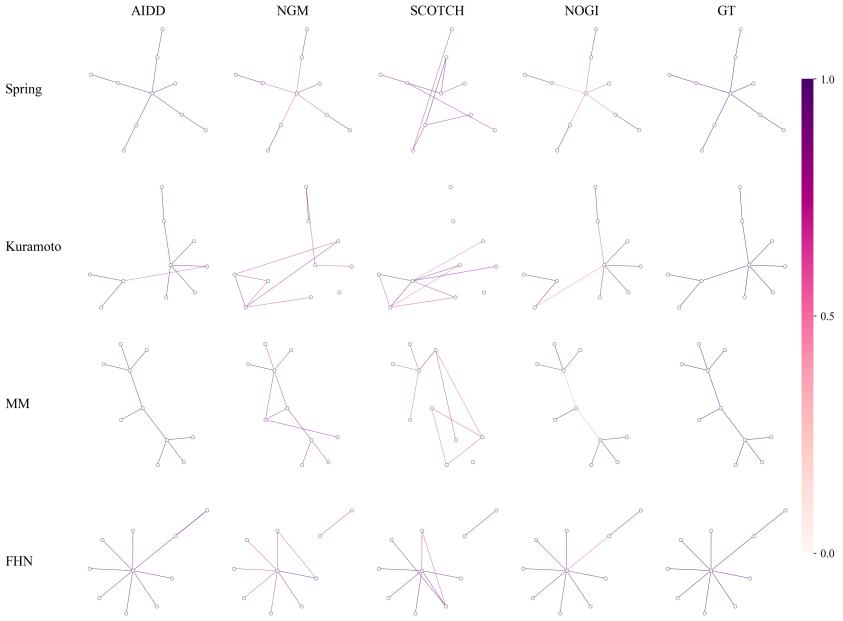

Figure 6: Recovered network structures on BA graphs with $N = 10$.

highest-weight inferred edges, where K is the number of ground-truth edges, so that all methods are displayed under the same edge budget.

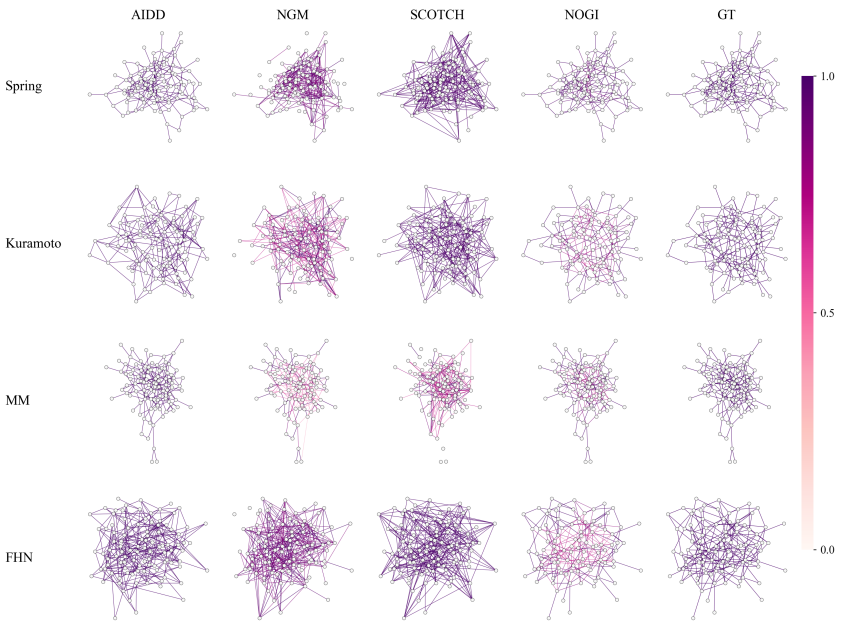

Figure 7: Recovered network structures on ER graphs with $N = 100$.

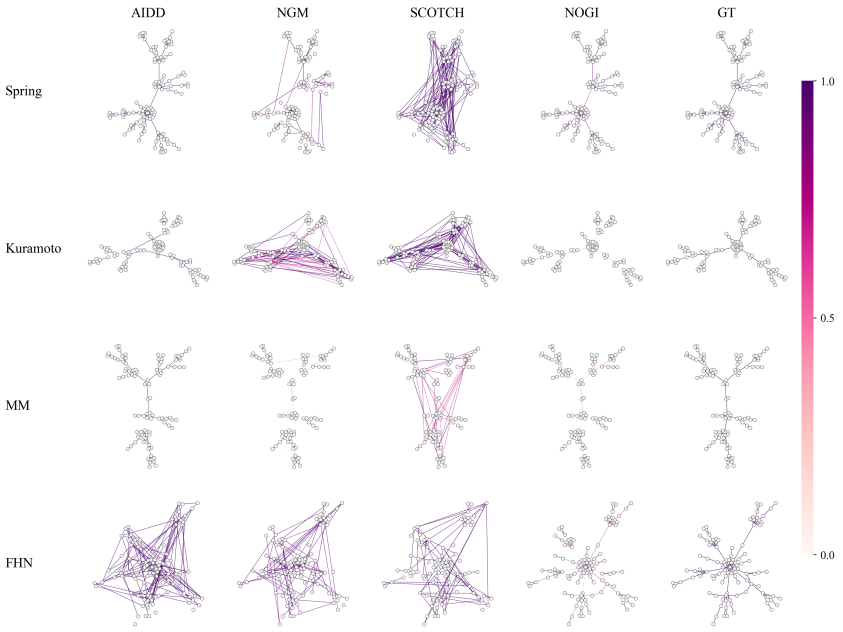

Figure 8: Recovered network structures on BA graphs with $N = 100$.

A.6.3 NETWORK TOPOLOGIES INFERRED IN NETSIM BENCHMARK

To supplement the results presented in Table 4, we further provide visualizations of the recovered network structures for the Netsim benchmark (see Figs. 9, 10, 11, 12). The 27 evaluated simulations are categorized into four groups according to the network size. `GT` denotes the ground truth. To ensure visual clarity, we plot only the top 30%, 20%, 15%, and 5% of edges with the highest weights for networks with $N = 5, 10, 15$, and 50 nodes, respectively. SCOTCH frequently assigns edges the wrong orientation, whereas NOGI more reliably recovers the direction of connections.

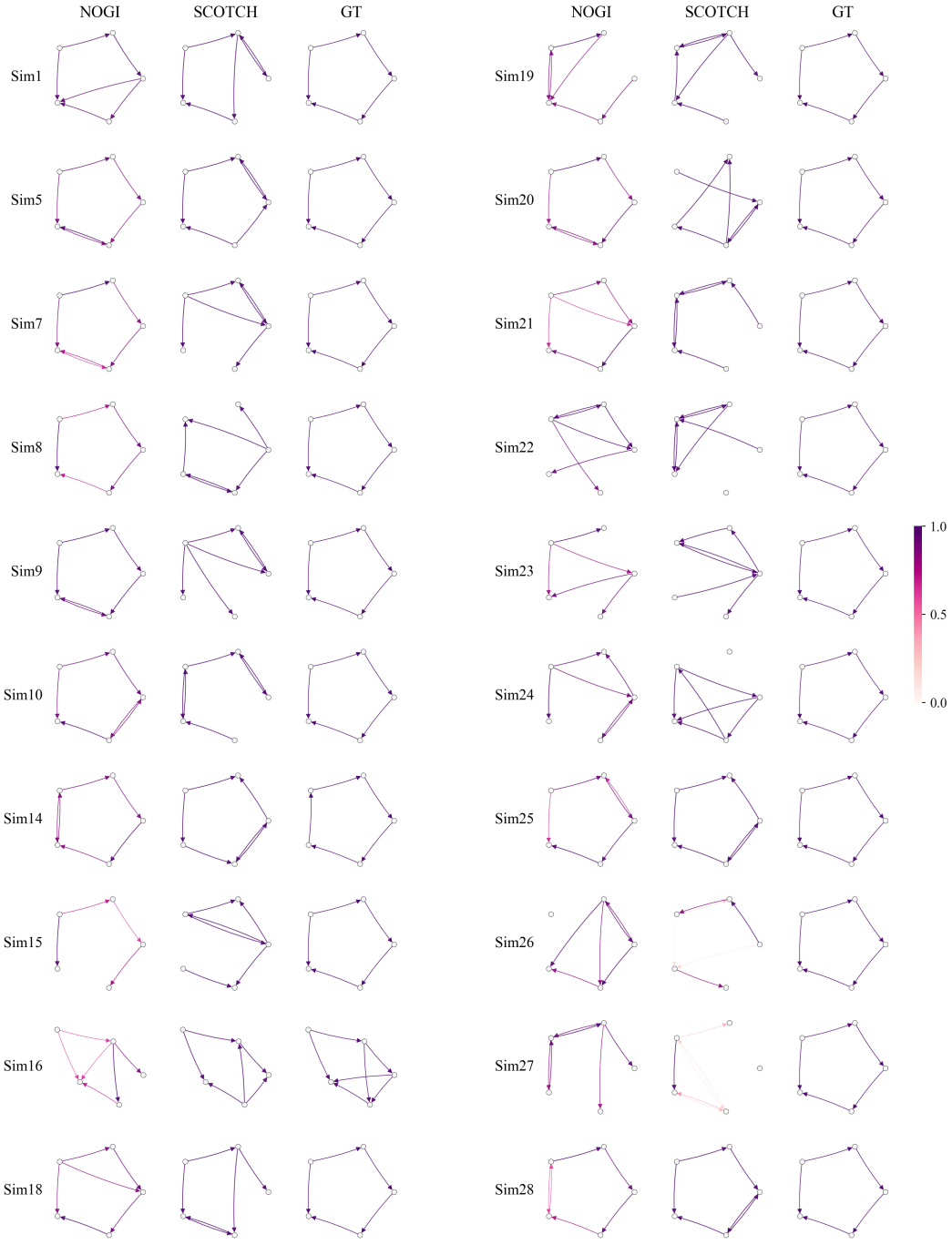

Figure 9: Recovered network structures on Netsim Benchmark with $N = 5$.

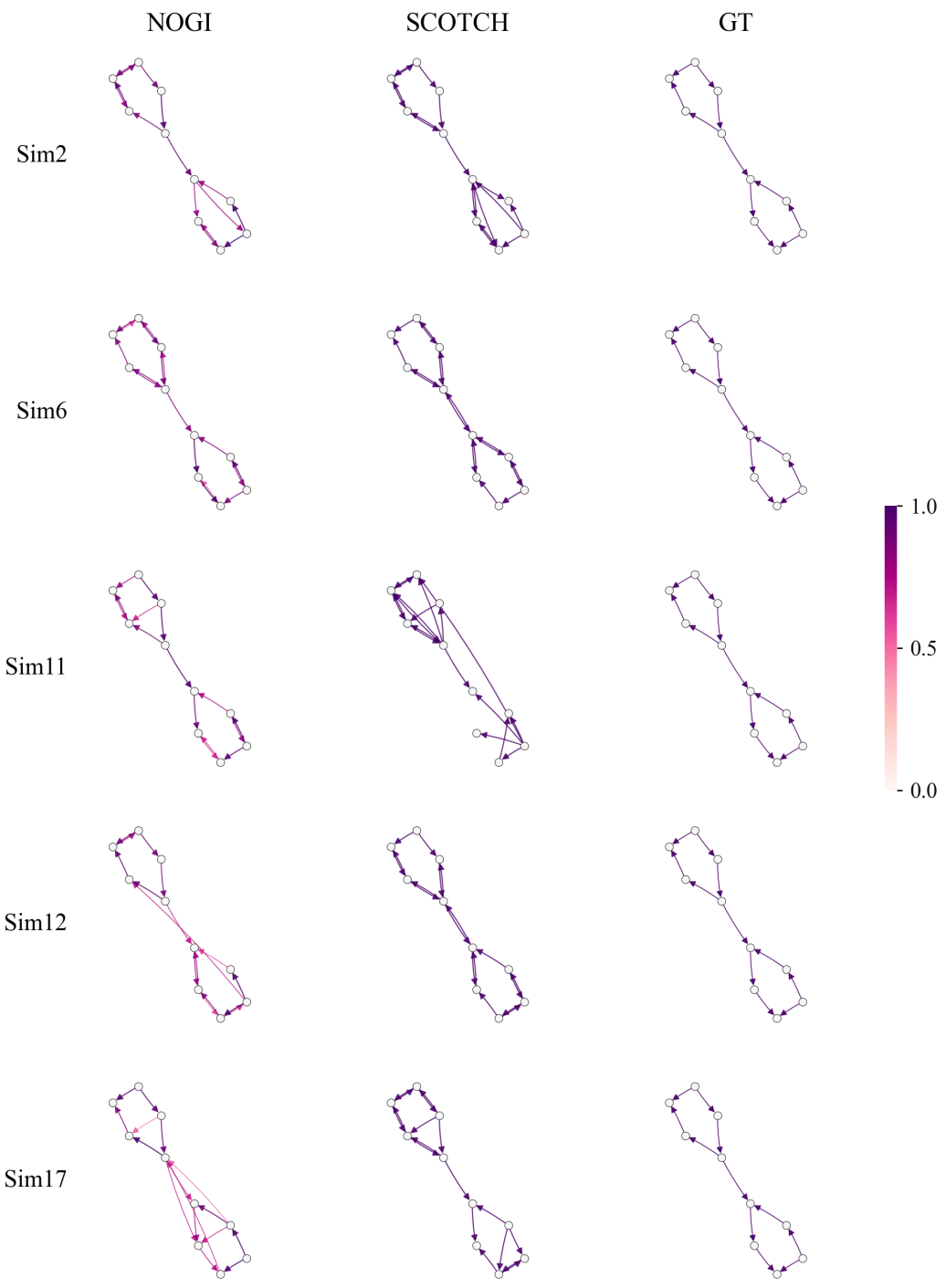

Figure 10: Recovered network structures on Netsim Benchmark with $N = 10$.

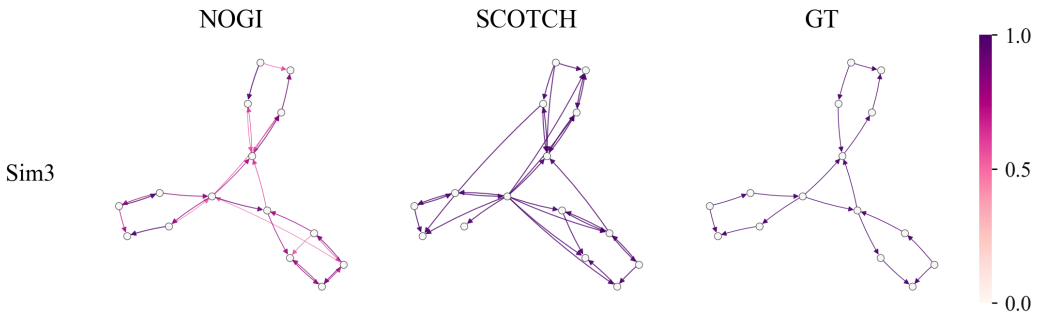

Figure 11: Recovered network structures on Netsim Benchmark with $N = 15$.

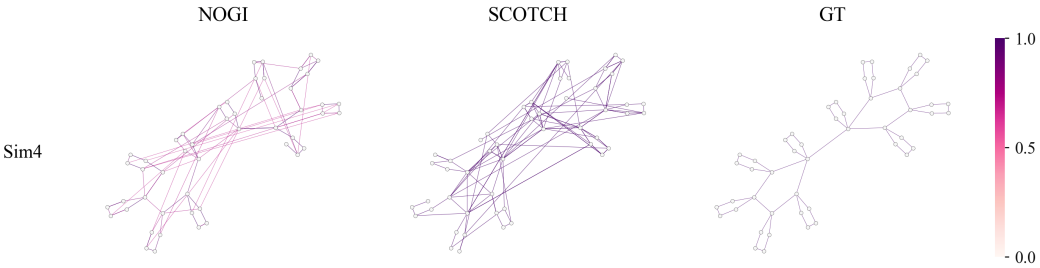

Figure 12: Recovered network structures on Netsim Benchmark with $N = 50$.