A Graph Dynamics Prior for Relational Inference

Liming Pan[1, 2*], **Cheng Shi**[3*], **Ivan Dokmanić**[3†]

[1]School of Cyber Science and Technology, University of Science and Technology of China, Hefei, China 230026,
[2]School of Computer and Electronic Information, Nanjing Normal University, China 210023,
[3] Departement Mathematik und Informatik, Universität Basel, Basel, Switzerland 4051
panlm99@gmail.com, cheng.shi@unibas.ch, ivan.dokmanic@unibas.ch

Abstract

Relational inference aims to identify interactions between parts of a dynamical system from the observed dynamics. Current state-of-the-art methods fit the dynamics with a graph neural network (GNN) on a learnable graph. They use one-step message-passing GNNs—intuitively the right choice since non-locality of multi-step or spectral GNNs may confuse direct and indirect interactions. But the effective interaction graph depends on the sampling rate and it is rarely localized to direct neighbors, leading to poor local optima for the one-step model. In this work, we propose a graph dynamics prior (GDP) for relational inference. GDP constructively uses error amplification in non-local polynomial filters to steer the solution to the ground-truth graph. To deal with non-uniqueness, GDP simultaneously fits a "shallow" one-step model and a polynomial multi-step model with shared graph topology. Experiments show that GDP reconstructs graphs far more accurately than earlier methods, with remarkable robustness to under-sampling. Since appropriate sampling rates for unknown dynamical systems are not known a priori, this robustness makes GDP suitable for real applications in scientific machine learning. Reproducible code is available at https://github.com/DaDaCheng/GDP.

Introduction

Understanding interactions is key to understanding the function of dynamical systems in physics (Arenas et al. 2008), biology, neuroscience (Izhikevich 2007), epidemiology (Pastor-Satorras et al. 2015), and sociology (Castellano, Fortunato, and Loreto 2009), to name a few. It is often time-consuming or even impossible to determine this structure experimentally: for example, neuronal connectivity is determined by painstaking analyses of electron microscopy images. On the other hand, there has been an explosion of availability of signal measurements. It is thus attractive to devise methods which determine interactions from the observed dynamics alone.

The seminal work on neural relational inference (NRI) showed that in some cases graph neural networks (GNNs) can perform well on this challenge (Kipf et al. 2018). GNN-based methods use a graph generator and a dynamics learner[1]: the graph generator produces a candidate graph while the dynamics learner tries to match the dynamics to data, acting as a surrogate for the original system. Since the dynamics of a node only depends on its neighbors, it is intuitive to try and emulate it with a single-step message-passing GNN. Multi-step message passing or spectral GNNs may confuse direct and indirect neighbors. Indeed, single-step architectures appear in the original NRI work and its various adaptations (Alet et al. 2019; Löwe et al. 2022; Graber and Schwing 2020; Ha and Jeong 2023; Zhu et al. 2022; Zhang et al. 2022; Wang and Pang 2022).

An implicit assumption in a single-step scheme is that the sampling rate (the number of samples per unit time) is sufficiently high. With a low sampling rate the effective interaction graph is non-local which causes a single-step surrogate to confuse direct and indirect interactions. Single-step message passing also limits the expressivity of neural surrogates. The vanilla GCN (Kipf and Welling 2017) and many other GNNs implicitly assume homophily and act as low-pass graph filters (Zhu et al. 2020). Although they can handle certain heterophilic data (Ma et al. 2022), single-layer GNNs can only implement a limited range of graph filters. Unknown nonlinear dynamics call for graph filters adaptive to data such as in ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) or GPR-GNN (Chien et al. 2021).

In this work, we propose a *graph dynamics prior* (GDP) for relational inference. The "prior" terminology is an analogy with the *deep image prior* used in inverse problems in image processing (Ulyanov, Vedaldi, and Lempitsky 2018), where the inductive bias of model (a convolutional neural network) steers reconstruction towards images with good properties, which is an implicit prior. Our model simultaneously uses a high-degree non-local polynomial and a "shallow" adjacency matrix to approximate the effective interactions between consecutive state samples. The polynomial filter is sensitive to graph perturbations which helps avoid poor local minima. As there are, in general, multiple graph matrices that can result in the same polynomial filters, simultaneously fitting a parallel single-step model resolves the direct interactions without converging to poor local minima

*These authors contributed equally.
†To whom correspondence should be addressed.

[1]Even though they are sometimes called differently, these two components are always present.

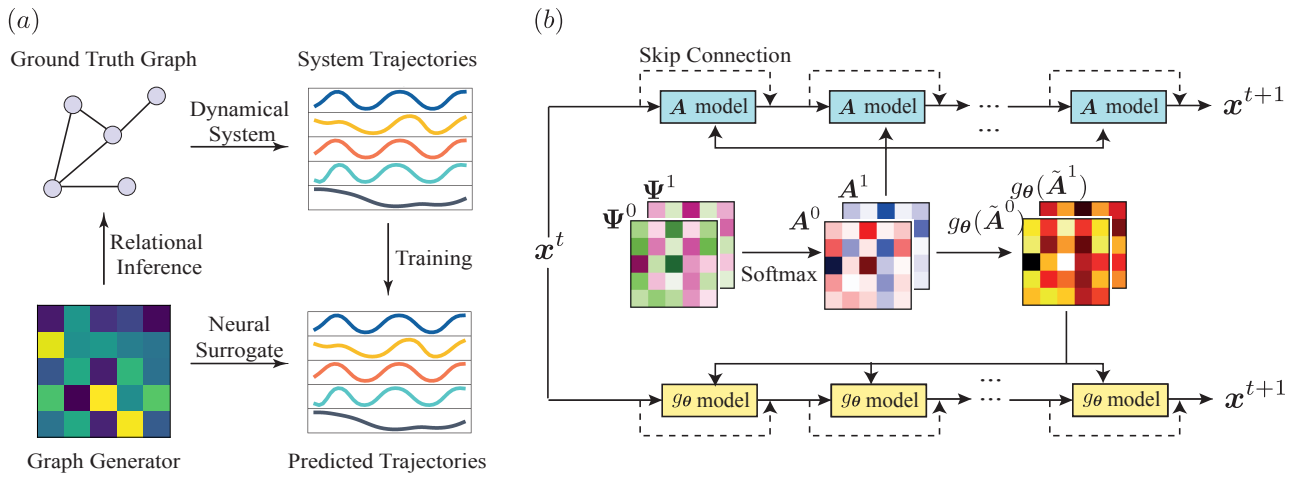

Figure 1: (a) An illustration of the relational inference problem. (b) The architecture of the full model. We train two dynamics surrogates with shared topology. The detailed architecture of $\mathbf{A}$ model and $g_{\boldsymbol{\theta}}$ model can be found in Section .

thanks to the gradients from the polynomial model; this resolves issues which traditionally prevented the use of polynomial filters in NRI. Experiments show that GDP achieves significantly higher inference accuracy than any of the earlier approaches. Notably, it finds the direct interactions even at very low sampling rates where earlier approaches severely degrade.

Related Work

Relational Inference. Classical approaches to relational inference (RI) measure correlations (Peng et al. 2009), mutual information (Wu et al. 2020), transfer entropy (Schreiber 2000) or causality (Quinn et al. 2011) of system trajectories. These approaches do not perform future state predictions. Other studies focus on RI with known dynamical models (Wang, Lai, and Grebogi 2016; Pouget-Abadie and Horel 2015). These are often designed for particular dynamics. When only system trajectories are observed, NRI (Kipf et al. 2018) infers the interaction graph in an unsupervised way while simultaneously predicting the state evolution. The scope of NRI has been extended to dynamic graphs (Graber and Schwing 2020), graphs with heterogeneous interactions (Ha and Jeong 2023), and modular meta-learning problems (Alet et al. 2019). It has also found applications in learning protein interactions (Zhu et al. 2022) and was adapted to make causal claims (Löwe et al. 2022). The model has been extended to the non-amortized setting, where the graph encoder is often removed (Löwe et al. 2022; Zhang et al. 2022). The dynamics learner part of these models is a GNN with one-step message passing.

Spectral Graph Neural Networks. Two basic design paradigms for GNNs are via graph (spatial) and spectral graph convolutions. "Spatial" graph convolution aggregates neighborhood information; spectral filters are generically non-local (Ortega et al. 2018). ChebyNet (Defferrard, Bresson, and Vandergheynst 2016) parameterizes the convolution kernel by Chebyshev polynomials of the diagonal matrix of Laplacian eigenvalues. Two other GNNs that use polynomial graph filters are APPNP (Gasteiger, Bojchevski, and Günnemann 2018) and GPR-GNN (Chien et al. 2021). By making the weights of matrix polynomials trainable, GPR-GNN adaptively implements low-pass or high-pass graph filters. For relational inference, spatial GNNs more straightforwardly preserve locality via one-step message-passing and thus have been widely considered in different methods.

Our Contributions

Contrary to prior belief, we show that RI by non-localized filters can work much better than shallow alternatives. While earlier work avoids direct–indirect confusion via one-step message-passing, it suffers from local minima even in the presence of weak indirect interactions which prevents it from fitting the dynamics. We conjecture and empirically demonstrate that a non-local model resolves this issue, provided that it is properly designed. Concretely, to mitigate the ambiguities arising from rooting matrix polynomials which prevented earlier uses of multi-step architectures, we run in parallel a local, single-step model with a shared adjacency matrix, but now benefiting from the "steering" by the multi-step model. This effectively results in a multi (two)-scale architecture. In addition to yielding better graphs, this strategy yields a much better model for the dynamics and greatly reduces the number of samples required to learn the graph. The two-scale architecture brings a remarkable performance improvement across the board.

Preliminaries

We consider a graph $G = (V, E)$ on $|V| = n$ vertices, which describes the interaction relations among components of a dynamical system. Let $\mathbf{A}$ be the adjacency matrix, $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ its symmetric normalized version, and $\tilde{\mathbf{L}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ the symmetric normalized Lapla-

Model	Graph	δt	Volume	MI	TE	NRI	GDP
Michaelis Menten	ER-50	1	50×10	77.84	53.66	54.09±2.22	**98.31±1.41**
		4	50×10	55.93	51.95	51.85±0.97	**88.66±8.73**
	BA-50	1	50×10	88.04	63.96	55.20±1.81	**93.02±3.94**
		4	50×10	50.18	60.30	52.26±1.23	**87.42±6.08**
Rössler Oscillators	ER-50	1	50×10	50.65	54.17	60.35±4.58	**99.89±0.23**
		4	50×10	52.28	54.13	51.95±1.28	**56.82±3.56**
	BA-50	1	50×10	56.28	62.64	59.81±5.74	**90.55±16.13**
		4	50×10	50.46	52.63	52.42±1.76	**59.22±5.24**
Diffusion	ER-50	1	20×10	56.00	57.63	70.66±7.80	**93.44±4.87**
		4	20×10	71.28	68.99	60.23±6.73	**93.39±4.87**
	BA-50	1	20×10	72.06	61.71	72.03±11.62	**94.41±3.23**
		4	20×10	86.38	69.77	59.01±5.73	**90.55±5.14**
Spring	ER-50	20	15×10	72.24	76.05	99.84±0.47	**99.99±0.02**
		60	15×10	71.43	69.17	97.47±2.93	**98.96±1.25**
	BA-50	20	15×10	91.16	84.67	98.17±5.40	**99.88±0.36**
		40	15×10	**92.82**	63.67	67.89±9.89	83.77±9.14
Kuramoto	ER-50	1	30×30	64.69	64.76	82.09±19.14	**94.93±12.94**
		4	30×30	75.34	63.53	95.96±5.01	**99.30±1.67**
	BA-50	1	20×30	55.46	61.87	69.70±18.16	**90.13±12.38**
		4	20×30	51.13	64.62	89.57±11.69	**97.48±2.78**
FJ	ER-50	1	20×10	53.66	83.64	97.67±1.06	**99.82±0.47**
		4	20×10	58.98	59.00	65.25±11.98	**75.48±10.60**
	BA-50	1	20×10	52.32	86.88	91.62±4.67	**92.63±13.46**
		4	20×10	50.90	67.13	67.27±9.59	**73.89±9.84**
CMN	ER-50	1	20×10	87.39	64.35	89.76±2.59	**97.58±3.38**
		4	20×10	93.13	74.08	89.94±1.42	**98.40±1.92**
	BA-50	1	20×10	87.84	71.51	83.35±2.30	**88.83±6.19**
		4	20×10	92.28	75.39	83.05±2.35	**92.97±5.26**
Netsim	–	1	5×200	94.73	74.83	71.57±1.57	**95.09±0.68**
		2	5×100	94.10	50.20	65.90±6.24	**94.70±0.16**

Table 1: Interaction graph inference accuracy measured by AUC and for various dynamical systems and inferring models. The results for NRI and GDP are averaged over 10 independent runs. ER-n or BA-n denotes the name number of nodes (n) of the graph and δt denotes the sampling interval. The sampling interval refers to second-time samplings in pre-generated trajectories, not samplings during solving the ODEs. In the VOLUME column, $a \times b$ corresponds to #trajectories $\times$ #sampled steps. Boldface marks the highest accuracy.

cian. We also use $\mathbf{M}$ to denote a general graph matrix, either the adjacency matrix or the Laplacian.

Graph Dynamical Systems and Relational Inference

Node i at time t is described by a state vector $\boldsymbol{x}_i^t \in \mathbb{R}^{d_s}$. We write $\boldsymbol{x}^t = (\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \cdots, \boldsymbol{x}_n^t)$ for the state of all nodes at time t. We consider both continuous- and discrete-time graph dynamical systems with synchronized updates,

$$\textbf{Continuous} \qquad \textbf{Discrete}$$
$$\dot{\boldsymbol{x}}_i^t = f_i(\boldsymbol{x}_i^t, (\boldsymbol{x}_k^t)_{k\in N_i}) \quad \boldsymbol{x}_i^{t+1} = f_i(\boldsymbol{x}_i^t, (\boldsymbol{x}_k^t)_{k\in N_i}),$$

where N_i is the set of vertex i's neighbours and the dot denotes the time derivative. In relational inference, we observe snapshots of a graph dynamical system $\{\boldsymbol{x}^t, t \in T\}$ at some set T of observation times and aim to find the interaction graph from these snapshots, without any knowledge about the form of f_i. Figure 1 (a) illustrates the common framework for neural network-based RI approach. While the ground truth graph and the dynamics (top row) are both unknown, a graph generator and a dynamics surrogate are trained simultaneously (bottom row). The unknown graph is then predicted to be the best graph that explains the snapshots $\{\boldsymbol{x}^t, t \in T\}$.

Polynomial Graph Filters

A degree-K polynomial graph filter with coefficients $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_K)$ is defined as $g_{\boldsymbol{\theta}}(\mathbf{M}) = \sum_{k=0}^K \theta_k \mathbf{M}^k$. Assuming $\mathbf{M}$ is symmetric,[2] we let $\mathbf{M} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top$ be its eigenvalue decomposition. Then $g_{\boldsymbol{\theta}}(\mathbf{M}) = \mathbf{U} g_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})\mathbf{U}^\top$, where $g_{\boldsymbol{\theta}}(\boldsymbol{\Lambda})$ applies element-wise to the diagonal elements of $\boldsymbol{\Lambda}$ and $g_{\boldsymbol{\theta}}(\lambda) = \sum_{k=0}^K \theta_k \lambda^k$. The scalar polynomial $g_{\boldsymbol{\theta}}(\lambda)$ is called the convolution kernel. By the Weierstrass approximation theorem, any continuous function on a bounded in-

[2]This assumption is made for simplicity of explanation; our proposed method easily handles directed graphs.

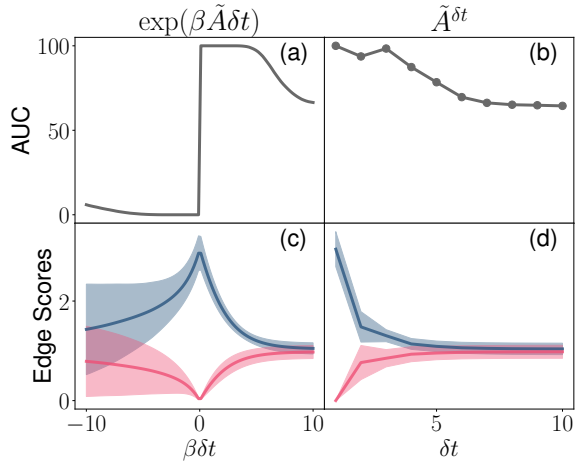

Figure 2: Effects of observation intervals on the effective interaction graph. The AUC for (a) continuous and (b) discrete dynamics. The absolute value of normalized average score for (c) continuous and (d) discrete dynamics. The shaded regions show the standard deviation of each edge class. The results are obtained on an Erdös–Rényi graph with $n = 30$ and $p = 0.3$.

terval can be approximated by a polynomial with arbitrary precision.

Interaction Retrieval

Effective Interaction Graph

Unless the sampling rate is very high, the effective graph modeled by the neural surrogate contains both direct and indirect interactions. Let us illustrate this. Consider a linear dynamics with scalar states. Let $\boldsymbol{x}^t = [x_1^t, \dots, x_n^t]^T$ be the vector of node states at time t and $\dot{\boldsymbol{x}}^t = \beta \mathbf{M} \boldsymbol{x}^t$. Solving the linear ODE, the states at t and $t + \delta t$ are related as $\boldsymbol{x}^{t+\delta t} = \exp(\beta \mathbf{M} \delta t) \boldsymbol{x}^t$, where $\exp(\beta \mathbf{M} \delta t)$ is the matrix exponential which encodes the effective interaction graph. Since

$$\exp(\beta \mathbf{M} \delta t) = \mathbf{I} + \beta \mathbf{M} \delta t + \frac{\beta^2}{2!} \mathbf{M}^2 \delta t^2 + \cdots,$$

the interactions in principle exist instantaneously for all path-reachable nodes. For small δt, the power series can be approximated by truncating at first order in δt and the interaction graph is effectively encoded by $\mathbf{M}$, but for moderate or large δt, we need to include the higher-order terms.

For general nonlinear dynamics, let $\bar{N}_i$ be the set of nodes that $x_i^{t+\delta t}$ *effectively* depends on. Let q_i be the effective transition function, $x_i^{t+\delta t} = q_i((x_k^t)_{k \in \bar{N}_i})$. If q_i is continuous, Kolmogorov–Arnold theorem lets us write (Khesin and Tabachnikov 2014; Zaheer et al. 2017) $x_i^{t+\delta t} = \rho_i(\sum_{j \in \bar{N}_i} J_{i,j} \phi(x_j^t))$, for some continuous ϕ and ρ_i. The function ϕ is independent of q_i, and the parameters $\mathbf{J} = (J_{i,j})$ can be interpreted as interaction strengths. To build a neural surrogate for the effective system, we can approximate ϕ and ρ_i by neural networks and identify the interaction strengths via training; the NRI decoder can be interpreted in this sense. NRI further distinguishes a node and its neighbours as $x_i^{t+\delta t} = x_i^t + \rho(x_i, \sum_{j \in N_i} A_{i,j} \phi(x_i^t, x_j^t))$, where ρ and ϕ are neural networks and $\mathbf{A} = (A_{i,j})$ trainable parameters.

While neural networks may approximate effective dynamics, the inferred matrix $\mathbf{A}$ is (at best) close to the *effective* interaction graph $\mathbf{J}$, rather than the true adjacency. In a linear system the effective graph is generated by a polynomial of the transition matrix which motivates the following *polynomial* neural surrogate:

$$x_i^{t+\delta t} = x_i^t + \rho\left(x_i, \sum_{j \in N_i} g_{\boldsymbol{\theta}}(\mathbf{A})_{ij} \phi(x_i^t, x_j^t)\right).$$

We assume the dynamics to be invariant to the neighbour permutations, and therefore let ρ be node-independent (Zaheer et al. 2017). Further, although the above form of the Kolmogorov–Arnold theorem considers scalar node states, we use it as a heuristic to motivate the functional form of the effective interaction graph in GDP for both scalar and vector states. We will experimentally show that this surrogate benefits RI in both linear and nonlinear cases in Section 5. In the following, we discuss (i) when the effective interaction graph can confuse direct and indirect interactions, and (ii) when and how a polynomial neural surrogate can help to set things straight.

Effect of Observation Intervals

To what extent the effective interaction graph reflects the direct interactions is determined by the intrinsic properties of dynamics and by the sampling interval δt. In this section, we empirically study the effect of the sampling rate in two cases where the effective graph can be determined exactly: continuous-time linear dynamics $\dot{\boldsymbol{x}}^t = \beta \tilde{\mathbf{A}} \boldsymbol{x}^t$ where the effective interaction graph is $\mathbf{J} = \exp(\beta \tilde{\mathbf{A}} \delta t)$, and a discrete-time linear system with synchronized updates $\boldsymbol{x}^{t+1} = \tilde{\mathbf{A}} \boldsymbol{x}^t$ where the effective graph is $\tilde{\mathbf{A}}^{\delta t}$. We compare the true graph defined by $\mathbf{A}$ with the effective graph $\mathbf{J}$ by plotting the AUC as a function of δt in Figure 2.

There is a qualitative distinction between the continuous and discrete systems, but in both performance deteriorates as δt grows. For continuous dynamics, the AUC remains close to 100% for small $\beta \delta t$, meaning that the effective graph is dominated by direct interactions. When $\beta \delta t$ becomes larger, direct–indirect confusion deteriorates performance. In fact, even for small $\beta \delta t$ the average scores for positive and negative classes become closer. This signifies a loss of stability which makes it more likely to misclassify edges.

For discrete dynamics, odd hops result in a larger AUC, which leads to the perhaps counter-intuitive conclusion that larger observation intervals do not always yield worse RI. Intuitively, there is always a length-3 walk between two directly connected nodes i, j as $i \to j \to i \to j$, but we can find a length-2 walk between only when they share a common neighbour. The general trend is still that large observation intervals result in a direct–indirect confusion.

From the examples, we can classify sampled dynamics into (i) those with effective interaction graphs closely mir-

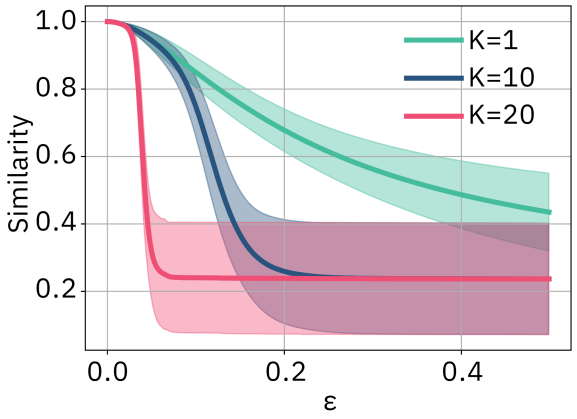

Figure 3: The noise amplifier effect of graph polynomials for RI. The results are obtained on an Erdös–Rényi graph with $n = 50$ and $p = 0.1$. Other parameters are chosen to be $\boldsymbol{\theta} = 1$ and $t = 1\text{e}{-5}$.

roring direct interactions and (ii) those where coarse sampling weakens this correlation. We use the terms "strong" and "weak" correlations informally, without a strict boundary between the two. In Section , we demonstrate that many known dynamical models fit the first category. Finally, we emphasize that unlike our proposed two-scale polynomial architecture, simply stacking multiple message-passing layers does not resolve undersampling and might even worsen performance; we show this in Appendix C.1.

Interaction Graph Retrieval and Noise Amplifier Effect of Graph Polynomials

Even if we can find the *effective* interaction graph (which is the best we can hope for without additional assumptions) and this graph is correlated with the true adjacency, the question is how to recover direct interactions. We discuss this issue in cases where the effective graph is either strongly or weakly correlated with the direct interactions.

First, we consider a weakly correlated example. Consider a linear system in which the effective graph is a polynomial of $\mathbf{M}$ with coefficients $\boldsymbol{\theta}^\star$, $\mathbf{J} = g_{\boldsymbol{\theta}^\star}(\mathbf{M})$. Suppose we find $\mathbf{J}$ and now want to find $\mathbf{M}$; we thus need to solve $g_{\boldsymbol{\theta}'}(\mathbf{M}') \approx \mathbf{J}$ for $\mathbf{M}'$ and $\boldsymbol{\theta}'$ (note that $\mathbf{M}'$ does not necessarily equal $\mathbf{M}$). But the solution of $g_{\boldsymbol{\theta}'}(\mathbf{M}') \approx \mathbf{J}$ is in general not unique even when we know the polynomial coefficients $\boldsymbol{\theta}$. For example, if $\mathbf{J} = \mathbf{M}^2$, the matrix square root equation $(\mathbf{M}')^2 = \mathbf{J}$ is solved by $\mathbf{U}\text{diag}(\mu_1, \cdots, \mu_n)\mathbf{U}^\intercal$, where $\mu_i = \pm\sqrt{\lambda_i}$. In order to identify the correct sign pattern we need to use additional prior knowledge about the graph, such as sparsity. In Lemma A.1 of Appendix A.1, we show that for a general polynomial with known $\boldsymbol{\theta}$ the solution is unique only when the convolution kernel $g_{\boldsymbol{\theta}}(\lambda)$ is injective (for example, the (matrix) exponential) and the graph matrix has no repeated eigenvalues. When $\boldsymbol{\theta}$ is not known, the number of solutions becomes much larger (we always have the trivial solution $\theta'_i = \delta_{i,1}$ and $\mathbf{M}' = \mathbf{J}$.)

In the strongly correlated case, the interaction graph is close to the direct interaction graph in the sense that positive and negative edges can be classified with non-trivial accuracy. While the equation $g_{\boldsymbol{\theta}'}(\mathbf{M}') \approx \mathbf{J}$ may still have many solutions we only want to build a binary classifier which is possible directly from the effective graph. This means that single-step message passing may suffice. However, as we show experimentally in Section , even weak indirect interactions trap this architecture in poor local minima. We now discuss how error amplification in polynomial filters solves the problem.

For $x_i^{t+\delta t} = x_i^t + \rho(x_i, \sum_{j\in N_i} M_{i,j}\phi(x_i^t, x_j^t))$, changing the value of $M_{i,j}$ only affects the next state of node i, $x_i^{t+\delta t}$. It means that errors in the learned interactions only generate gradients locally on the graph. But if we approximate the effective interaction graph by a polynomial, $g_{\boldsymbol{\theta}}(\mathbf{M}) = \sum_{k=0}^{K}\theta_k\mathbf{M}^k \approx \mathbf{J}$, then a perturbation in $M_{i,j}$ propagates through several hops, generating large loss for multiple nodes and thus removing local minima. The issue is that, while the above polynomial equation can always be solved (consider again the trivial assignment $\theta_k = \delta_{k,1}$, $\mathbf{M} = \mathbf{J}$), the solution does not necessarily correspond to the direct interaction graph.

Before we show how to address this issue in Section , let us demonstrate experimentally how including higher-order terms can make the model more sensitive to errors in the graph and produce better gradients. Let $\mathbf{M}_\epsilon = \mathbf{M} + \epsilon\boldsymbol{\Xi}$, where $\boldsymbol{\Xi}$ is a perturbation to the graph matrix $\mathbf{M}$. Consider the graph filter $\mathbf{M}_\epsilon + tg_{\boldsymbol{\theta}}(\mathbf{M}_\epsilon)$, and let $\boldsymbol{y}_{t,\epsilon} = (\mathbf{M}_\epsilon + tg_{\boldsymbol{\theta}}(\mathbf{M}_\epsilon))\boldsymbol{x}$ be the filtered signal; $\boldsymbol{y}_{0,0} = \mathbf{M}\boldsymbol{x}$ corresponds to the unperturbed case. We use cosine similarity $\cos(\boldsymbol{y}_{0,0}, \boldsymbol{y}_{t,\epsilon})$ to quantify the effects of graph perturbations. In Appendix A.2, we show that in the case $\epsilon = 0$, the cosine similarity is bounded from below as

$$\cos(\boldsymbol{y}_{0,0}, \boldsymbol{y}_{t,0}) \geq 1 - t^2|O_N(1)| + O_t(t^3),$$

which indicates that the cosine similarity is close to one with t being sufficiently small. Then, we show numerically that when ϵ deviates from zero, the cosine similarity becomes significantly smaller. We consider uniform random noise $\boldsymbol{\Xi}$ and plot the cosine similarity $\cos(\boldsymbol{y}_{0,0}, \boldsymbol{y}_{t,\epsilon})$ versus ϵ in Figure 3: the similarity decays rapidly when increasing K. The above results show that a polynomial graph filter can amplify the noise in the graph when measured through the filtered signals.

GDP for Relational Inference

Graph Generator We now introduce the full architecture of GDP. We consider the non-amortized setting without an encoder (Löwe et al. 2022; Zhang et al. 2022). We use a simple generator where the probability of an edge (i, j) is

$$A^a_{i,j} = \left(\text{Softmax}(\beta[\Psi^0_{i,j}, \Psi^1_{i,j}])\right)_a \text{ for } a \in \{0, 1\}, \quad (1)$$

$\Psi^0_{i,j}, \Psi^1_{i,j} \in \mathbb{R}$ are trainable latent variables, and β is the inverse temperature. Note that $A^1 = \mathbf{1}\mathbf{1}^\intercal - A^0$. Including both in the model (as below) significantly speeds up convergence.

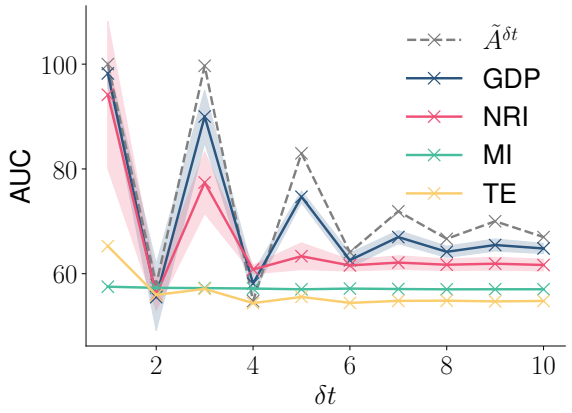

Figure 4: AUC versus sampling rate δt. The gray dashed line denotes the results from explicit interaction graph. For all the experiments, we keep the volume of training data (#trajectories $\times$ #sampled steps) identical. The results are obtained in ER-30 with data volume 30×20.

Neural Dynamics Surrogate We generate two probability matrices, $\mathbf{A}^{(a)}$ for $a \in \{0, 1\}$ and the corresponding (probabilistic) graph filters $\mathbf{F}^{(a)} = g_{\boldsymbol{\theta}}(\tilde{\mathbf{A}}^{(a)})$, where as before $\tilde{\mathbf{A}}^{(a)}$ is the symmetric normalized version of $\mathbf{A}^a$, and the filter coefficients $\boldsymbol{\theta}$ are trainable. We use in-degree (for the direction of message-passing) to normalize the adjacency matrix in the directed graph case. The dynamics surrogate predicts the next state as

$$\begin{aligned} \tilde{\boldsymbol{h}}^t_{(i,j)} &= \sum_{a\in\{0,1\}} F^a_{i,j} \tilde{f}^a_e \left(\boldsymbol{x}^t_i, \boldsymbol{x}^t_j\right), \\ \boldsymbol{x}^{t+1}_j &= \boldsymbol{x}^t_j + \tilde{f}_v \left(\sum_{i\neq j} \tilde{\boldsymbol{h}}^t_{i,j}\right). \end{aligned} \tag{2}$$

In the above architecture, $\tilde{f}^a_e$ are edge-wise MLPs, and $\tilde{f}_v$ is a vertex-wise MLP. As explained in the previous section, a polynomial filter removes the local minima generated by indirect interactions, but it cannot guarantee that the learned $\mathbf{A}$ is close to direct interactions since the roots of the matrix polynomial are not unique. Therefore, we simultaneously train another parallel dynamics neural surrogate using only the adjacency matrix, i.e., replacing $F^a_{i,j}$ by $A^a_{i,j}$ in Equation (2). The loss is simply the sum of MSEs of each neural surrogate. This strategy encourages the solution to stay close to the ground truth interactions while also leveraging the error amplification from graph polynomials. The overall architecture of the proposed model is shown in Figure 1(b). We call the model in Equation 2 using the adjacency matrix as $\mathbf{A}$ model, and the model using a polynomial graph filter as $g_{\boldsymbol{\theta}}$ model. The two models work in parallel with shared $\mathbf{A}$.

Application to Stochastic Dynamics (fMRI)

We further show that with appropriate modifications GDP can be adapted to work with stochastic systems such as functional brain region dynamics in fMRI. Indeed, with an addition of temporal smoothing it performs considerably better than a single-step model and as well as the state-of-the-art method based on mutual information, but unlike that method we learn a model for the dynamics. Using multiple $\mathbf{A}$- or $g_{\boldsymbol{\theta}}$-models and deeper GNNs further improves performance. We leave a more detailed analysis of stochastic dynamics for future work.

Experiments

We consider several representative graph dynamical systems, both continuous and discrete, to validate the proposed algorithm. The continuous-time systems include (i) the Michaelis–Menten kinetics (Karlebach and Shamir 2008), a model for gene regulation circuits; (ii) Rössler oscillators (Rössler 1976) on graphs, which can generate chaotic dynamics; (iii) diffusion, which is a simple a continuous-time linear dynamics; (iv) a network-of-springs model which describes particles interacting via Hooke's law; and (v) the Kuramoto model (Kuramoto 1975) which is a network of phase-coupled oscillators. The discrete-time systems include (vi) Friedkin-Johnsen dynamics (Friedkin and Johnsen 1990), a classical model for describing opinion formation (Abebe et al. 2018; Okawa and Iwata 2022), polarization and filter bubble (Chitra and Musco 2020) in social networks; and (vii) the coupled map network (CMN) (Garcia et al. 2002), a discrete-time model with chaotic behavior. Moreover, we considered a publicly available fMRI dataset (viii) Netsim (Smith et al. 2011), comprising realistic simulated data. A more detailed description of the dynamics and and data generation details can be found in Appendix B.1. The graphs in all datasets but Netsim are undirected. We include further experiments on directed graphs in Appendix C.5. Importantly, we also carry out experiments on real-world data in Appendix C.6 (a dataset of traffic information and a gene regulatory network of S. cerevisiae yeast), and analyze the impacts of graph topology on inference accuracy in Appendix C.7.

Results on Relational Inference

We compare GDP to several baselines. The first one is NRI. The original NRI is designed for the amortized setting where the trajectories do not share the underlying graph. As we primarily consider the classical non-amortized case, we use the version of NRI without a graph encoder (Löwe et al. 2022). We further consider two statistical approaches based on mutual information (MI) and transfer entropy (TE). Implementation details for the baselines can be found in Appendix B.2. The hyperparameters for GDP are summarized in Appendix B.3. As Equation 2 is invariant to swaps of edge types $a \in \{0, 1\}$, it is possible that the learned graph corresponds to the complement graph of direct interactions. This ambiguity is discussed in Appendix B.4.

We apply GDP to the simulated systems and compare it to the baselines. We conduct experiments on Erdős–Rényi (ER) and Barabási–Albert (BA) graphs of different sizes and measure the interaction recovery accuracy by AUC. Table 1 summarizes the average AUC with standard deviations. The

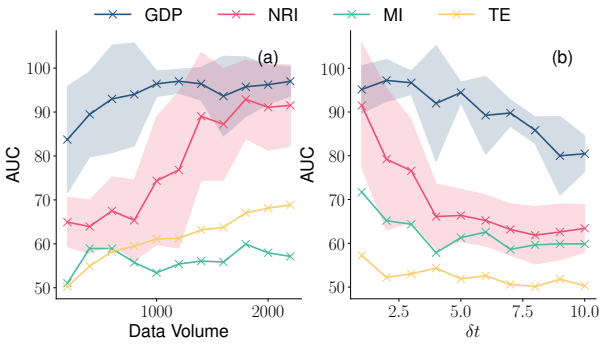

Figure 5: AUC versus (a) data volume and (b) sampling rate δt . The experiments are performed on an ER-20 graph. In (a), the trajectory length and sampling interval are fixed as 10 and 5. We increase the data volume by including more trajectories. In (b), the data volume is fixed as of 50×10.

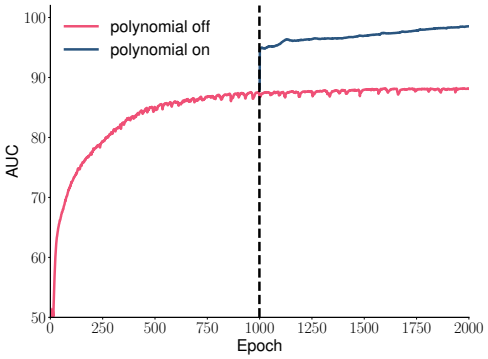

Figure 6: Polynomial graph filters help to escape from local minima. The experiments are performed with Kuramoto model on an ER-50 graph.

first four columns describe the dynamical model, graph type, sampling rate and data volume, respectively. As both NRI and GDP can reach higher accuracy with increasing data volume, the volume of data in Table 1 is determined by the rough criterion that GDP reaches above 90 AUC in the short sampling interval case. The data volumes are kept the same when increasing the sampling interval.

From the experiments, GDP significantly improves the baseline methods. For example, in the Michaelis–Menten model, GDP reaches good accuracy when the other baselines cannot recover helpful information (with an AUC of about 50.00). GDP shows remarkable robustness to undersampling. For example, for the Spring model in BA-50, while NRI and GDP generate accurate predictions with small sampling intervals, GDP degrades much less when increasing the sampling interval. The improvement is not limited to the large sampling rate case, as the error amplifier mechanism of the polynomial filter still works in this case. A phenomenon worth noticing is that the results generally display large fluctuations, which suggests that the loss landscape has many poor local minima. Using a polynomial filter helps escape these poor minima and improves the inference accuracy. In Appendix C.2, we further study the dependence of model performance on the polynomial order K. In Appendix C.3, we perform ablation studies to show that using only the polynomial filter is insufficient to generate stable predictions, as, in general, multiple graph matrices can result in the same polynomial filters. An interesting phenomenon is that a "good" graph for predicting the dynamics turns out to be close to the true graph. We further verify the that the true graph is a local attractor for our model in Appendix C.4.

Robustness to Sampling Rate and Data Volume

We analyze the dependence of GDP's performance on sampling rates and data volume. For discrete-time linear dynamics $\boldsymbol{x}^{t+1} = \tilde{\mathbf{A}}\boldsymbol{x}^t$, there is little correlation between the effective and direct interaction graph $\tilde{\mathbf{A}}$ for even sampling intervals. This is confirmed using RI algorithms, depicted in Figure 4. The gray dashed line displays results from $\tilde{\mathbf{A}}^{\delta t}$. Predictably, even-interval sampling confuses GDP and the three baselines, preventing recovery of direct interactions; imposing constraints such as sparsity seems essential.

The results from $\tilde{\mathbf{A}}^{\delta t}$ provide a rough upper bound for inference accuracy if we only use the effective interactions as the prediction. Both GDP and the other three baselines are below this line, which suggests that even when some positive edges are, in principle, distinguishable, these algorithms can still not find them. The discrepancy between the upper bound and algorithmic results is more pronounced when δt is increased (for odd δt). When $\delta t = 3$, the bound is ≈ 100, close to the $\delta t = 1$ case, but both NRI and GDP are further away in the former case. GDP nonetheless performs the best at all sampling rates.

We consider the Michaelis–Menten model for continuous-time dynamics. We increase δt while keeping the volume of data fixed. The results are in Figure 5 (b). For all algorithms the AUC decreases with δt, suggesting that direct and indirect interactions are more easily confused at larger sampling intervals. Still, GDP shows better robustness to the sampling rates. We next increase the number of training trajectories; Figure 5 (a) shows the results. GDP performs best in all cases. It is the least sensitive to data volume and accurate even with small data sets. This may be essential in real applications where samples are hard or expensive to get.

Polynomial Filters Help Escape Bad Local Minima

We design experiments to provide empirical evidence that polynomial filters can help escape bad local minima. We begin by training a model with only the one-step message passing part; once the AUC reaches a plateau we switch on the polynomial part. Figure 6 plots the evolution of the AUC over training epochs. The blue curve corresponds to when we insist on the one-step-only model: it stays approximately constant after 1000 epochs. After activating the polynomial part, the AUC increases sharply in a single epoch, signaling that we immediately obtained a much more accurate graph. This phenomenon suggests that the polynomial filter indeed produces gradients that help escape the poor local minimum to which a one-step model converged.

Discussion

The experiments show that in a broad range of qualitatively diverse dynamical systems and for a broad range of sampling rates, the non-local neural surrogate in GDP indeed induces a favorable inductive bias and removes poor local minima that cause problems for the earlier local models. As a result, GDP achieves state-of-the-art performance across the board, often by a large margin and at much lower data volumes. We considered a setting where all trajectories share the same graph but our model can be extended to the amortized setting by using a graph encoder which preserves permutation invariance. The proposed model could also be adapted to make causal claims by using only the adjacency part at test time, as was done in (Löwe et al. 2022).

GDP, and other NRI-type methods in general still have several limitations. Firstly, although we have incorporated polynomial filters to address the non-local interactions induced by coarse temporal sampling, GDP works when the effective interaction graph is strongly correlated with the direct interaction graph. This seems to be the case for all combinations of dynamics and sampling rates we tested, but a more challenging task is to look at the strongly mixed case where there is only a weak correlation between the effective and the true graph. Secondly, we currently consider all nodes in the network to be observed; in practice, we only get partial observation whose topology may change with time. Thirdly, GDP may not be suitable in situations where the nodes are highly heterogeneous as in, for example, some metabolic networks. Finally, our claims and understanding of the method are currently based on heuristics and experiment; a precise theory is yet to be worked out.

Acknowledgments

LP would like to acknowledge support from National Natural Science Foundation of China under Grand No. 62006122 and 42230406. CS and ID were supported by the European Research Council (ERC) Starting Grant 852821—SWING.

References

Abebe, R.; Kleinberg, J.; Parkes, D.; and Tsourakakis, C. E. 2018. Opinion dynamics with varying susceptibility to persuasion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1089–1098.

Alet, F.; Weng, E.; Lozano-Pérez, T.; and Kaelbling, L. P. 2019. Neural relational inference with fast modular meta-learning. *Advances in Neural Information Processing Systems*, 32.

Arenas, A.; Díaz-Guilera, A.; Kurths, J.; Moreno, Y.; and Zhou, C. 2008. Synchronization in complex networks. *Physics Reports*, 469(3): 93–153.

Castellano, C.; Fortunato, S.; and Loreto, V. 2009. Statistical physics of social dynamics. *Reviews of Modern Physics*, 81(2): 591.

Chien, E.; Peng, J.; Li, P.; and Milenkovic, O. 2021. Adaptive Universal Generalized PageRank Graph Neural Network. In *International Conference on Learning Representations*.

Chitra, U.; and Musco, C. 2020. Analyzing the impact of filter bubbles on social network polarization. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, 115–123.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 3844–3852.

Friedkin, N. E.; and Johnsen, E. C. 1990. Social influence and opinions. *Journal of Mathematical Sociology*, 15(3-4): 193–206.

Garcia, P.; Parravano, A.; Cosenza, M.; Jiménez, J.; and Marcano, A. 2002. Coupled map networks as communication schemes. *Physical Review E*, 65(4): 045201.

Gasteiger, J.; Bojchevski, A.; and Günnemann, S. 2018. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *International Conference on Learning Representations*.

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

Ha, S.; and Jeong, H. 2023. Learning Heterogeneous Interaction Strengths by Trajectory Prediction with Graph Neural Network. In *International Conference on Learning Representations*.

Izhikevich, E. M. 2007. *Dynamical systems in neuroscience*. MIT press.

Karlebach, G.; and Shamir, R. 2008. Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, 9(10): 770–780.

Khesin, B. A.; and Tabachnikov, S. L. 2014. *ARNOLD: Swimming Against the Tide: Swimming Against the Tide*, volume 86. American Mathematical Society.

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

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

Kuramoto, Y. 1975. Self-entrainment of a population of coupled non-linear oscillators. In *International Symposium on Mathematical Problems in Theoretical Physics: January 23–29, 1975, Kyoto University, Kyoto/Japan*, 420–422. Springer.

Löwe, S.; Madras, D.; Zemel, R.; and Welling, M. 2022. Amortized causal discovery: Learning to infer causal graphs from time-series data. In *Conference on Causal Learning and Reasoning*, 509–525. PMLR.

Ma, Y.; Liu, X.; Shah, N.; and Tang, J. 2022. Is Homophily a Necessity for Graph Neural Networks? In *International Conference on Learning Representations*.

Okawa, M.; and Iwata, T. 2022. Predicting opinion dynamics via sociologically-informed neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1306–1316.

Ortega, A.; Frossard, P.; Kovačević, J.; Moura, J. M.; and Vandergheynst, P. 2018. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5): 808–828.

Pastor-Satorras, R.; Castellano, C.; Van Mieghem, P.; and Vespignani, A. 2015. Epidemic processes in complex networks. *Reviews of Modern Physics*, 87(3): 925.

Peng, J.; Wang, P.; Zhou, N.; and Zhu, J. 2009. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486): 735–746.

Pouget-Abadie, J.; and Horel, T. 2015. Inferring graphs from cascades: A sparse recovery framework. In *International Conference on Machine Learning*, 977–986. PMLR.

Quinn, C. J.; Coleman, T. P.; Kiyavash, N.; and Hatsopoulos, N. G. 2011. Estimating the directed information to infer causal relationships in ensemble neural spike train recordings. *Journal of Computational Neuroscience*, 30(1): 17–44.

Rössler, O. E. 1976. An equation for continuous chaos. *Physics Letters A*, 57(5): 397–398.

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

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

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2018. Deep Image Prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9446–9454.

Wang, A.; and Pang, J. 2022. Iterative structural inference of directed graphs. In *Advances in Neural Information Processing Systems*.

Wang, W.-X.; Lai, Y.-C.; and Grebogi, C. 2016. Data based identification and prediction of nonlinear and complex dynamical systems. *Physics Reports*, 644: 1–76.

Wu, T.; Breuel, T.; Skuhersky, M.; and Kautz, J. 2020. Discovering nonlinear relations with minimum predictive information regularization. arXiv:2001.01885.

Zaheer, M.; Kottur, S.; Ravanbakhsh, S.; Poczos, B.; Salakhutdinov, R. R.; and Smola, A. J. 2017. Deep sets. *Advances in Neural Information Processing Systems*, 30.

Zhang, Y.; Guo, Y.; Zhang, Z.; Chen, M.; Wang, S.; and Zhang, J. 2022. Universal framework for reconstructing complex networks and node dynamics from discrete or continuous dynamics data. *Physical Review E*, 106(3): 034315.

Zhu, J.; Wang, J.; Han, W.; and Xu, D. 2022. Neural relational inference to learn long-range allosteric interactions in proteins from molecular dynamics simulations. *Nature Communications*, 13(1): 1–16.

Zhu, J.; Yan, Y.; Zhao, L.; Heimann, M.; Akoglu, L.; and Koutra, D. 2020. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33: 7793–7804.