Learning Graph Topology With Functional Priors via Bilevel Optimization

Chenyue Zhang, Shangyuan Liu, Hoi-To Wai, and Anthony Man-Cho So

***Abstract*—Learning graph topology of complex networks is challenging due to limited data availability and imprecise data models. Different from prior works that focus on structural priors with explicit control on macroscopic properties such as sparsity, this paper proposes a novel *functional prior* approach for graph topology learning. We adopt a task-oriented formation hypothesis, under which some complex networks may be shaped by functional pressures associated with a particular task, and incorporate this hypothesis as a regularizer for graph learning. Mathematically, we formulate an equilibrium-constrained bilevel-type problem, where the lower level specifies an equilibrium for the associated task on a candidate graph topology and the upper-level problem trades off between data fitting and task performance. We design a two-timescale gradient descent (TTGD) algorithm and show that, under verifiable conditions, it achieves a sublinear convergence rate in the squared projected-gradient residual of the resulting upper-level problem. We provide theoretical insights on the graph topology learned from the functional priors and show that the resulting regularizers subsume a broad class of graph filter regularizers, including polynomial graph regularizers as special cases. We show via extensive experiments on synthetic and real datasets that, when the chosen functional prior is reasonably aligned with the underlying formation mechanism, the proposed formulation can improve graph-support recovery when data are insufficient.**

***Index Terms*—Graph topology learning, functional priors, bilevel optimization.**

I. Introduction

GRAPH-BASED structures are widely used across data science to capture relationships among features and labels. Learning with graph structures plays a key role in many real-world applications such as life science, graph neural networks, and recommendation systems. However, natural graph topologies are not always available. Consequently, the inference of unknown graph topology from nodal observations has become an important focus across fields of machine learning, signal processing, sociology, and biology [1].

Such a problem, also known as graph topology learning, has been a longstanding challenge in data science and signal processing [2], [3], [4]. Earlier works have focused on identifying graph signal models, i.e., *data generation models*, that relate the nodal observations to the graph topology and developing algorithms to efficiently infer the graph topology. For example, the works [5], [6] proposed to learn the graph topology from smooth signals, the works [4], [7], [8], [9] utilized a Gaussian Markov Random Field (GMRF) to learn a sparse graph topology, the works [10], [11] developed physics-inspired models for graph learning, and the work [12] utilized models that are inspired by graph signal processing [13], [14]. As the networks of interest become more complex, recent developments in graph topology learning have switched to the rising issues of limited data availability and imprecise data models. On one hand, the number of nodal observations is often smaller than the total number of nodes. This type of data scarcity can arise from the high measurement costs [15], privacy concerns [16], or difficulty in collecting complete observation data in large, complex systems [17]. As a result, the graph topology learning problem becomes vulnerable to random noise and often struggles to produce accurate inference [18]. On the other hand, the interplay between nodal observations and underlying graph topology is often complicated, which cannot be captured by the simplified data generation models deployed in existing works.

As a remedy, a common practice is to incorporate prior information in the learning process. Existing literature focuses on applying *structural priors* developed from explicit features of real network structures. They are usually driven by empirical observations on common graph topologies. For this class of priors, a canonical hypothesis is edge sparsity. In fact, a common observation across real networks is that the graph topology admits a small number of edges and thus a sparse graph representation [19], [20]. Based on this observation, a series of works proposed to incorporate sparsity regularizers in different graph learning paradigms and achieved empirical successes [4], [5], [6]. Recently, structural priors focusing on the clustering properties of graphs have been studied. Examples include k-component graphs [21], bipartite graphs [22], and connected graphs [23]. As demonstrated in [24], [25], [26], carefully crafted regularizers based on the graph spectra can

Received 22 August 2025; revised 3 May 2026 and 7 August 2026; accepted 20 August 2026. Date of publication 26 August 2026; date of current version 25 September 2026. The associate editor coordinating the review of this article and approving it for publication was Elvin Isufi. *(Corresponding author: Hoi-To Wai.)*

Chenyue Zhang was with the Department of SEEM, The Chinese University of Hong Kong, Shatin, Hong Kong SAR of China. She is now with the Department of Electrical and Computer Engineering, Cornell University, Cornell Tech, New York, NY 10044 USA (e-mail: cyuezhang@link.cuhk.edu.hk).

Shangyuan Liu was with the Department of SEEM, The Chinese University of Hong Kong, Shatin, Hong Kong SAR of China. He is now with Qube Research & Technologies, Central, Hong Kong SAR of China (e-mail: shangyuanliu@link.cuhk.edu.hk).

Hoi-To Wai and Anthony Man-Cho So are with the Department of SEEM, The Chinese University of Hong Kong, Hong Kong SAR of China (e-mail: htwai@se.cuhk.edu.hk; manchoso@se.cuhk.edu.hk).

Digital Object Identifier 10.1109/TSP.2026.3727897

induce desired clustering characteristics in the learned graph. Note that, however, these priors are imposed directly on the graph topology itself in a macroscopic manner and do not account for the more intrinsic relationships between nodes.

This paper initiates the study of a *functional prior* approach to graph topology learning. We aim to rigorously capture the complex dynamics and relationships between nodes in the graph topology learning process. The key idea is a *task-induced formation model* that treats the graph topology as a latent variable optimized to perform a specific task. Here, the task can be complex and nonlinear such as maximizing social welfare in social networks or enhancing resilience in biological networks. We notice that recent works in network science have demonstrated that network topology may play a crucial role in shaping the performance of the networked systems. For social networks, prior works have shown the importance of network topology in network game-induced tasks such as optimal targeting [27] and optimal pricing [28], suggesting that certain network structures may inherently favor specific applications. For biological networks, the work [29] studied the relationship between network topology and resilience to network failure. Particularly, it is found that natural networks such as ecological and gene regulatory networks tend to have a heterogeneous degree distribution, which provides evidence for a resilient network topology. In these examples, the topology of a natural networked system can be modeled as reflecting task-related functional pressures, which summarize aspects of the underlying task performed by the system.

These findings have motivated the current paper to look beyond the direct *structural priors* approach for graph topology learning, and to develop a novel class of *functional priors* for efficient graph topology learning. In a nutshell, the functional prior framework encapsulates the purported complex dynamics of the network into a merit function that regularizes the graph learning objective, leading to a *biased* graph topology estimator. We develop the latter merit function as a shaping function on the equilibria or stationary states of a suitable network dynamics model, which are then treated as intermediate variables that implicitly depend on the latent graph topology. As we will demonstrate later, the advantage of doing so is that the merit function can be defined to impose a wide class of desired properties on the interpretable equilibria or stationary states, e.g., maximizing social welfare in social networks, or maximizing resilience to network failure in gene regulatory networks. Mathematically speaking, the functional prior framework gives rise to a novel bilevel optimization problem for graph topology learning. Here, the lower level is an equilibrium/stationarity equation that governs the network dynamics, while the upper-level problem trades off between data fitting and desired task performance, as shown in Fig. 1.

Computationally, recent literature on bilevel optimization has made significant progress in developing efficient algorithms for tackling such problems. For example, the work [30] proposed a two-timescale stochastic approximation (TTSA) framework that utilizes a pair of step sizes for tackling the coupled KKT system, the work [31] proposed a penalty method, and the work [32] proposed a truncated gradient unrolling based method.

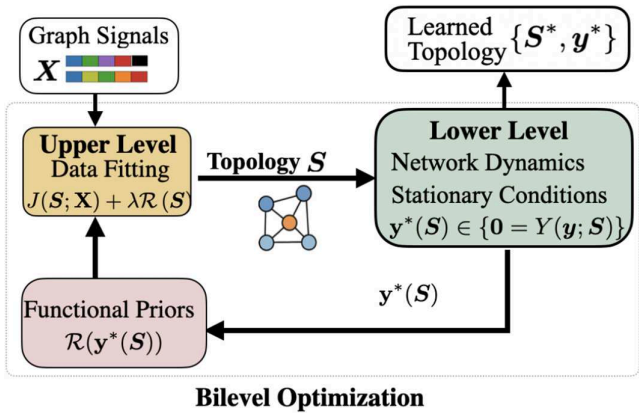


Fig. 1. Overview of the bilevel optimization framework for graph topology learning with functional priors. The upper-level problem minimizes a data fitting term $J(\boldsymbol{S};\mathbf{X})$ augmented with a task-driven regularizer $\mathcal{R}(y^*(\boldsymbol{S}))$, where the lower-level equilibrium equation specifies the stationary state $y^*(\boldsymbol{S})$ of a network dynamics model.

Among others, we develop a two-timescale gradient descent (TTGD) algorithm inspired by TTSA [30] to tackle the bilevel optimization problem. The contributions of this paper are as follows:

- We propose and formulate a framework for graph topology learning with functional priors (GLFP). The key feature is a task-inspired formation model for the unknown graph topology, which induces a functional-prior regularizer. The graph learning problem is then formulated as an equilibrium-constrained bilevel-type problem, where the upper level is a regularized graph learning objective and the lower level specifies the task equilibrium or stationary state.
- As applications of the GLFP framework, we study two specific cases: (i) a social welfare maximization prior for learning social network topology, and (ii) a resilience maximization prior for learning gene regulatory network topology. In both cases, we provide empirical evidence of the functional priors in real networks and design the induced graph topology learning problems with different data observation models that specify the GLFP framework.
- To understand the optimal solutions that GLFP leads to, we illustrate how they can be approximated by "high-order" polynomial regularizers. By analyzing the optimality conditions of the approximations, we provide theoretical insights into the graph topology learned from GLFP. Focusing on the two special cases mentioned, we show that the drawn insights coincide with the topology features observed in real networks. Inspired by the approximations, we further suggest that our proposed functional prior regularizers may generalize over classical structural priors.
- Finally, we design an efficient TTGD algorithm to tackle the GLFP problem. The TTGD algorithm is inspired by the TTSA framework [30] and extends TTSA to tackle variational inequalities in the lower-level problems. To study the convergence behavior of TTGD, we use the squared norm of the projected-gradient residual as the stationarity measure, which is standard in the literature (see, e.g.,

[33]). Under the stated assumptions, we show that TTGD achieves a sublinear convergence rate in this measure for the bilevel problem.

We remark that the idea of optimizing the graph topology for a task-specific application has been explored in several prior studies. For example, the work [34] modified social network topology for polarization reduction, the work [35] modified network topology to enhance performance in action-coordination, and the work [36] focused on learning network structures tailored for node classification. Their primary goal is to improve downstream performance in control or graph machine learning applications. In this paper, we focus on improving the accuracy of learning real networks. More specifically, we study data-limited node-level (microscopic) topology inference, recovering interactions among nodes from limited nodal observations and task-related prior information. Compared to our conference version [37], this paper includes a general GLFP framework, strengthened theoretical results, and extended numerical experiments.

Notation. The notation we use in this paper is standard. We use $[m]$ to denote the set $\{1, 2, \ldots, m\}$ for any positive integer m. Let the Euclidean space of all real vectors/matrices be equipped with the inner product $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle := \mathrm{Tr}(\boldsymbol{X}^\top \boldsymbol{Y})$ for any matrices $\boldsymbol{X}, \boldsymbol{Y}$ and denote the induced Frobenius norm by $\|\cdot\|_F$ (or $\|\cdot\|_2$ when the argument is a vector). For a vector $\boldsymbol{x} \in \mathbb{R}^m$, we use x_i to denote its i-th element and $\boldsymbol{x}_{-i} \in \mathbb{R}^{m-1}$ to denote the vector without x_i. We use $\mathsf{Diag}(\boldsymbol{x})$ to denote the diagonal matrix whose diagonal elements are given by $\boldsymbol{x}$. For any matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$, let $\|\boldsymbol{X}\|_2$ be the operator norm, $\|\boldsymbol{X}\|_1 := \sum_{i,j} |X_{ij}|$, $\|\boldsymbol{X}\|_\infty := \max_i \sum_j |X_{ij}|$, and $\mathrm{diag}(\boldsymbol{X})$ be the vector formed by the diagonal entries of a square matrix $\boldsymbol{X}$. Given a point $\boldsymbol{w}$ and a closed convex set $\mathcal{C}$, we use $\mathsf{Proj}_{\mathcal{C}}(\boldsymbol{w}) := \arg\min_{\boldsymbol{v} \in \mathcal{C}} \|\boldsymbol{v} - \boldsymbol{w}\|_2$ to denote the projection of $\boldsymbol{w}$ onto $\mathcal{C}$. We use 1 (resp. 0) to denote an all-one vector (resp. all-zero vector) whose dimension will be clear from the context. For a set $\mathcal{S}$, we let $\iota_{\mathcal{S}}(\cdot)$ be its indicator function, i.e., $\iota_{\mathcal{S}}(x) = 0$ if $x \in \mathcal{S}$, $\iota_{\mathcal{S}}(x) = \infty$ if $x \notin \mathcal{S}$.

II. Preliminaries

This section reviews the basic concepts and models for graph topology learning. To fix notation, we first consider a networked system characterized by a directed graph $\mathcal{G} = (V, E)$ with node set $V = [N]$ and edge set $E \subseteq V \times V$, such that the ordered tuple $(i, j) \in E$ indicates an edge from i to j. To describe $\mathcal{G}$, we denote its *graph shifting operator* (GSO) by $\boldsymbol{S} \in \mathbb{R}^{N \times N}$, such that $S_{ij} = 0$ if $(j, i) \notin E$. For instance, $\boldsymbol{S}$ can be the (weighted) adjacency matrix or the (weighted) Laplacian matrix, and it is not necessarily symmetric. Notice that in graph signal processing, a related concept is that of *linear graph filter* $H(\cdot) : \mathbb{R}^{N \times N} \to \mathbb{R}^{N \times N}$ with order p [13], which is defined as a p-th order matrix polynomial of a GSO $\boldsymbol{S}$. Mathematically, it takes the form $H(\boldsymbol{S}) = \sum_{i=0}^{p} h_i \boldsymbol{S}^i$. Here, we denote $\boldsymbol{S}^0 = \boldsymbol{I}$ by convention and $\{h_i\}_{i=0}^p$ are the graph filter coefficients.

Graph Topology Learning. The problem pertains to inferring the unknown graph $\mathcal{G}$, or equivalently, its GSO matrix $\boldsymbol{S}$, from a set of *graph signals* defined on the node set V. The graph signals are denoted by $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)}$ such that for each $m \in [M]$, the graph signal $\boldsymbol{x}^{(m)} \in \mathbb{R}^N$ describes the nodes' states. Throughout, we set $\boldsymbol{X} = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(M)})$ to simplify notation. As in statistical learning, the general formulation for graph topology learning can be expressed as

$$\min_{\boldsymbol{S} \in \mathcal{S}} J(\boldsymbol{S}; \boldsymbol{X}) + \lambda \mathcal{R}(\boldsymbol{S}), \tag{1}$$

where $J(\boldsymbol{S}; \boldsymbol{X})$ is the *data fitting* term, $\mathcal{S} \subseteq \mathbb{R}^{N \times N}$ refers to the feasible set of GSOs, $\mathcal{R}(\boldsymbol{S})$ describes the regularizer pertaining to the prior knowledge on the unknown graph, and $\lambda > 0$ is a regularization parameter.

Generation Model. The data fitting term $J(\boldsymbol{S}; \boldsymbol{X})$ in (1) pertains to the *generation model* for graph signal observations. In scenarios where $\boldsymbol{X}$ are low pass or smooth graph signals [38], a popular strategy is to fit an adjacency matrix $\boldsymbol{S}$ such that $\boldsymbol{X}$ is 'smooth' with respect to (w.r.t.) the latter, i.e.,

$$J_{\mathsf{smo}}(\boldsymbol{S}; \boldsymbol{X}) = \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}),\ D_{ij} = \tfrac{1}{2M} \|\boldsymbol{x}_i^{\mathrm{row}} - \boldsymbol{x}_j^{\mathrm{row}}\|_2^2 \tag{2}$$

for all $i, j \in [N]$. Observe that $J_{\mathsf{smo}}(\boldsymbol{S}; \boldsymbol{X})$ can be expressed as $\frac{1}{2M} \sum_{m=1}^{M} \sum_{i,j} S_{ij} |x_i^{(m)} - x_j^{(m)}|^2$ and measures the average Dirichlet energy of the graph signals w.r.t. a proposed $\boldsymbol{S}$.[1] The set of feasible adjacency matrices can be chosen as

$$\mathcal{S}_{\mathsf{ng}} = \{\boldsymbol{S} \in \mathbb{R}^{N \times N} : \boldsymbol{S} \geq 0,\ \mathrm{diag}(\boldsymbol{S}) = 0,\ \boldsymbol{S}\mathbf{1} = c\mathbf{1}\}, \tag{3}$$

where $c > 0$ is enforced to avoid the trivial solution. Along the same line, alternative models include factor analysis [5], stationary graph signals [12], etc. They involve similar forms for the data fitting term $J(\boldsymbol{S}; \boldsymbol{X})$ and the constraint set $\mathcal{S}$.

In scenarios pertaining to network dynamics systems such as gene regulatory networks, the input $\boldsymbol{P} \in \mathbb{R}^{N \times M}$, whose m-th column records the exogenous intervention or external input applied in experiment m, is available alongside graph signals $\boldsymbol{X}$ of the steady-state responses. The following data-fitting loss is proposed in [39], [40]:

$$J_{\mathsf{per}}(\boldsymbol{S}; \boldsymbol{X}) = \|\boldsymbol{S}\boldsymbol{X} + \boldsymbol{P}\|_F^2. \tag{4}$$

This loss encourages each observed steady-state response to satisfy the linearized equilibrium relation $\boldsymbol{S}\boldsymbol{x}^{(m)} \approx -\boldsymbol{p}^{(m)}$. We adopt the set of feasible adjacency matrices

$$\mathcal{S}_{\mathsf{nd}} = \{\boldsymbol{S} \in \mathbb{R}^{N \times N} : \boldsymbol{S} \geq 0, \mathrm{diag}(\boldsymbol{S}) = 0, 1^\top \boldsymbol{S} 1 = a\}, \tag{5}$$

where $a > 0$ is enforced to avoid the trivial solution.

We remark that there are alternative models to (2) and (4), where they depend on the types of available data; see [2], [3].

Formation Model. The regularizer $\mathcal{R}(\boldsymbol{S})$ in (1) pertains to the *prior* information on the graph topology. It is especially important when the number of graph signal observations is insufficient, i.e., $N \gg M$. From a high level, the prior information is related to a *formation model* which characterizes how the graph $\mathcal{G}$ is generated. In the existing literature, common choices entail *explicit* and *macroscopic* control over the graph

[1] The formulation (2) is commonly used to infer a symmetric GSO matrix $\boldsymbol{S}$. We do not impose such a constraint in our discussions to avoid restricting the design of our functional prior; see Sec. III-A.

topology. For example, the sparsity prior is relevant to the multivariate exponential distribution. It is shown in [24] that this prior distribution can induce the regularizer

$$\mathcal{R}_{\ell_1}(\boldsymbol{S}) = \|\boldsymbol{S}\|_1 \tag{6}$$

in the maximum a posteriori (MAP) estimate.[2] Another example is the prior that the graph is formed with a modular structure which has K densely connected components [25], [26]. This formation model is related to the regularizer

$$\mathcal{R}_{\mathsf{mod}}(\boldsymbol{S}) = \sum_{i=K+1}^{N} \sigma_i(\boldsymbol{S}), \tag{7}$$

where $\sigma_i(\cdot)$ represents the i-th largest singular value of $\boldsymbol{S}$.

III. Problem Formulation

This paper proposes a graph learning framework that departs from the structure-inspired designs for the regularizer $\mathcal{R}(\boldsymbol{S})$, e.g., (6), (7). Instead, we develop *task-inspired* regularizers that *implicitly* regulate the graph topology through the latter's influence on tasks performed by the network. At a high level, our framework is developed through a new *formation model* of the graph topology. Motivated by the possibility that functional pressures may shape real-world networks, we adopt the following working hypothesis:

The topologies of some real-world networks may be shaped by task-specific functional pressures represented through a merit function.

For example, a possible formation model for socio-economic networks is that the graph topology leads to the highest payoff at the Nash Equilibrium (NE) in a network game [41]; for gene regulatory networks, the graph topology leads to resilient states in the stationary states of a bio-physical system [42], [43].

Overall, consider a general merit function design that is defined via the stationarity condition for network states

$$\boldsymbol{y}^\star(\boldsymbol{S}) \in \mathcal{Y}^\star(\boldsymbol{S}) := \{\boldsymbol{y} \in \mathcal{Y} : \mathsf{Y}(\boldsymbol{y}; \boldsymbol{S}) = 0\},$$

where $\mathcal{Y} \subseteq \mathbb{R}^N$ is the set of feasible states such that $\mathsf{Y}(\cdot; \boldsymbol{S}) \in \mathbb{R}^N$ is a stationary condition map. For example, in network games, it is related to the fixed point of best response dynamics. Subsequently, the merit function is given by the implicit function

$$\mathcal{R}(\boldsymbol{S}) := \min_{\boldsymbol{y} \in \mathcal{Y}^\star(\boldsymbol{S})} \hat{\mathcal{R}}(\boldsymbol{y}) + (\tilde{\beta}/2)\|\boldsymbol{S}\|_F^2, \tag{8}$$

such that $\hat{\mathcal{R}} : \mathcal{Y} \to \mathbb{R}$ quantifies the merit of the network stationary states and the Frobenius norm term is added to stabilize the optimization solution with $\tilde{\beta} > 0$. For example, $\hat{\mathcal{R}}(\boldsymbol{y})$ is the negated sum of payoffs in a network game. Together, this leads to the *bilevel program*

$$\begin{aligned} \min_{\boldsymbol{S} \in \mathcal{S}, \boldsymbol{y} \in \mathcal{Y}} \quad & \Phi(\boldsymbol{S}; \boldsymbol{y}) := J(\boldsymbol{S}; \boldsymbol{X}) + \lambda \hat{\mathcal{R}}(\boldsymbol{y}) + (\lambda \tilde{\beta}/2)\|\boldsymbol{S}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{y} \in \mathcal{Y}^\star(\boldsymbol{S}), \end{aligned} \tag{GLFP}$$

which combines the data fitting term from the *generation model* and the regularizer term from the *formation model*.

We use the term "bilevel" in the broad implicit-response sense. The lower level in (GLFP) is the equilibrium equation $\mathsf{Y}(\boldsymbol{y}; \boldsymbol{S}) = 0$. It represents a lower-level optimization problem only when Y is the optimality map of such a problem.

We shall describe regularizers given in the form of (8) as *functional regularizers*, which depart from the structural regularizers considered in the prior works. We next discuss two special cases of the formation model in (GLFP).

A. Case Study 1: Network Games

We consider a formation model driven by benchmarking the total welfare resulting from the Nash Equilibrium (NE) of a network game [28]. The game setting is related to learning socio-economic networks, where agents are assumed to be rational with actions shaped by a certain utility function. The lower-level state $\boldsymbol{y}$ collects the agents' equilibrium actions induced by a candidate graph topology, so the functional prior favors graphs with high aggregate welfare at equilibrium.

In this setting, each node represents an agent and the edge weights represent the strength of trust between pairs of neighboring agents. Each agent is endowed with a payoff function $U_i(\cdot)$ that depends on the actions of his/her neighbors and himself/herself. For each agent i, s/he finds an action $y_i^\star$

$$y_i^\star = \arg\max_{y_i \in \mathcal{Y}_i} U_i(y_i, \boldsymbol{y}_{-i}; \boldsymbol{S}), \tag{9}$$

where $\boldsymbol{y}_{-i}$ denotes the vector $\boldsymbol{y} = (y_1, \ldots, y_N)$ with the i-th agent's action removed and $\mathcal{Y}_i \subseteq \mathbb{R}$ is the set of admissible actions. The agents are non-cooperative and aim to maximize their own payoffs (9). To this end, the NE describes the set of actions where no agent shall change his/her action. Mathematically, assuming that the NE exists and is unique, it is defined through the best response map

$$\mathsf{T}_i(\boldsymbol{y}; \boldsymbol{S}) := \arg\max_{y_i \in \mathcal{Y}_i} U_i(y_i, \boldsymbol{y}_{-i}; \boldsymbol{S}), \ \forall\, i \in V.$$

Collecting the component maps gives the vector operator

$$\mathsf{T}(\boldsymbol{y}; \boldsymbol{S}) := \big(\mathsf{T}_1(\boldsymbol{y}; \boldsymbol{S}), \ldots, \mathsf{T}_N(\boldsymbol{y}; \boldsymbol{S})\big)^\top.$$

In particular, $\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S}) = (y_1^{\mathsf{NE}}(\boldsymbol{S}), \ldots, y_N^{\mathsf{NE}}(\boldsymbol{S}))$ is said to be an NE if

$$y_i^{\mathsf{NE}}(\boldsymbol{S}) = \mathsf{T}_i(\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S}); \boldsymbol{S}), \ \forall\, i \in V.$$

To simplify notation, we denote $\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})$ as the fixed point to the equation $\boldsymbol{y} = \mathsf{T}(\boldsymbol{y}; \boldsymbol{S})$. This yields a special case of (GLFP) with

$$\mathsf{Y}(\boldsymbol{y}; \boldsymbol{S}) := \boldsymbol{y} - \mathsf{T}(\boldsymbol{y}; \boldsymbol{S}).$$

We consider two models for the payoff function. The first model is a *generalized linear-quadratic* (LQ) game [28], [44], [45], [46] whose payoff is given by

$$U_i^{lq}(y_i, \boldsymbol{y}_{-i}; \boldsymbol{S}) = -\tfrac{y_i^2}{2} + y_i \left(\textstyle\sum_{j=1}^N S_{ij} f(y_j) + b_i\right) \tag{10}$$

[2] It is shown in [6] that one can equivalently use the squared-Frobenius regularizer $\mathcal{R}(\boldsymbol{S}) = \|\boldsymbol{S}\|_F^2$.

over $\mathcal{Y}_i = [0, \infty)$, where b_i is agent i's node-specific intrinsic incentive (or marginal benefit) that is independent of peer effects, and $f(\cdot)$ is an interaction function that modulates the effects of neighbors' action on agent i. The generalized LQ game is a canonical model for strategic interaction on a network [47]. In this model, each agent i chooses a nonnegative action level y_i, which can be interpreted as effort or participation intensity. The first term $-y_i^2/2$ represents a private cost of exerting action, while the term $\sum_{j=1}^N S_{ij} f(y_j)$ captures how agent i's incentive is shaped by the actions of its neighbors on the graph. The interaction function $f(\cdot)$ acts on individual neighbors' actions before aggregation, so the generalized LQ game captures individually mediated spillovers. See [44] for conditions on the existence and uniqueness of NE for the above model.

The second model is a *race and tournament* (RT) game [48], [49], [50] whose payoff is

$$U_i^{rt}(y_i, \boldsymbol{y}_{-i}; \boldsymbol{S}) = -\tfrac{y_i^2}{2} + y_i \left(g\left(\textstyle\sum_{j=1}^N S_{ij} y_j \right) + b_i \right) \quad (11)$$

over $\mathcal{Y}_i = [\underline{b}_i, a_i]$, where $\underline{b}_i$ and a_i (with $0 < \underline{b}_i < a_i$) are the lower and upper action bounds, respectively, and $g(\cdot)$ is a nonlinear function that captures the interactions between agent i and his/her neighbors. The RT game models competitive interactions with bounded actions $y_i \in [\underline{b}_i, a_i]$. Note that the nonlinear function $g(\cdot)$ acts on the aggregated neighbor activity $\sum_{j=1}^N S_{ij} y_j$, so the return to effort depends nonlinearly on the overall local competitive environment. Thus, the RT game has a different interaction structure from the generalized LQ game. The latter captures individually mediated spillovers, whereas the former is suited for settings where incentives depend on the aggregate level of peer activity. See [49] for conditions on the existence and uniqueness of NE for the above model.

In general, the graph topology $\boldsymbol{S}$ directly affects the NE $\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})$ [28], [44]. Herein, it is possible to use the *total welfare* [27] to measure the performance of a socio-economic system, defined as

$$\mathsf{Wel}(\boldsymbol{S}) := \mathbf{1}^\top \boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S}).$$

It can be viewed as the overall economic gain. Notice that $\mathsf{Wel}(\boldsymbol{S})$ depends on $\boldsymbol{S}$ via the NE of the network game. We formally state the formation-model hypothesis —

the graph topology of socio-economic and human-made systems shall maximize the total welfare $\mathsf{Wel}(\boldsymbol{S})$

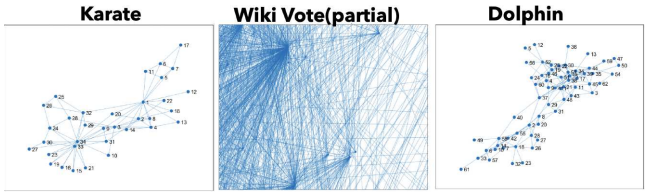
and thus the *merit function* with $\mathcal{R}(\boldsymbol{S}) = -\mathsf{Wel}(\boldsymbol{S}) + \frac{\tilde{\beta}}{2}\|\boldsymbol{S}\|_{\mathsf{F}}^2$.

Directly verifying the above hypothesis is impossible as the ground truth formation model is unknown. We consider an empirical approach through testing on real-world networks. It relies on the insight that if the hypothesis holds, then the corresponding $\boldsymbol{S}$ would be a *local maximum* to $\mathsf{Wel}(\boldsymbol{S})$ such that any perturbation to $\boldsymbol{S}$ could lead to a significant drop in $\mathsf{Wel}(\boldsymbol{S})$. We verify the above phenomenon by applying random rewiring to the graph topology of real-world networks and comparing the reduction in $\mathsf{Wel}(\boldsymbol{S})$. We fix $\boldsymbol{b} = 1$ and consider the payoff function in (10) with linear interaction function $f(x) = x$. We compute the perturbed welfare ratio

$$\mathbb{E}\left[(\mathsf{Wel}(\boldsymbol{S}_{\mathsf{pt}}) - \mathbf{1}^\top \boldsymbol{b}) / (\mathsf{Wel}(\boldsymbol{S}_{\mathsf{og}}) - \mathbf{1}^\top \boldsymbol{b}) \right], \quad (12)$$

where $\boldsymbol{S}_{\mathsf{og}}$ and $\boldsymbol{S}_{\mathsf{pt}}$ are respectively the unweighted adjacency matrices of the original and randomly rewired graphs. Table I shows the welfare ratio on the real-world networks. Observe that human-made networks (e.g., `Karate`) suffer a significant drop in the welfare ratio after rewiring, while other networks (e.g., `Dolphins`) are less sensitive to rewiring. This observation supports our hypothesis. We remark that related observations can also be found in [27], [51], [52] on how real networks show traits of self-optimization.

TABLE I
TOP: VISUALIZING THE TOPOLOGY OF SEVERAL TESTED NETWORKS. BOTTOM: IMPACT OF RANDOM REWIRING ON THE WELFARE RATIO AFTER REWIRING; CF. (12)



Rewiring	10%	30%	50%
`Karate`	**94.06%**	**84.72%**	**78.30%**
`WikiVote`	**96.28%**	**90.32%**	**86.11%**
`LesMiserables`	**93.79%**	**84.43%**	**78.56%**
`JazzMusician`	**97.43%**	**93.47%**	**90.77%**
`SchoolNet`	**97.28%**	**93.47%**	**91.20%**
`MalawiVillage`	**96.72%**	**91.82%**	**88.61%**
`Dolphins`	98.15%	95.13%	93.08%
`Ant`	98.23%	95.90%	94.61%
`Weaver`	99.43%	98.49%	97.80%

Finally, we consider the smoothness based generation model (2) with the above formation model to yield

$$\min_{\boldsymbol{S}, \boldsymbol{y} \in \mathcal{Y}} \ \Phi(\boldsymbol{S}; \boldsymbol{y}) := \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}) + \beta \|\boldsymbol{S}\|_{\mathrm{F}}^2 - \lambda \mathbf{1}^\top \boldsymbol{y}$$
$$\text{s.t. } \boldsymbol{y} - \mathsf{T}(\boldsymbol{y}; \boldsymbol{S}) = \mathbf{0}, \ \boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}, \quad \text{(GL-NG)}$$

where $\mathcal{Y} = \{\boldsymbol{y} \in \mathbb{R}^N : y_i \in \mathcal{Y}_i, \forall\, i \in [N]\}$, $\mathcal{S}_{\mathsf{ng}}$ was defined in (3), and $\beta = \frac{\lambda \tilde{\beta}}{2} > 0, \lambda > 0$ are regularization parameters.

B. Case Study 2: Network Dynamics

We consider a formation model driven by the resilience property of gene regulatory network (GRN) dynamics. The GRN dynamics model the time varying states, quantified by their expression levels, of genes in a cell. This case study targets GRNs, where the candidate graph topology determines how genes influence one another through a nonlinear stationary dynamics model. The lower-level variable therefore represents a steady-state gene-expression profile, and the functional prior rewards graph topologies that sustain large nonzero stationary responses under perturbations.

In a GRN, each node represents a gene and the edge weights represent the influence strengths between genes (possibly directed). The state of each gene is represented by its expression level, which can be affected by the expression levels of its neighboring genes. To capture the changes in the gene's expression level, a widely used model is the Michaelis-Menten (MM) system of differential equations [53], [54]

$$\frac{\mathrm{d}}{\mathrm{d}\,t} y_i(t) = -y_i(t) + \sum_{j=1}^{N} S_{ij} \frac{y_j(t)^2}{y_j(t)^2+1}, \ \forall\, i \in V, \tag{13}$$

where $y_i(t)$ denotes the i-th gene's expression level at time t. The linear decay term $-y_i(t)$ models the natural degradation of gene products, while the nonlinear regulatory input from neighboring genes takes the form of a saturating response function. In particular, the term $y_j(t)^2/(y_j(t)^2+1)$ possesses a Hill-function-type nonlinearity: When the expression level of gene j is low (resp., high), its regulatory effect is weak (resp., saturates). Such saturation is a standard feature in biochemical regulation models and is one reason why MM systems are widely used in systems biology [53], [54]. Under (13), a stationary state of the cell $\boldsymbol{y}^\star$ defines a collection of gene states that do not change, i.e.,

$$\mathcal{Y}^\star(\boldsymbol{S}) := \left\{ \boldsymbol{y} \in \mathbb{R}^N : y_i = \sum_{j=1}^{N} S_{ij} \frac{y_j^2}{y_j^2+1}, \ \forall\, i \in V \right\}. \tag{14}$$

Note that in general $|\mathcal{Y}^\star(\boldsymbol{S})| > 1$. For example, it must hold that $0 \in \mathcal{Y}^\star(\boldsymbol{S})$, while there may be other non-zero solutions satisfying (14).

In the context of GRN dynamics, the resilience property refers to the ability of a GRN to maintain non-zero stationary states after the system is perturbed due to, e.g., gene knockout or suppression of regulatory interactions. We are interested in the notion of *universal resilience* that is governed by the GRN dynamics equations and graph topology, independent of the specific types of perturbations. To this end, the work [29] proposed a metric based on the heterogeneity of graph topology and showed that the latter is correlated with the resilience of GRN against edge deletion perturbations.

Departing from [29], we propose an alternative resilience metric by directly measuring the number of non-zero genes in the stationary state. In particular, we consider

$$\mathsf{Rob}(\boldsymbol{S}) = \max_{\boldsymbol{y} \in \mathcal{Y}^\star(\boldsymbol{S})} \sum_{i=1}^{N} \frac{1-e^{-\sigma y_i}}{1+e^{-\sigma y_i}}, \tag{15}$$

which approximates the number of non-zero genes in the stationary states resulting from $\boldsymbol{S}$. Here, $\sigma > 0$ is to control the approximation error and a larger σ leads to a closer approximation. On one hand, we have $\mathsf{Rob}(\boldsymbol{S}) = 0$ if and only if the GRN only has null stationary states. On the other hand, a GRN with higher $\mathsf{Rob}(\boldsymbol{S})$ has more genes that are active, which intuitively implies a more resilient GRN. Eq. (15) motivates the following formation-model hypothesis for GRN topology $\boldsymbol{S}$ —

some biological networks may be shaped by pressures favoring the resilience property approximated by $\mathsf{Rob}(\boldsymbol{S})$.

Accordingly, we use the *merit function* $\mathcal{R}(\boldsymbol{S}) = -\mathsf{Rob}(\boldsymbol{S}) + (\tilde{\beta}/2)\|\boldsymbol{S}\|_{\mathsf{F}}^2$. Note that

$$\frac{1-e^{-\sigma y_i}}{1+e^{-\sigma y_i}} = 2\frac{1}{1+e^{-\sigma y_i}} - 1.$$

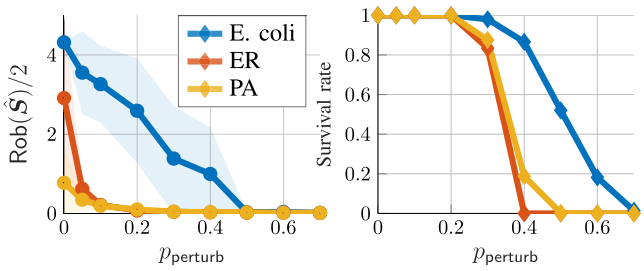


Fig. 2. Resilience in gene regulatory networks in E. coli [29]. The x-axis p_{perturb} is the proportion of link loss. Left: $\mathsf{Rob}(\hat{\boldsymbol{S}})/2$ averaged over perturbed graphs. Right: The survival rate, i.e., the fraction of perturbed graphs that retain non-null stationary states.

Hence, the sigmoid-based objective used below is equivalent to (15) up to an additive constant and a rescaling of the regularization parameter.

Direct verification of this hypothesis is difficult because the true formation mechanism is unknown. We adopt an empirical approach by examining $\mathsf{Rob}(\boldsymbol{S})$ when $\boldsymbol{S}$ is taken as the *E. coli* GRN [29], which has $N = 1454$ genes and $|E| = 3170$ edges,[3] and comparing its corresponding values to control graphs generated as Erdős-Rényi (ER) or Preferential Attachment (PA) graphs with the same number of genes and edges. The left panel of Fig. 2 compares average values of $\mathsf{Rob}(\hat{\boldsymbol{S}})$ against p_{perturb}, where $\hat{\boldsymbol{S}}$ is obtained by randomly deleting a fraction of p_{perturb} edges from $\boldsymbol{S}$. Observe that *E. coli* has a consistently higher $\mathsf{Rob}(\hat{\boldsymbol{S}})$ than the control graphs with similar edge densities. This observation is consistent with the modeling hypothesis. In the right panel of Fig. 2, we compare the survival rates of the GRNs (i.e., the probability of attaining non-null stationary states) under edge deletion perturbations. A similar pattern is observed in the survival-rate results, showing that *E. coli* admits a more resilient topology than the random controls.

Finally, we combine the formation model with a generation model inspired by the linear dynamic approximation in GRN inference [55], [56]. Utilizing observations of the GRN stationary states after knockout experiments as $\boldsymbol{X} \approx -\boldsymbol{S}^{-1}\boldsymbol{P}$, we consider the following graph topology learning problem:

$$\min_{\boldsymbol{S},\boldsymbol{y}} \ \Phi(\boldsymbol{S};\boldsymbol{y}) := \|\boldsymbol{S}\boldsymbol{X} + \boldsymbol{P}\|_F^2 + \beta\|\boldsymbol{S}\|_F^2 - \lambda \sum_{i=1}^{N} \frac{1}{1+e^{-\sigma y_i}}$$
$$\text{s.t. } y_i = \sum_{j=1}^{N} \frac{S_{ij} y_j^2}{y_j^2+1}, \ \forall\, i \in V, \ \boldsymbol{S} \in \mathcal{S}_{\mathsf{nd}}. \tag{GL-GENE}$$

Here, $\beta = \frac{\lambda\tilde{\beta}}{2} > 0, \lambda > 0$ are regularization parameters and $\mathcal{S}_{\mathsf{nd}}$ was defined in (5) with the normalization constant $a > 0$. We note that, unlike the network-game case, the lower-level stationary system in (GL-GENE) may admit multiple equilibria. This distinction will be important when we discuss the convergence of our proposed algorithm in Sec. IV.

[3] We took the largest connected component in the *E. coli* graph.

IV. TTGD Algorithm

This section develops a two-timescale gradient descent (TTGD) algorithm to tackle the graph learning problem with functional priors. Recall that (GLFP) can be written as

$$\min_{\boldsymbol{S}\in\mathcal{S},\boldsymbol{y}\in\mathcal{Y}} \Phi(\boldsymbol{S};\boldsymbol{y}) \text{ s.t. } \mathsf{Y}(\boldsymbol{y};\boldsymbol{S})=0. \tag{16}$$

Note that $\mathsf{Y}(\boldsymbol{y};\boldsymbol{S})=0$ consists of N nonlinear equality constraints. For the purpose of illustration, we assume that (i) *the set of constraints admits a unique solution* $\boldsymbol{y}^\star(\boldsymbol{S})$ *for any* $\boldsymbol{S}\in\mathcal{S}$, *i.e.,* $\mathcal{Y}^\star(\boldsymbol{S})=\{\boldsymbol{y}^\star(\boldsymbol{S})\}$, and (ii) $\boldsymbol{y}^\star(\boldsymbol{S})$ *is Lipschitz continuous w.r.t.* $\boldsymbol{S}$. These are common assumptions in the bilevel optimization literature [30], [57]. Under these assumptions, Problem (16) can be reformulated as

$$\min_{\boldsymbol{S}\in\mathcal{S}} \ell(\boldsymbol{S}) := \Phi(\boldsymbol{S};\boldsymbol{y}^\star(\boldsymbol{S})).$$

The above problem can be handled via the standard projected gradient descent (PGD) algorithm: At iteration $k\geq 0$,

$$\boldsymbol{S}^{k+1}=\mathsf{Proj}_{\mathcal{S}}(\boldsymbol{S}^k-\gamma\nabla\ell(\boldsymbol{S}^k)),\ \forall\, k\geq 0, \tag{17}$$

where $\gamma>0$ is the step size, $\mathsf{Proj}_{\mathcal{S}}(\cdot)$ is the Euclidean projection onto $\mathcal{S}$. The challenge of (17), however, lies in the gradient computation $\nabla\ell(\boldsymbol{S}^k)$ since $\ell(\boldsymbol{S})$ has an implicit dependence on $\boldsymbol{S}$ via $\boldsymbol{y}^*(\cdot)$. To proceed, let us define the map

$$\begin{aligned}\widehat{\nabla}\Phi(\boldsymbol{S};\boldsymbol{y}) &:= \nabla_{\boldsymbol{S}}\Phi(\boldsymbol{S};\boldsymbol{y})\\ &\quad-(\mathrm{J}_{\boldsymbol{S}}\mathsf{Y}(\boldsymbol{y};\boldsymbol{S}))^\top(\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}(\boldsymbol{y};\boldsymbol{S}))^{-\top}\nabla_{\boldsymbol{y}}\Phi(\boldsymbol{S};\boldsymbol{y}),\end{aligned} \tag{18}$$

where $\mathrm{J}_{\boldsymbol{S}}\mathsf{Y}(\cdot),\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}(\cdot)$ denote the Jacobian of the operator Y w.r.t. $\boldsymbol{S},\boldsymbol{y}$, respectively; and $\nabla_{\boldsymbol{S}}\Phi(\cdot),\nabla_{\boldsymbol{y}}\Phi(\cdot)$ denote the partial gradient taken w.r.t. $\boldsymbol{S},\boldsymbol{y}$, respectively. When $\mathsf{Y}(\cdot)$ is smooth, it is shown in [57] that with $\bar{\boldsymbol{y}}^k:=\boldsymbol{y}^\star(\boldsymbol{S}^k)$, the gradient of $\ell(\boldsymbol{S}^k)$ can be computed as

$$\nabla\ell(\boldsymbol{S}^k)=\widehat{\nabla}\Phi(\boldsymbol{S}^k;\bar{\boldsymbol{y}}^k). \tag{19}$$

In other words, evaluating $\nabla\ell(\boldsymbol{S}^k)$ requires the unique solution to $\mathsf{Y}(\boldsymbol{y};\boldsymbol{S}^k)=0$ in $\boldsymbol{y}$.

In practice, the unique solution $\boldsymbol{y}^\star(\boldsymbol{S}^k)$ may not be available, so the TTGD update uses $\widehat{\nabla}\Phi(\boldsymbol{S}^k;\boldsymbol{y}^{k+1})$ as an approximation of $\nabla\ell(\boldsymbol{S}^k)$. The approximation is exact only when $\boldsymbol{y}^{k+1}=\boldsymbol{y}^\star(\boldsymbol{S}^k)$.

To generate $\boldsymbol{y}^k$ that tracks $\boldsymbol{y}^\star(\boldsymbol{S}^k)$, we observe that

$$\mathsf{Y}(\boldsymbol{y};\boldsymbol{S})=0\Leftrightarrow\mathsf{F}(\boldsymbol{y};\boldsymbol{S})=\boldsymbol{y},\quad \mathsf{F}(\boldsymbol{y};\boldsymbol{S}):=\boldsymbol{y}-\mathsf{Y}(\boldsymbol{y};\boldsymbol{S}).$$

The lower-level TTGD update applies the relaxed map

$$\mathsf{F}_\alpha(\boldsymbol{y};\boldsymbol{S}):=(1-\alpha)\boldsymbol{y}+\alpha\mathsf{F}(\boldsymbol{y};\boldsymbol{S})=\boldsymbol{y}-\alpha\mathsf{Y}(\boldsymbol{y};\boldsymbol{S}).$$

With a small stepsize α, it can be shown that F_α is a contractive self-map under mild conditions; see Sec. IV-A. The above observations suggest a two-timescale algorithm mimicking (17) that simultaneously updates $\boldsymbol{y}$ and $\boldsymbol{S}$ at different paces — a larger step size for $\boldsymbol{y}$ and a smaller step size for $\boldsymbol{S}$. In this way, $\boldsymbol{S}^k$ will appear to be 'static' w.r.t. $\boldsymbol{y}^k$. Hence, the fast convergence of the relaxed fixed point iteration of $\boldsymbol{y}^k$ allows it to track the true solution $\boldsymbol{y}^\star(\boldsymbol{S}^k)$. Accordingly, $\widehat{\nabla}\Phi(\boldsymbol{S}^k;\boldsymbol{y}^{k+1})$ approaches $\nabla\ell(\boldsymbol{S}^k)$.

Inspired by [30] and the above, we formalize our algorithm as follows: Let $\gamma>0$, $\alpha\in(0,1]$ be the step sizes and consider

Two Timescale Gradient Descent (TTGD): For $k\geq 0$,

$$\boldsymbol{y}^{k+1}=\boldsymbol{y}^k+\alpha(\mathsf{F}(\boldsymbol{y}^k;\boldsymbol{S}^k)-\boldsymbol{y}^k), \tag{20a}$$

$$\boldsymbol{S}^{k+1}=\mathsf{Proj}_{\mathcal{S}}(\boldsymbol{S}^k-\gamma\widehat{\nabla}\Phi(\boldsymbol{S}^k;\boldsymbol{y}^{k+1})). \tag{20b}$$

In the update of $\boldsymbol{y}^k$ in (20a), we have resorted to a relaxed version of the fixed point iteration with $\alpha\in(0,1]$. Our analysis will show that such a relaxation is necessary to guarantee convergence in several cases. Since $\boldsymbol{y}^{k+1}$ is formed by a convex combination between $\boldsymbol{y}^k$ and $\mathsf{F}(\boldsymbol{y}^k;\boldsymbol{S}^k)$, by induction $\boldsymbol{y}^{k+1}\in\mathcal{Y}$ for any k as long as $\boldsymbol{y}^0\in\mathcal{Y}$.

Although TTGD is developed under the assumption that lower-level equations have a unique solution, the TTGD iterates remain well defined and can still be computed when the uniqueness assumption is violated. In the latter case, $\boldsymbol{y}^\star(\boldsymbol{S})$ may not be a singleton and $\nabla\ell(\cdot)$ is not well defined. Thus, $\widehat{\nabla}\Phi(\boldsymbol{S}^k;\boldsymbol{y}^{k+1})$ should be interpreted only as a computable search direction. Nevertheless, the updates themselves remain well-defined as long as $\mathsf{Y}(\cdot;\boldsymbol{S})$ is differentiable and $\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}$ is invertible.

Remark 1: The inverse in (18) need not be formed explicitly. Instead, the implicit-gradient term can be evaluated by solving the adjoint linear system

$$\left(\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}\right)^\top\boldsymbol{v}=\nabla_{\boldsymbol{y}}\Phi.$$

For directed graphs, the Jacobian $\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}$ may not be symmetric. Hence, standard conjugate gradient is not directly applicable in general. Instead, one can use truncated Neumann approximations or asymmetric Krylov methods. The key computational bottleneck is solving the dense $N\times N$ adjoint linear system. A direct factorization costs $\mathcal{O}(N^3)$ FLOPs per iteration in the dense setting. Alternatively, one may approximate $\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}$ by its diagonal part $\boldsymbol{D}$ [58]. Writing $\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}=\boldsymbol{D}-\boldsymbol{E}=\boldsymbol{D}(I-\boldsymbol{D}^{-1}\boldsymbol{E})$, if $\|\boldsymbol{D}^{-1}\boldsymbol{E}\|<1$, then

$$\left(\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}\right)^{-1}=(I-\boldsymbol{D}^{-1}\boldsymbol{E})^{-1}\boldsymbol{D}^{-1}=\sum_{q=0}^{\infty}(\boldsymbol{D}^{-1}\boldsymbol{E})^q\boldsymbol{D}^{-1}.$$

The diagonal approximation retains the leading-order term $\boldsymbol{D}^{-1}$. Note that in a network, the off-diagonal entries of $\mathrm{J}_{\boldsymbol{y}}\mathsf{Y}$ encode inter-node interactions in the equilibrium system. Discarding them may lead to inaccurate gradient estimates when network coupling effects are strong. Our convergence result below (Theorem 1) focuses on the case where the full Jacobian is used and does not cover the diagonal approximation.

A. Convergence Analysis for TTGD

This subsection studies the convergence of TTGD. Throughout, we characterize the convergence behavior via the stationary measure for optimizing a smooth non-convex function ℓ over a closed convex set $\mathcal{S}$. For any $\gamma>0$, we define

$$\mathsf{G}_\gamma(\boldsymbol{S}):=\|\gamma^{-1}(\boldsymbol{S}-\mathsf{Proj}_{\mathcal{S}}(\boldsymbol{S}-\gamma\nabla\ell(\boldsymbol{S})))\|_F^2$$

as the squared Frobenius norm of the projected-gradient residual. A point $\boldsymbol{S}$ is called a stationary point of (16) if it satisfies

$0 \in \nabla \ell(\boldsymbol{S}) + \partial \iota_{\mathcal{S}}(\boldsymbol{S})$ [59]. Here, $\partial \iota_{\mathcal{S}}(\boldsymbol{S})$ is the subdifferential of the indicator function $\iota_{\mathcal{S}}$ at $\boldsymbol{S}$ [60]. From the definition of projection operators, it is obvious that if $\boldsymbol{S} = \mathsf{Proj}_{\mathcal{S}}(\boldsymbol{S} - \gamma \nabla \ell(\boldsymbol{S}))$ for some $\gamma > 0$, then $\boldsymbol{S}$ is a stationary point. When $\mathsf{G}_\gamma(\boldsymbol{S})$ is small, we expect that $\boldsymbol{S}$ is close to a stationary point.

We specify a few general conditions that are sufficient for the convergence of TTGD:

H1: The set $\mathcal{S}$ is convex and compact.

H2: The set $\mathcal{Y}$ is nonempty, closed, and convex. For any $\boldsymbol{S} \in \mathcal{S}$ and $\boldsymbol{y} \in \mathcal{Y}$, $\mathsf{F}(\boldsymbol{y}; \boldsymbol{S}) \in \mathcal{Y}$. Here, $\mathsf{F}(\boldsymbol{y}; \boldsymbol{S}) := \boldsymbol{y} - \mathsf{Y}(\boldsymbol{y}; \boldsymbol{S})$.

H3: There exists a $\mu_y \in (0, 1)$ such that for any $\boldsymbol{S} \in \mathcal{S}$ and $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}$, the following chain holds:

$$(1-\mu_y)\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2^2 \le \langle \mathsf{Y}(\boldsymbol{y}_1; \boldsymbol{S}) - \mathsf{Y}(\boldsymbol{y}_2; \boldsymbol{S}), \boldsymbol{y}_1 - \boldsymbol{y}_2 \rangle \le \|\mathsf{Y}(\boldsymbol{y}_1; \boldsymbol{S}) - \mathsf{Y}(\boldsymbol{y}_2; \boldsymbol{S})\|_2 \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2 \le (1+\mu_y)\|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2^2.$$

H4: The objective function $(\boldsymbol{S}, \boldsymbol{y}) \mapsto \Phi(\boldsymbol{S}; \boldsymbol{y})$ is smooth. In particular, there exists an $L_\Phi > 0$ such that for all $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \mathcal{S}$ and $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}$,

$$\|\nabla \Phi(\boldsymbol{S}_1; \boldsymbol{y}_1) - \nabla \Phi(\boldsymbol{S}_2; \boldsymbol{y}_2)\|_F \le L_\Phi \|(\boldsymbol{S}_1; \boldsymbol{y}_1) - (\boldsymbol{S}_2; \boldsymbol{y}_2)\|_F.$$

Here, $\nabla \Phi = (\nabla_{\boldsymbol{S}} \Phi; \nabla_{\boldsymbol{y}} \Phi)$ is the gradient of $\Phi(\boldsymbol{S}; \boldsymbol{y})$ taken w.r.t. $(\boldsymbol{S}; \boldsymbol{y})$.

H5: The mapping $(\boldsymbol{y}, \boldsymbol{S}) \mapsto \mathsf{Y}(\boldsymbol{y}; \boldsymbol{S})$ is smooth. In particular, for all $B > 0$, there exists an $L_{\mathsf{Y}} > 0$ such that for all $\boldsymbol{S}_1, \boldsymbol{S}_2 \in \mathcal{S}$ and $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}$ satisfying $\|\boldsymbol{y}_1\|_2, \|\boldsymbol{y}_2\|_2 \le B$, we have

$$\|\mathsf{J}\mathsf{Y}(\boldsymbol{y}_1; \boldsymbol{S}_1) - \mathsf{J}\mathsf{Y}(\boldsymbol{y}_2; \boldsymbol{S}_2)\|_F \le L_{\mathsf{Y}} \|(\boldsymbol{S}_1; \boldsymbol{y}_1) - (\boldsymbol{S}_2; \boldsymbol{y}_2)\|_F,$$

where $\mathsf{J}\mathsf{Y} = (\mathsf{J}_{\boldsymbol{y}}\mathsf{Y}; \mathsf{J}_{\boldsymbol{S}}\mathsf{Y})$ is the Jacobian of $\mathsf{Y}(\boldsymbol{y}; \boldsymbol{S})$ w.r.t. $(\boldsymbol{y}; \boldsymbol{S})$.

Note that a mapping is smooth if it has continuous partial derivatives of all orders. For a closed domain, smoothness indicates that the mapping is the restriction of a smooth mapping defined on an open neighborhood of that domain. For the relaxed map $\mathsf{F}_\alpha(\boldsymbol{y}; \boldsymbol{S}) = \boldsymbol{y} - \alpha \mathsf{Y}(\boldsymbol{y}; \boldsymbol{S})$, H2–H3 give

$$\|\mathsf{F}_\alpha(\boldsymbol{y}_1; \boldsymbol{S}) - \mathsf{F}_\alpha(\boldsymbol{y}_2; \boldsymbol{S})\|_2^2 \le \left[1 - 2\alpha(1-\mu_y) + \alpha^2(1+\mu_y)^2\right] \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|_2^2.$$

Thus, F_α is a contractive self-map for $0 < \alpha < 2(1-\mu_y)/(1+\mu_y)^2$, whereas F_1 need not be contractive. This yields a unique solution to $\mathsf{Y}(\boldsymbol{y}; \boldsymbol{S}) = 0$. For directed graphs, $\mathsf{J}_{\boldsymbol{y}}\mathsf{Y}$ may be asymmetric, but H3 implies that

$$\mathrm{Sym}(\mathsf{J}_{\boldsymbol{y}}\mathsf{Y}) \succeq (1-\mu_y) I, \qquad \sigma_{\min}(\mathsf{J}_{\boldsymbol{y}}\mathsf{Y}) \ge 1 - \mu_y,$$

i.e., the Jacobian remains nonsingular. Together with H1, it can be shown that $\{\boldsymbol{y}^k\}$ from updates (20a) with properly chosen α is bounded. Furthermore, H4 and H5 imply that the gradient map $\widehat{\nabla}\Phi(\cdot)$ is locally Lipschitz continuous w.r.t. $\boldsymbol{S}$ and $\boldsymbol{y}$, respectively. Together, $\widehat{\nabla}\Phi(\cdot)$ is Lipschitz continuous over the sequence $\{(\boldsymbol{S}^k, \boldsymbol{y}^k)\}$ generated by TTGD updates (20). Based on the above observations, we adapt the proof in [30] and obtain the following convergence result. The full proof is deferred to the supplementary material. We give the main argument below.

Theorem 1: Suppose that H1 to H5 hold. Let L_ℓ, $L_{\widehat{\Phi}}$, and L_y be the constants defined in the supplementary material. Set

$$\alpha = \frac{1-\mu_y}{(1+\mu_y)^2}, \qquad \gamma \le \min\left\{\frac{3}{4L_\ell}, \frac{\alpha(1-\mu_y)}{4L_{\widehat{\Phi}} L_y}\right\}.$$

Then, for any $K \ge 1$, we have

$$\min_{k=1,\ldots,K} \mathsf{G}_\gamma(\boldsymbol{S}^k) = \mathcal{O}(K^{-1}).$$

Proof sketch: The proof has three parts. We first show that the lower-level iteration (20a) is contractive, so $\boldsymbol{y}^k$ tracks $\boldsymbol{y}^\star(\boldsymbol{S}^k)$. We then use this to control the gap between the computable implicit gradient $\widehat{\nabla}\Phi(\boldsymbol{S}^k; \boldsymbol{y}^{k+1})$ and the exact gradient $\nabla \ell(\boldsymbol{S}^k)$. Finally, the Jacobian regularity assumptions ensure that the implicit gradient is well defined along the iterates. The result then follows from a projected descent argument. □

Network Games. We next provide sufficient conditions for satisfying H1 to H5 by specializing (16) to the network games settings in Sec. III-A. Throughout our discussion, we take $\mathcal{S} = \mathcal{S}_{\mathsf{ng}}$ as specified in (3) with the parameter $c > 0$. Define

$$\kappa_{\mathsf{ng}} := \max_{\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}} \|\boldsymbol{S}\|_2.$$

This constant is finite because $\mathcal{S}_{\mathsf{ng}}$ is compact. Moreover, $\boldsymbol{S}1 = c1$ implies that $c = \rho(\boldsymbol{S}) \le \|\boldsymbol{S}\|_2$, hence $\kappa_{\mathsf{ng}} \ge c$.

We first consider the assumptions about the functional regularizer $\mathcal{R}(\boldsymbol{S})$ induced by LQ games.

H6: Consider the payoff function $U_i^{lq}(\cdot)$ in (10) with $\mathcal{Y} = [0, \infty)^N$.

- For all $i \in [N]$, the marginal benefit satisfies $b_i \ge b^\star > 0$.
- The interaction function is smooth and satisfies $f(0) = 0$, $f(y) > 0$ for any $y > 0$.
- There exist $L_{f,1}, L_{f,2}$ such that the interaction function satisfies $|f'(y)| \le L_{f,1}$, $|f''(y)| \le L_{f,2}$ for any $y \in (0, \infty)$. Moreover, we have $L_{f,1} < \kappa_{\mathsf{ng}}^{-1}$.

Under the above assumption, the work [44] showed that an NE exists and is unique for any $\boldsymbol{S} \in \mathcal{S}$. Furthermore, we observe that

Proposition 1: Under H6, there exists a smooth map $\tilde{\mathsf{Y}}(\cdot)$ such that $\tilde{\mathsf{Y}}(\boldsymbol{y}; \boldsymbol{S}) = \mathsf{Y}(\boldsymbol{y}; \boldsymbol{S})$ for any $\boldsymbol{y} \in \mathcal{Y}$, $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$. Furthermore, H1 to H5 hold for the special case of (GL-NG) with $\tilde{\mathsf{Y}}(\cdot)$ and $\Phi(\cdot)$.

The full proof is deferred to the supplementary material. We next consider RT games to instantiate the functional regularizer design:

H7: Consider the payoff function $U_i^{rt}(\cdot)$ in (11) with $\mathcal{Y} = \prod_{i=1}^N [\underline{b}_i, a_i] := \{\boldsymbol{y} \in \mathbb{R}^N : y_i \in [\underline{b}_i, a_i], \forall\, i \in [N]\}$.

- For all $i \in [N]$, the marginal benefit and action bounds satisfy $0 < \underline{b}_i \le b_i < a_i$ and $b_i \ge b^\star > 0$.
- With $a_{\max} := \max_i a_i$, the interaction function is smooth and satisfies $g(0) = 0$, $g(x) > 0$ for any $x \in (0, c\, a_{\max}]$.
- There exist $L_{g,1}, L_{g,2}$ such that the interaction function satisfies $|g'(x)| \le L_{g,1}$, $|g''(x)| \le L_{g,2}$ for any $x \in [0, c\, a_{\max}]$. Moreover, we have

$$L_{g,1} < \min\left\{\kappa_{\mathsf{ng}}^{-1}, \min_{i \in [N]} \frac{a_i - b_i}{c\, a_{\max}}\right\}.$$

Under the above assumption, the work [49] showed that an NE exists and is unique for any $\boldsymbol{S} \in \mathcal{S}$. Similarly, we observe that

Proposition 2: Under H7, there exists a smooth map $\tilde{\mathsf{Y}}(\cdot)$ such that $\tilde{\mathsf{Y}}(\boldsymbol{y}; \boldsymbol{S}) = \mathsf{Y}(\boldsymbol{y}; \boldsymbol{S})$ for any $\boldsymbol{y} \in \mathcal{Y}$, $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$. Furthermore, H1 to H5 hold for the special case of (GL-NG) with $\tilde{\mathsf{Y}}(\cdot)$ and $\Phi(\cdot)$.

Under H6 and H7, respectively, Theorem 1 applies to the LQ and RT instances of (GL-NG).

Remark 2: For the graph learning formulation (GL-GENE) considered in Sec. III-B, H2 and H3 are violated whenever the system $\mathsf{Y}(\boldsymbol{y};\boldsymbol{S})=0$ admits a nontrivial solution $\boldsymbol{y}\neq 0$ alongside the persisting trivial solution $\boldsymbol{y}=0$. Since the regime with nontrivial stationary states is precisely the case of interest in GL-GENE, Theorem 1 does not provide a convergence guarantee for TTGD when it is applied to (GL-GENE). Nevertheless, the numerical results in Sec. VI-B suggest that TTGD can still be used as a heuristic and produce meaningful graph estimates in this more challenging regime.

V. Structural Interpretation for $\mathcal{R}(S)$

This section investigates the relationship between functional priors and structural priors on graph learning. We note that as (GLFP) gives a *biased* graph estimator with $\lambda>0$, even when $M\gg 1$, an optimal solution to (GLFP) may not coincide with the true underlying graph. Thus, it may be futile to discuss exact recoverability with (GLFP). In this section, we focus on analyzing the optimal solutions to graph learning problems that are regularized by functional priors and suggest the types of graph structure they induce. As bilevel optimization problems are often difficult to analyze, we concentrate on the two case studies in Sec. III and consider the approximations to the latter.

Network Games. Our first step is to construct an approximation of (GL-NG) that is amenable to analysis. Consider the following properties, which are implied by H6, H7 in the previous section:

1) The functions $f(\cdot), g(\cdot)$ are ℓ_1-Lipschitz continuous with $\ell_1<\frac{1}{c}$.
2) We have $f(y)>0$ for any $y>0$, whereas $g(y)>0$ for any $y\in(0, c\,a_{\max}]$.
3) For any $\boldsymbol{S}\in\mathcal{S}_{\mathsf{ng}}$, $\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})$ is in the interior of $\mathcal{Y}$.

Observe the following proposition:

Proposition 3: Consider the bilevel problem (GL-NG). Under the above conditions, we have

$$\|\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})\|_2\leq B^\star=\frac{\sqrt{N}\,\|\boldsymbol{b}\|_\infty}{1-c\ell_1},\ \forall\,\boldsymbol{S}\in\mathcal{S}_{\mathsf{ng}}.$$

Consequently, for any $\boldsymbol{S}\in\mathcal{S}_{\mathsf{ng}}$, we have

$$\begin{aligned}&\mathrm{Tr}(\boldsymbol{S}^\top\boldsymbol{D})+\beta\|\boldsymbol{S}\|_F^2-\lambda\ell_1 1^\top\boldsymbol{S}\boldsymbol{b}-\lambda 1^\top\bar{\boldsymbol{c}}\\&\leq\mathrm{Tr}(\boldsymbol{S}^\top\boldsymbol{D})+\beta\|\boldsymbol{S}\|_F^2-\lambda 1^\top\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})\\&\leq\mathrm{Tr}(\boldsymbol{S}^\top\boldsymbol{D})+\beta\|\boldsymbol{S}\|_F^2-\lambda\mu_1 1^\top\boldsymbol{S}\boldsymbol{b}-\lambda 1^\top\hat{\boldsymbol{c}},\end{aligned}\tag{21}$$

where $\bar{\boldsymbol{c}}=\frac{(c\ell_1)^2}{1-c\ell_1}\|\boldsymbol{b}\|_\infty\mathbf{1}+\boldsymbol{b}$, $\hat{\boldsymbol{c}}=\frac{(c\mu_1)^2}{1-c\mu_1}b^\star\mathbf{1}+\boldsymbol{b}$ with $b^\star=\min_{i\in[N]}b_i$. For LQ games, $\mu_1=\frac{\min_{y\in[b^\star,B^\star]}f(y)}{B^\star}$. For RT games, $\mu_1=\frac{\min_{y\in[b^\star,B_{\mathrm{RT}}]}g(cy)}{cB_{\mathrm{RT}}}$, where $B_{\mathrm{RT}}:=\min\{B^\star,a_{\max}\}$.

The proof is relegated to Appendix A-A. Eq. (21) implies that (GL-NG) can be approximated by

$$\min_{\boldsymbol{S}\in\mathcal{S}_{\mathsf{ng}}}\ \mathrm{Tr}(\boldsymbol{S}^\top\boldsymbol{D})+\beta\|\boldsymbol{S}\|_F^2-\tilde{\lambda}\mathbf{1}^\top\boldsymbol{S}\boldsymbol{b}\tag{GL-NG-App}$$

for some $\tilde{\lambda}>0$ proportional to λ. Importantly, (GL-NG-App) is a convex program with affine constraints. Analyzing the KKT conditions of (GL-NG-App) leads to

Proposition 4: There exists an $\boldsymbol{\eta}\in\mathbb{R}^N$ such that any optimal solution to Problem (GL-NG-App) is given by

$$S_{ij}^\star=\tfrac{1}{2\beta}\max\left\{0,\tilde{\lambda}b_j+\eta_i-D_{ij}\right\}.$$

for any $i\neq j$ and $S_{ii}^\star=0$. It also holds that $\boldsymbol{S}^\star\mathbf{1}=c\mathbf{1}$.

The proof is in Appendix A-B.

Together, Propositions 3 and 4 show that when $\tilde{\lambda}\gg 1$, an optimal solution to (GL-NG) can be approximated as $S_{ij}^\star\approx\frac{\tilde{\lambda}}{2\beta}b_j$ for any $i,j\in V$. As the marginal benefits $\boldsymbol{b}$ may vary between agents in practice, any optimal solution to (GL-NG-App) (and thus (GL-NG)) gives a graph topology that exhibits a *'multiple hub' structure* where the majority of edges will be emanating from nodes with large b_j. This observation coincides with the human-made network examples studied in Table I, where we observe a number of 'hub' nodes.

Proposition 4 provides a structural characterization of the biased estimator induced by the functional prior, rather than a guarantee of exact topology recovery. When $\lambda>0$, the functional prior introduces a task-aligned bias, so the learned graph should be understood as balancing data fidelity and the prescribed functional objective. In particular, Proposition 4 explains why hub-promoting structures emerge in the large-λ regime, which is consistent with the network-formation viewpoint underlying the functional prior.

Network Dynamics. Similar to the previous case, we proceed to derive an approximation of (GL-GENE).

Proposition 5: Consider the problem (GL-GENE) and denote its optimal value by $\upsilon_{\mathsf{GENE}}^\star$. We have

$$\upsilon_{\mathsf{GENE}}^\star\geq\min_{\boldsymbol{S}\in\mathcal{S}_{\mathsf{nd}}}\ \|\boldsymbol{S}\boldsymbol{X}+\boldsymbol{P}\|_F^2+\beta\|\boldsymbol{S}\|_F^2-\hat{\lambda}1^\top\boldsymbol{S}^2 1-\frac{N\lambda}{2},\tag{GL-GENE-App}$$

where $\hat{\lambda}=\frac{\sigma\lambda}{8}$.

The proof is relegated to Appendix A-C. Proposition 5 gives a one-sided approximation of (GL-GENE). This approximation is less tight compared to that of Proposition 3. Still, it provides useful insights into the optimal solution to (GL-GENE). Additionally, we remark that the regularization term in (GL–GENE-App) is related to the resilience quantity proposed in [29], which is defined as

$$\beta_{\mathrm{eff}}(\boldsymbol{S})=\frac{1^\top\boldsymbol{S}d(\boldsymbol{S})}{1^\top\boldsymbol{S}1},\qquad d(\boldsymbol{S}):=\boldsymbol{S}1.$$

This proxy reflects a degree-heterogeneity effect at the topology level. Under $1^\top\boldsymbol{S}1=a$, $\beta_{\mathrm{eff}}(\boldsymbol{S})$ is proportional to the regularization term $1^\top\boldsymbol{S}^2 1$ in (GL-GENE-App).

The single-level problem (GL-GENE-App) remains nonconvex in general due to the term $-\mathbf{1}^\top\boldsymbol{S}^2\mathbf{1}$. Therefore, we concentrate on the solutions that directly maximize $\mathbf{1}^\top\boldsymbol{S}^2\mathbf{1}$, which correspond to the regime with $\lambda\gg 1$. This observation motivates the structural characterization:

Proposition 6: For any $\boldsymbol{S}^\star\in\arg\max_{\boldsymbol{S}\in\mathcal{S}_{\mathsf{nd}}}1^\top\boldsymbol{S}^2 1$, there exists a pair $(i^\star,j^\star)$ such that $i^\star\neq j^\star$ and

$$S_{ij}^\star=\begin{cases}\frac{a}{2}, & \text{if } (i,j)=(i^\star,j^\star)\text{ or }(j^\star,i^\star),\\ 0, & \text{otherwise.}\end{cases}$$

The proof can be found in Appendix A-D. In other words, when $\lambda \gg 1$, every optimal solution to (GL-GENE-App) corresponds to a graph with only one bidirected edge.

As with Proposition 4, this result should be interpreted as a structural characterization of the prior-dominated estimator in the large-λ regime, rather than as a claim that the underlying GRN topology is recovered by such a one-edge structure. In practice, the regularization parameter λ is moderate, so the actual estimator balances data fidelity against the resilience-promoting bias rather than collapsing to the limiting one-edge solution.

A. Functional Priors as Generalized Graph Filter Priors

Our last endeavor is to interpret the functional priors via the conduit of generalized graph filter priors. In particular, we show that any polynomial regularizer of the GSO $\boldsymbol{S}$ can be written as a functional prior regularizer.

To facilitate our discussion, we first introduce the notion of graph filter (GF) regularizers for use in (1) to induce structural properties in the learned graph. Formally

Definition 1: A graph filter (GF) regularizer $\mathcal{R}_{\mathsf{GF}} : \mathbb{R}^{N\times N} \to \mathbb{R}$ is

$$\mathcal{R}_{\mathsf{GF}}(\boldsymbol{S}) = f \circ H(\boldsymbol{S}).$$

Here, $f : \mathbb{R}^{N\times N} \to \mathbb{R}$ is a linear operator and $H : \mathbb{R}^{N\times N} \to \mathbb{R}^{N\times N}$ is a linear graph filter defined as $H(\boldsymbol{S}) = h_0\boldsymbol{I} + \sum_{i=1}^{p} h_i \boldsymbol{S}^i$, where p is the filter order and $\{h_i\}_{i=0}^{p}$ are the filter's coefficients [38].

The class of GF regularizers is general enough to cover most of the structural regularizers. We consider two examples.

Example 1: For a symmetric GSO, the structural regularizer $\mathcal{R}(\boldsymbol{S}) = \|\boldsymbol{S}\|_F^2$ satisfies $\|\boldsymbol{S}\|_F^2 = \mathrm{Tr}(\boldsymbol{S}^2)$. Hence, one may take $f(\cdot) = \mathrm{Tr}(\cdot)$ and $H(\boldsymbol{S}) = \boldsymbol{S}^2$ [6].

Example 2: The spectral regularizer $\mathcal{R}(\boldsymbol{S})$ is designed to regulate the clustering properties of a graph and can be expressed as a function of singular values of $\boldsymbol{S}$ (e.g. (7)) [25]. Note that for a symmetric GSO, $\boldsymbol{S}$ admits an eigen-decomposition $\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^\top$ and $H(\boldsymbol{S}) = \boldsymbol{U}(\sum_{i=1}^{N} h_i \boldsymbol{\Lambda}^i)\boldsymbol{U}^\top$. Hence, a spectral regularizer can be expressed as a GF regularizer with $f(\cdot) = \mathrm{Tr}(\cdot)$ and a carefully designed low-pass graph filter H that attenuates the high-pass part of the singular values of $\boldsymbol{S}$.

In fact, the approximate forms of the bilevel problems described above, i.e., (GL-NG-App) and (GL-GENE-App), are also special cases of GF regularizers. For the term $-1^\top \boldsymbol{S}\boldsymbol{b}$ in Proposition 3, we have $f(\cdot) = -1^\top(\cdot)\boldsymbol{b}$ and $H(\boldsymbol{S}) = \boldsymbol{S}$. For the term $-1^\top \boldsymbol{S}^2 1$ in Proposition 5, we have $f(\cdot) = -1^\top(\cdot)1$ and $H(\boldsymbol{S}) = \boldsymbol{S}^2$.

Finally, we conclude with the following lemma, which shows that any GF regularizer can be expressed as a functional prior regularizer, and the corresponding graph learning problem can be written as a special case of (16).

Lemma 1: For any GF regularizer $\mathcal{R}_{\mathsf{GF}}(\boldsymbol{S}) = f \circ H(\boldsymbol{S})$, there exists a $\mathsf{Y} : \mathbb{R}^N \times \mathbb{R}^{N\times N} \to \mathbb{R}^N$ such that for all $\boldsymbol{S} \in \mathbb{R}^{N\times N}$,

$$\mathcal{R}_{\mathsf{GF}}(\boldsymbol{S}) = 1^\top \boldsymbol{y}(\boldsymbol{S}) \tag{22}$$

and $\boldsymbol{y}(\boldsymbol{S})$ is the unique solution to $\mathsf{Y}(\boldsymbol{y}; \boldsymbol{S}) = 0$.

Proof: As $f : \mathbb{R}^{N\times N} \to \mathbb{R}$ is a linear operator, the Riesz representation theorem [61] shows that there exists a constant matrix $\boldsymbol{Z} \in \mathbb{R}^{N\times N}$ such that $f(H(\boldsymbol{S})) = \mathrm{Tr}(\boldsymbol{Z}^\top H(\boldsymbol{S})) = 1^\top \boldsymbol{y}$, where $\boldsymbol{y} = \mathsf{diag}(\boldsymbol{Z}^\top H(\boldsymbol{S}))$. Subsequently, we may choose $\mathsf{Y}(\cdot; \boldsymbol{S})$ as the mapping $\mathsf{Y}(\boldsymbol{y}; \boldsymbol{S}) = \boldsymbol{y} - \mathsf{diag}(\boldsymbol{Z}^\top H(\boldsymbol{S}))$. Observe that the desired condition (22) holds. □

VI. Numerical Experiments

This section presents numerical experiments to validate the effectiveness of the proposed graph learning method using functional priors induced by network games in Sec. III-A and gene network dynamics in Sec. III-B. Throughout, the area under the receiver operating characteristic curve (AUC) is used as the primary metric for graph-support recovery. Since the ground-truth graphs are evaluated through their binary supports and the learned continuous edge weights are used as ranking scores, AUC does not measure precise edge-weight recovery. Thus, complementary weight- and support-recovery metrics are reported in the supplementary material. Unless otherwise stated, TTGD is terminated when

$$\frac{\|\boldsymbol{S}^{k+1} - \boldsymbol{S}^k\|_F}{\max\{1, \|\boldsymbol{S}^k\|_F\}} \le 10^{-6}.$$

The parameter β is selected from the grid $\{1, 5, 10, 50, 100, 200\}$ based on the validation AUC, separately for each experiment. The step sizes (α, γ) are tuned for stable TTGD updates, while c is fixed as the normalization parameter in the feasible set $\mathcal{S}_{\mathsf{ng}}$. The assumptions in Sec. IV provide sufficient conditions for convergence and are not explicitly enforced in every numerical setting.

A. Network Games

In this subsection, we focus on the case with priors induced by network games in (GL-NG).

• **Synthetic Data.** Our first experiment evaluates the graph learning performance using synthetic random graphs and synthetic graph signals. We generate $\mathcal{G}$ as a preferential attachment (PA) graph with $N = 50$ nodes and one edge to attach for every new node. The probability of attaching to an existing node is proportional to its degree, normalized over the total degree, i.e., $P(i) = d_i / \sum_j d_j$. Additionally, we generate $\mathcal{G}$ as an Erdős–Rényi (ER) graph with $N = 50$ nodes and connection probability of 0.1. We concentrate on a scenario with limited data acquired, where only $M = 10 \ll N$ smooth graph signals are observed. Each graph signal is generated via a low pass graph filter as $\boldsymbol{x}_m = \exp(\boldsymbol{S}/2)\boldsymbol{u}_m + \boldsymbol{w}_m$, where $\boldsymbol{u}_m \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ is an i.i.d. white noise excitation and $\boldsymbol{w}_m \sim \mathcal{N}(\boldsymbol{0}, \omega^2\boldsymbol{I})$ is an i.i.d. additive noise with $\omega = 0.2$. We also fix $(\beta, c) = (200, 0.95)$ for (GL-NG).

We benchmark (GL-NG) against four methods: (i) Smooth-GL method [6], (ii) approximate bilevel problem in (GL-NG-App) (referred as linear approx.), (iii) QuadGame-GL method [62], which learns network structure from network games model; (iv) SpecTemp method [12], which uses the

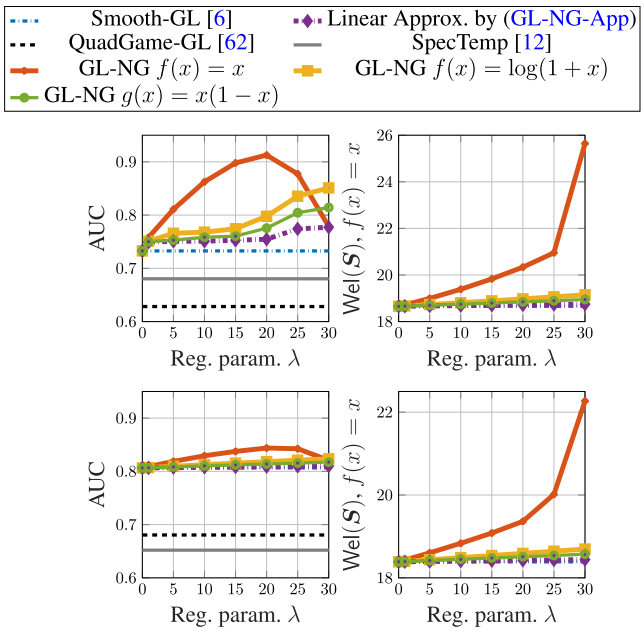


Fig. 3. Performance of (GL-NG) using TTGD for learning from PA graphs (top) and ER graphs (bottom). Left: AUC scores. Right: Social welfare values.

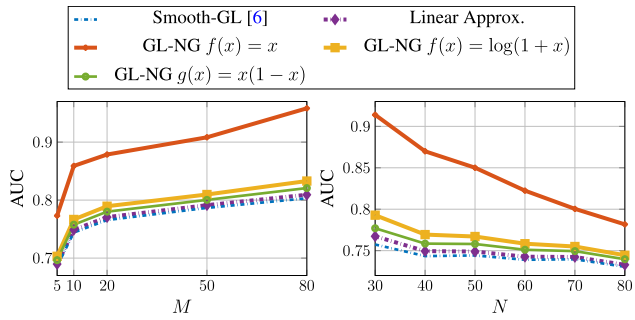


Fig. 4. Performance of TTGD applied to (GL-NG) on PA graphs. Left: AUC versus M with $N = 50$. Right: AUC versus N with $M = 10$. In both panels, $\lambda = 10$.

spectral signature of the covariance matrix. For (GL-NG), the network games (10) and (11) are specified with the normalized marginal benefit vector $\boldsymbol{b} = \max(\boldsymbol{v}_1, 0)/\|\max\{\boldsymbol{v}_1, 0\}\|_1$, where $\boldsymbol{v}_1$ is the top eigenvector of the Euclidean distance matrix $\boldsymbol{D}$. The matrix $\boldsymbol{D}$ is given by $D_{ij} = \|\boldsymbol{x}_i^{\text{row}} - \boldsymbol{x}_j^{\text{row}}\|_2^2$. This data-derived choice is used only as a simple proxy for heterogeneous intrinsic incentives. When no useful node-level information is available, a homogeneous vector $\boldsymbol{b} \propto 1$, normalized in the same manner, provides a natural default. For the RT experiments, we set $\boldsymbol{a} = 1$ and $\underline{b}_i = b_i$ for all $i \in [N]$.

For the linear-quadratic (LQ) game (10), we consider two different interaction functions, namely $f(x) = x$ and $f(x) = \log(1+x)$; for the race & tournament (RT) game (11), we consider the interaction function $g(x) = x(1-x)$. These choices encode different peer-effect assumptions: $f(x) = x$ gives additive unsaturated spillovers, $f(x) = \log(1+x)$ gives diminishing marginal spillovers, and $g(x) = x(1-x)$ gives a non-monotone response to aggregate peer activity. In our experiments, the bilevel optimization problem is tackled using TTGD with the following step sizes. For LQ games with $f(x) = x$, $f(x) = \log(1+x)$, the step sizes are $(\alpha, \gamma) = (0.5, 0.003)$, while for RT games with $g(x) = x(1-x)$, the step sizes are $(\alpha, \gamma) = (0.1, 0.0005)$. The algorithm terminates after 700 iterations for $f(x) = x$ and 195 iterations for $f(x) = \log(1+x)$. For $g(x) = x(1-x)$, the algorithm terminates after 250 iterations.

Fig. 3 shows the performance of graph learning algorithms against the regularization parameter λ, averaged over 20 Monte-Carlo trials. Note that as λ increases, the regularized graph learning objective will become more dependent on the network games prior and, according to Proposition 4, (GL-NG) tends to learn a graph topology with few hub nodes.

We concentrate on the performance of (GL-NG). For PA graphs with hub structures, we observe substantial improvement in AUC as λ increases. As a control, for ER graphs that lack obvious hub structures, the improvement in AUC is only modest. The social welfare consistently improves for both graph types as λ grows. These results corroborate that of Proposition 4, which shows that with $M \ll N$, the functional regularizer in (GL-NG) promotes a hub structure and improves the graph learning performance. We next examine how the performance varies with the sample size and the network size in Fig. 4. The results show a consistent advantage of the proposed GL-NG model with $f(x) = x$: Its AUC improves steadily as M increases, while it remains clearly superior to the competing methods as N increases. The figure in the supplementary material reports the runtime as N increases. It shows the expected increase in cost for the full TTGD update.

Our second experiment considers synthetic graph signal observations while focusing on $\mathcal{G}$ given by the `Karate Club` graph, which consists of $N = 34$ nodes. We have $M = 50$ samples of smooth graph signals generated from the Gaussian Markov Random Field (GMRF) model with the precision matrix given by the graph Laplacian [5]. Our aim is to showcase the necessity of tackling the bilevel problem (GL-NG) using TTGD instead of the single-level optimization approximation (GL-NG-App). Fig. 5 shows the *Pareto fronts* of the bilevel solution and the approximate solution, computed by varying the regularization parameter λ that trades off between the smooth-GL objective $J(\boldsymbol{S}; \boldsymbol{X})$ and Wel($\boldsymbol{S}$). As expected, we observe that the TTGD algorithm achieves a better Pareto front than the approximate solution in all cases. The three welfare functions have different scales, so the vertical axes should be compared only within each panel; a better front gives higher welfare at a comparable data-fitting loss.

• **Real Data: Case Study I.** We evaluate our proposed TTGD algorithm on two real-world social network datasets in the limited data setting, characterized by $M \ll N$.

The `Groningen` dataset contains $M = 13$ signals taken from a friendship network of students in University of Groningen with $N = 38$ nodes in 1996. The signals capture each student's level of interest in social activities (e.g., attending concerts and movies). We set $\beta = 10$ and define $\mathbf{b}$ to encode

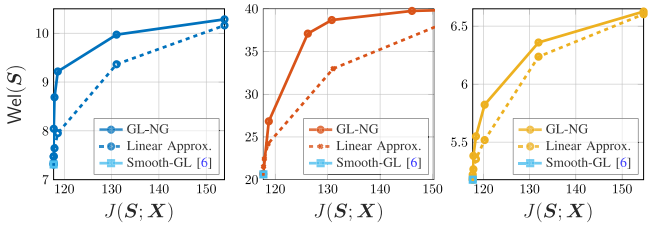


Fig. 5. Comparing the social welfare $\text{Wel}(\boldsymbol{S}) = \mathbf{1}^\top \boldsymbol{y}^{\text{NE}}(\boldsymbol{S})$ against the data fidelity term $J(\boldsymbol{S}; \boldsymbol{X})$ for (GL-NG). Left: Prior with $f(x) = \log(1+x)$. Middle: Prior with $f(x) = x$. Right: Prior with $g(x) = x(1-x)$.

the study program, where Program $= 1$ corresponds to the 4-year track and Program $= 2$ to the 2-year track. Additionally, we normalize $\mathbf{b}$ as $\mathbf{b} = \mathbf{b}/\mathbf{1}^\top\mathbf{b}$. The `Dutch` dataset comprises $N = 26$ nodes and $M = 7$ signals, representing the friendship network of teenagers in a Dutch school in 2003–2004. The signals measure negative behavior, such as the frequency of stealing or alcohol consumption. Again, we set $\beta = 10$ and let $\mathbf{b}$ encode ethnicity, where Ethnicity $= 1$ is Dutch and Ethnicity $= 2$ otherwise. We then normalize $\mathbf{b}$ accordingly. These covariates are used as simple, interpretable proxies for heterogeneous intrinsic incentives and are not claimed to be causal determinants of the observed friendship networks.

Table II presents the highest AUC values by tuning λ for our (GL-NG)-based methods, alongside the corresponding social welfare values. We observe that in both datasets, (GL-NG)-based methods consistently outperform other benchmark methods in terms of AUC and welfare. Specifically, for `Groningen`, the LQ game induced prior with the interaction $f(x) = x$ achieves the best AUC performance. For `Dutch`, the RT game induced prior with the interaction $g(x) = x(1-x)$ gives the best AUC performance. A cautious interpretation is that `Groningen`'s cooperative-activity signals are compatible with additive spillovers, whereas the `Dutch` risky-behavior signals may be better represented by bounded, non-monotone aggregate effects. Note that we anticipate different real-world network data may inherently have different interaction dynamics. Nevertheless, our (GLFP) framework improves performance by incorporating functional priors.

• **Real Data: Case Study II.** Our second set of real data experiments considers learning the graph topologies from a set of real data taken from Indian villages [63]. The dataset consists of survey data from 40 villages, each taken as an individual graph to be learned by (GL-NG). Here, the networks have sizes ranging from $N = 77$ to $N = 330$ agents, and $M = 16$ samples of graph signals are observed for each network.

We set the parameters in (GL-NG) with $\lambda = 100$, $\beta = 50$, $\boldsymbol{b} = \mathtt{h} + 1$, where $\mathtt{h}_i \in \{0, 1\}$ encodes the microfinance-related survey indicator ' `hhSurveyed`' and $\boldsymbol{b}$ is normalized such that $\boldsymbol{b}^\top 1 = 1$. Fig. 6 shows the boxplots of AUC values for graph topologies learned under different settings and algorithms, compared against the ground-truth network reported in [63]. We observe that (GL-NG) generally yields higher AUC values than the benchmark methods. Similarly, Table III reports the maximum/average/minimum gain in social welfare relative to the

TABLE II
COMPARING THE AUC AND WELFARE OF DIFFERENT GRAPH LEARNING MODELS ON GRONINGEN AND DUTCH

	AUC	Wel($\boldsymbol{S}$) by x	Wel($\boldsymbol{S}$) by $\log(1+x)$	Wel($\boldsymbol{S}$) by $x(1-x)$
`Groningen` dataset				
GL-NG ($f(x) = x$)	**0.6357**	**27.7721**	**8.7035**	5.9668
GL-NG ($f(x) = \log(1+x)$)	0.6207	26.5906	8.6225	5.9557
GL-NG ($g(x) = x(1-x)$)	0.6201	26.5722	8.6432	**5.9739**
Linear Approx	0.5354	20.0577	7.5192	5.4005
Smooth-GL [6]	0.5354	20.0566	7.5190	5.4003
QuadGame-GL [62]	0.5490	14.5312	5.7325	4.2091
SpecTemp [12]	0.4890	19.7591	7.4576	5.3658
Glasso [4]	0.5617	16.6454	6.4513	4.6939
`Dutch` dataset				
GL-NG ($f(x) = x$)	0.6127	21.3681	6.7664	4.6940
GL-NG ($f(x) = \log(1+x)$)	0.6159	22.5223	6.9465	4.7810
GL-NG ($g(x) = x(1-x)$)	**0.6186**	**23.0427**	**7.0260**	**4.8187**
Linear Approx	**0.6186**	21.4615	6.7831	4.7030
Smooth-GL [6]	0.5840	18.6902	6.3571	4.6357
QuadGame-GL [62]	0.6048	18.3942	5.9083	4.1413
SpecTemp [12]	0.4828	20.6126	6.6480	4.6357
Glasso [4]	0.5786	18.1852	5.3394	2.7934

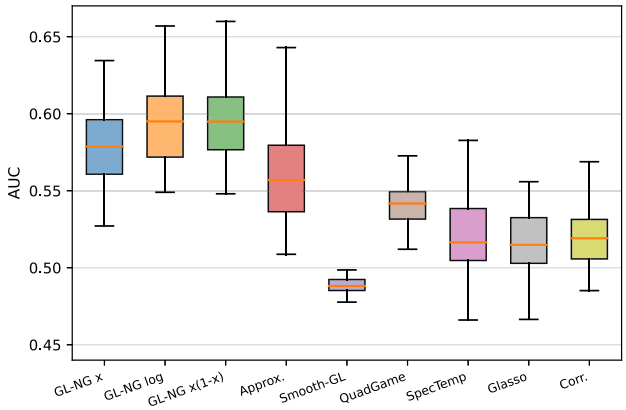


Fig. 6. Comparing the AUC performance of the graph learnt from `IndianVillage` [63].

ground truth. We observe that the graphs learned by (GL-NG) attain higher welfare values under the prescribed network-game model, which is consistent with the role of the functional prior in steering the estimator toward graph structures that reach high values on the target task. A natural network may settle for a compromise among multiple competing objectives, whereas our regularized estimator emphasizes a prescribed functional aspect. It is therefore possible for the learned graph to attain a higher task-related score; see also Sec. V. The structural recovery quality is evaluated separately through AUC.

B. Gene Regulatory Dynamics

In this subsection, we focus on the case with priors induced by gene regulatory dynamics in (GL-GENE).

• **Synthetic Data (DREAM4).** The first experiment considers the *In-Silico* (i.e., synthetic) data from the DREAM4 Challenge. The DREAM4 networks are sub-networks of curated transcriptional regulatory networks from E. coli (RegulonDB 6.0 [64]) and S. cerevisiae [65], with all self-loops removed. The dataset was used in the DREAM4 network inference challenge and contains $N = 100$ genes and $M = 100$ perturbation

TABLE III
GAIN IN SOCIAL WELFARE OF THE GRAPHS LEARNED FROM THE INDIANVILLAGE DATA [63]. THE THREE ROW BLOCKS REPORT WELFARE GAINS EVALUATED UNDER THE INTERACTION FUNCTIONS $f(x) = x$, $f(x) = \log(1 + x)$, AND $g(x) = x(1 - x)$, RESPECTIVELY

$\mathsf{Wel}(\hat{\boldsymbol{S}}) - \mathsf{Wel}(\boldsymbol{S}^{\mathsf{true}})$	**Maximum**	**Average**	**Minimum**
GL-NG ($f(x) = x$)	**4.2754**	**3.2077**	**2.1693**
Linear Approx.	-0.4790	-1.4241	-2.3927
Smooth-GL [6]	-2.3288	-5.8141	-8.6115
QuadGame-GL [62]	-1.9386	-3.5631	-5.5949
SpecTemp [12]	0.2653	-2.8122	-6.6532
Glasso [4]	-0.7666	-4.1383	-10.3377
Correlation	-2.6829	-3.3693	-4.4143
GL-NG ($f(x) = \log(1 + x)$)	**1.4464**	**1.1892**	**0.8461**
Linear Approx.	-0.0676	-0.6144	-1.1429
Smooth-GL [6]	-0.5440	-2.8292	-5.0419
QuadGame-GL [62]	-0.8206	-1.6814	-3.3074
SpecTemp [12]	0.1453	-1.2226	-3.6324
Glasso [4]	-0.3700	-2.0347	-4.9003
Correlation	-0.7413	-1.4614	-1.9076
GL-NG ($g(x) = x(1 - x)$)	**1.0312**	**0.7769**	**0.5476**
Linear Approx.	-0.0292	-0.4454	-0.9317
Smooth-GL [6]	-0.4165	-1.0737	-1.7126
QuadGame-GL [62]	-0.5334	-1.2174	-2.5642
SpecTemp [12]	0.1147	-0.8539	-2.6901
Glasso [4]	-0.2443	-1.4959	-3.7367
Correlation	-0.4475	-1.0201	-1.3848

experiments. Note that the effect of knockouts in GRNs is complex and the data-fitting loss (4) is a simplification. This experiment pertains to the case with imprecise data models. In the following, we consider (GL-GENE) with the parameters $a = 240$, $\beta = 10, \sigma = 1$.

We also report the level of resilience for each learned graph topology subject to random edge removal. We simulate $n_{\mathsf{per}} = 200$ perturbations by randomly deleting edges from the learned graph with probability $p \in \{0, 0.1\}$, producing a perturbed topology set $\hat{\mathcal{S}}$. We then compute the expected resilience metric as $\mathbb{E}[\mathsf{Rob}(\boldsymbol{S})] = n_{\mathsf{per}}^{-1} \sum_{\boldsymbol{S} \in \hat{\mathcal{S}}} \mathsf{Rob}(\boldsymbol{S})$.

Fig. 7 reports the AUC together with the corresponding resilience values of the graph topology found by (GL-GENE) as λ varies within $[0, 500]$, which illustrates the effect of random edge removal on resilience. Recall that under the fixed-total-weight constraint $1^\top \boldsymbol{S} 1 = a$, the topology-level resilience proxy in [29] reduces to the same degree-based score underlying (GL-GENE-App). Hence, the empirical performance of (GL-GENE-App) serves as an indirect evaluation of the topology-level mechanism highlighted in [29]. As a benchmark, we compare the performance of (GL-GENE-App), the ridge regression [39], [40], and the resilience of the ground truth topology. We additionally tested Correlation and SmoothGL in the same perturbation-response setting; their DREAM4 AUC values are only 0.2363 and 0.2359, respectively. As expected, network-game-based baselines perform substantially worse in this GRN setting, which confirms that the issue is not simply the choice of baseline, but the mismatch between their modeling assumptions and the perturbation-response nature of the GRN data. For easier comparison, we use a scaled regularization parameter in (GL-GENE-App) that is set as $\hat{\lambda} = \lambda/1600$.

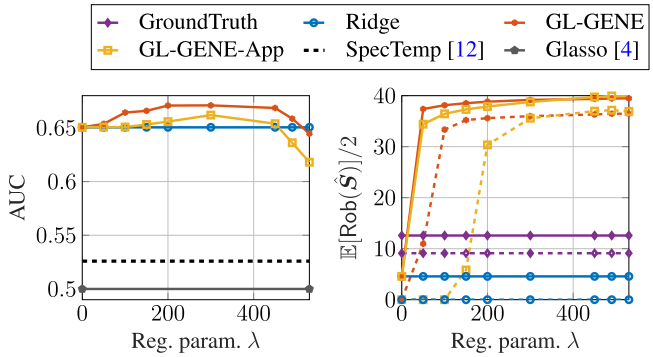


Fig. 7. Performance of algorithms on DREAM4. Left: AUC. Right: Expected resilience metric $\mathbb{E}[\mathsf{Rob}(\hat{\mathbf{S}})]$ across λ. Dashed line represents the perturbation set with $p = 0.1$.

We observe from the figure that both the proposed GL-GENE method and its approximation lead to improved AUC and robustness of the learned graphs. The main takeaway is the trade-off across different λ: Moderate regularization improves AUC and resilience, whereas excessive regularization eventually reduces AUC as the estimator overemphasizes the resilience prior. This behavior can be viewed as prior overfitting. A resilience value above the ground truth is therefore not evidence of more accurate topology recovery.

• **Real Data (DREAM5 *E. coli*).** The second experiment considers the *In-Vivo* (i.e., real) data from the DREAM5 challenge [66]. The dataset contains gene expression data from *E. coli* under various perturbation conditions, with $N = 4511$ genes and $M = 56$ perturbation conditions. The steady-state expression levels are treated as the observed data, denoted by $\boldsymbol{X} \in \mathbb{R}^{4511 \times 56}$. We consider (GL-GENE) with the parameters $a = 3500, \beta = 1, \sigma = 100$. For the small- and moderate-scale experiments above, we compute the implicit gradient by directly solving the full adjoint system. This is computationally manageable at the corresponding dimensions and avoids introducing an additional hypergradient approximation. For DREAM5, where $N = 4511$, solving the full system becomes substantially more expensive. We therefore report both the full-Jacobian GL-GENE variant and the diagonal approximation DA-GLFP (cf. Remark 1). The initialization of $\boldsymbol{y}$, $\boldsymbol{S}$ for (GL-GENE-App), (GL-GENE), and DA-GLFP are all taken as the output of ridge regression. Each algorithm is run for 5000 iterations. On a laptop computer equipped with Apple M1 Pro, it takes 93 minutes for (GL-GENE), while it only takes 44 minutes for ridge regression, 47 minutes for (GL-GENE-App), and 48 minutes for DA-GLFP. We ignore GLASSO in this setup as the program did not finish within 4 hours.

We evaluate the inference performance by comparing with the DREAM5 gold standard, which represents a subset of the gene regulatory network. The gold standard is treated as an unweighted directed support and the inferred continuous edge weights are used directly as ranking scores. All predicted edges are ranked by their inferred weights, and following the DREAM5 protocol, the top 10^5 links are used to compute the AUC. To assess the robustness of the graph topology learned, we report the $\mathsf{Rob}(\boldsymbol{S})$ values for each learned graph topology.

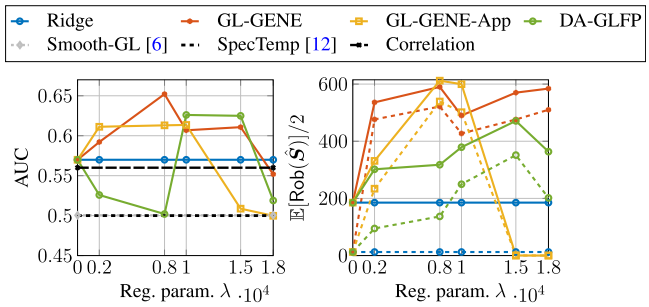


Fig. 8. Performance of algorithms on DREAM5. Left: AUC. Right: Expected resilience metric $\mathbb{E}[\mathrm{Rob}(\hat{\mathbf{S}})]$ across λ. Dashed line represents the perturbation set with $p = 0.1$.

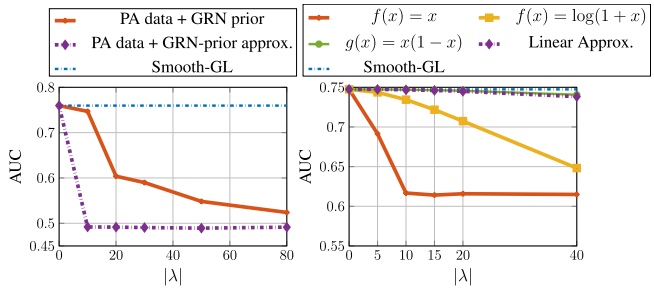


Fig. 9. Prior-misspecification controls on the PA-graph experiment. Left: Replacing the network-game welfare prior with the GRN resilience prior. Right: Randomizing the node-level benefit vector $\boldsymbol{b}$.

In addition, we simulate $n_{\mathsf{per}} = 50$ perturbations by randomly deleting edges from the learned graph with probability $p = 0.1$, forming a perturbed set $\hat{\mathcal{S}}$.

Fig. 8 presents the performance of the algorithms as λ increases. The regularization parameter $\hat{\lambda}$ in (GL-GENE-App) is scaled relative to that in (GL-GENE), with $\hat{\lambda} = \lambda/240000$. Observe that the proposed GL-GENE method and its approximation outperform the vanilla ridge regression in terms of AUC for a wide range of λ. Among all methods, (GL-GENE) yields the best trade-off between AUC and resilience. However, the resilience of the topology learned by the approximate method (GL-GENE-App) varies across runs. In comparison, although DA-GLFP performs slightly worse than (GL-GENE), its resilience is more consistent than that of (GL-GENE-App). This is consistent with the model mismatch explanation above: Off-the-shelf baselines designed for smooth graph signals or network-game-induced functional priors are not well aligned with perturbation-response GRN data, so their performance is substantially worse.

C. Sensitivity to Prior Misspecification

We examine the sensitivity of GLFP to misspecification of the functional prior. Fig. 9 reports two control experiments that retain the same PA-graph data-fitting setup used above. In the left panel, we replace the network-game welfare prior with the GRN resilience prior. In the right panel, we replace the node-level benefit vector $\boldsymbol{b}$ with i.i.d. random values in $(0, 1)$. In both cases, increasing $|\lambda|$ weakens or reverses the AUC gain. These results illustrate that the functional prior acts as a task-aligned inductive bias whose benefit depends on its alignment with the underlying network.

VII. Conclusion

We introduce an equilibrium-constrained bilevel-type framework for graph learning with functional priors. We show that the squared projected-gradient residuals of the iterates generated by the TTGD algorithm converge at a sublinear rate under strong-monotonicity and regularity conditions. Our experiments indicate improved support identification when the prior is aligned with the formation mechanism, while the added diagnostics show that weight recovery and AUC need not move together, and that misspecified or excessive regularization can be harmful. Topics for future research include multiple-equilibrium theory, consistency guarantees, and scalable asymmetric linear solvers.

Appendix A
Proofs of Structural Interpretations

A. Proof of Proposition 3

As $\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})$ is in the interior of $\mathcal{Y}$, it is also the solution to $\boldsymbol{y} = \widetilde{\mathsf{T}}(\boldsymbol{y}; \boldsymbol{S})$. Here, $\widetilde{\mathsf{T}}_i(\boldsymbol{y}; \boldsymbol{S}) := \arg\max_{y_i} U_i(y_i, \boldsymbol{y}_{-i}; \boldsymbol{S})$ is the unconstrained best response map. Consequently, the Lipschitzness and positivity of f and g imply that $0 \le \boldsymbol{y}^{\mathsf{NE}} \le \ell_1 \boldsymbol{S}\boldsymbol{y}^{\mathsf{NE}} + \boldsymbol{b}$ for any $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$. Since $\boldsymbol{S} \ge 0$ with row sums $\boldsymbol{S}1 = c1$, we have $\|\boldsymbol{S}\|_\infty = c$. Taking the ∞-norm on both sides of the preceding componentwise inequality gives

$$\|\boldsymbol{y}^{\mathsf{NE}}\|_\infty \le c\ell_1 \|\boldsymbol{y}^{\mathsf{NE}}\|_\infty + \|\boldsymbol{b}\|_\infty,$$

and hence $\|\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})\|_\infty \le \|\boldsymbol{b}\|_\infty/(1 - c\ell_1)$. Using $\|\cdot\|_2 \le \sqrt{N}\|\cdot\|_\infty$, we obtain

$$\|\boldsymbol{y}^{\mathsf{NE}}(\boldsymbol{S})\|_2 \le \frac{\sqrt{N}\,\|\boldsymbol{b}\|_\infty}{1 - c\ell_1}.$$

Since $c\ell_1 < 1$, iterating the componentwise bound $\boldsymbol{y}^{\mathsf{NE}} \le \ell_1 \boldsymbol{S}\boldsymbol{y}^{\mathsf{NE}} + \boldsymbol{b}$ and using $(\boldsymbol{S}^k \boldsymbol{b})_i \le c^k \|\boldsymbol{b}\|_\infty$ for $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$, we have

$$\begin{aligned}\boldsymbol{y}^\star &\le (\boldsymbol{I} + \ell_1 \boldsymbol{S})\boldsymbol{b} + \|\boldsymbol{b}\|_\infty \left((c\ell_1)^2 + (c\ell_1)^3 + \cdots\right) 1 \\ &= (\boldsymbol{I} + \ell_1 \boldsymbol{S})\boldsymbol{b} + (1 - c\ell_1)^{-1}(c\ell_1)^2 \|\boldsymbol{b}\|_\infty 1.\end{aligned}$$

Hence, for any $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$,

$$\begin{aligned}&\mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}) + \beta\|\boldsymbol{S}\|_F^2 - \lambda 1^\top \boldsymbol{y}^\star(\boldsymbol{S}) \\ &\ge \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}) + \beta\|\boldsymbol{S}\|_F^2 - \lambda\ell_1 1^\top \boldsymbol{S}\boldsymbol{b} - \lambda 1^\top \bar{\boldsymbol{c}},\end{aligned}$$

where $\bar{\boldsymbol{c}} = \frac{(c l_1)^2}{1 - c l_1}\|\boldsymbol{b}\|_\infty \mathbf{1} + \boldsymbol{b}$. This proves the first part of (21).

On the other hand, there exists a $\mu_1 \le \ell_1$ such that for any $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$, we have $\boldsymbol{y}^\star \ge \mu_1 \boldsymbol{S}\boldsymbol{y}^\star + \boldsymbol{b}$. Applying recursion gives

$$\begin{aligned}\boldsymbol{y}^\star &\ge (\boldsymbol{I} + \mu_1 \boldsymbol{S})\boldsymbol{b} + b^\star \left((c\mu_1)^2 + (c\mu_1)^3 + \cdots\right) 1 \\ &= (\boldsymbol{I} + \mu_1 \boldsymbol{S})\boldsymbol{b} + (1 - c\mu_1)^{-1}(c\mu_1)^2 b^\star 1.\end{aligned}$$

Consequently, for any $\boldsymbol{S} \in \mathcal{S}_{\mathsf{ng}}$,

$$\begin{aligned}&\mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}) + \beta\|\boldsymbol{S}\|_F^2 - \lambda 1^\top \boldsymbol{y}^\star(\boldsymbol{S}) \\ &\le \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}) + \beta\|\boldsymbol{S}\|_F^2 - \lambda\mu_1 1^\top \boldsymbol{S}\boldsymbol{b} - \lambda 1^\top \hat{\boldsymbol{c}},\end{aligned}$$

with $\hat{\boldsymbol{c}} = \frac{(c\mu_1)^2}{1 - c\mu_1} b^\star \mathbf{1} + \boldsymbol{b}$. This proves the second part of (21).

B. Proof of Proposition 4

Introducing the dual variables $\lambda \in \mathbb{R}, \boldsymbol{\eta} \in \mathbb{R}^N, \boldsymbol{h} \in \mathbb{R}^N, \boldsymbol{\mu} \in \mathbb{R}_+^{N\times N}$, the Lagrangian of (GL-NG-App) is

$$\mathcal{L}(\boldsymbol{S}, \lambda, \boldsymbol{\eta}, \boldsymbol{\mu}, \boldsymbol{h}) = \mathrm{Tr}(\boldsymbol{S}^\top \boldsymbol{D}) + \beta\|\boldsymbol{S}\|_F^2 - \tilde{\lambda}1^\top \boldsymbol{S}\boldsymbol{b} - \boldsymbol{\eta}^\top(\boldsymbol{S}1 - c1) - \langle \boldsymbol{\mu}, \boldsymbol{S}\rangle - \sum_{i=1}^N h_i \boldsymbol{e}_i^\top \boldsymbol{S}\boldsymbol{e}_i.$$

The first order necessary condition of (GL-NG-App) yields

$$\boldsymbol{D} + 2\beta\boldsymbol{S} - \tilde{\lambda}1\boldsymbol{b}^\top - \boldsymbol{\mu} - \boldsymbol{\eta}1^\top - \mathrm{Diag}(\boldsymbol{h}) = 0,$$

while the complementary slackness condition yields $S_{ij}\mu_{ij} = 0$. Together with $\boldsymbol{S} \geq 0$, for $i \neq j$, we have two cases:

Case 1. If $S_{ij} > 0$, then we must have $\mu_{ij} = 0$ and thus

$$S_{ij} = (2\beta)^{-1}(-D_{ij} + \tilde{\lambda}b_j + \eta_i).$$

Case 2. If $S_{ij} = 0$, then with $\mu_{ij} \geq 0$, we get

$$-D_{ij} + \tilde{\lambda}b_j + \eta_i \leq 0.$$

Examining the two cases yields the stated expression for $S_{ij}^\star$.

C. Proof of Proposition 5

Observe that $y_i = \sum_{j=1, j\neq i}^N S_{ij}\frac{y_j^2}{y_j^2+1}$ for any $\boldsymbol{S} \in \mathcal{S}_{\mathsf{nd}}$ and $i \in V$. We first notice the elementary inequality $(1+e^{-\sigma y})^{-1} \leq (1/4)\sigma y + (1/2)$, which holds for any $y \geq 0$. As $\boldsymbol{y}$ satisfies the constraints in (GL-GENE), we have

$$\begin{aligned}\Phi(\boldsymbol{S}; \boldsymbol{y}) &= \|\boldsymbol{S}\boldsymbol{X} + \boldsymbol{P}\|_F^2 + \beta\|\boldsymbol{S}\|_F^2 - \lambda 1^\top \frac{1}{1+e^{-\sigma \boldsymbol{y}}} \\ &\geq \|\boldsymbol{S}\boldsymbol{X} + \boldsymbol{P}\|_F^2 + \beta\|\boldsymbol{S}\|_F^2 - \sigma\lambda 1^\top \boldsymbol{y}/4 - N\lambda/2.\end{aligned}$$

For $y \geq 0$, as $\frac{y^2}{1+y^2} \leq \frac{1}{2}y$, we can infer from (14) that

$$\begin{aligned}y_i &= \sum\nolimits_{j\neq i} S_{ij}\frac{y_j^2}{y_j^2+1} \leq \left(\frac{\boldsymbol{S}\boldsymbol{y}}{2}\right)_i \\ &= \sum\nolimits_{j\neq i} (\boldsymbol{S}^2)_{ij}\frac{y_j^2/2}{y_j^2+1} \leq \left(\frac{\boldsymbol{S}^2 1}{2}\right)_i.\end{aligned}$$

Combining the above with the displayed lower bound proves the statement.

D. Proof of Proposition 6

For any $\boldsymbol{S} \in \mathcal{S}_{\mathsf{nd}}$, define

$$R_i := \sum_{j=1}^N S_{ij}, \qquad C_i := \sum_{j=1}^N S_{ji}, \qquad i \in [N].$$

Then, $\sum_{i=1}^N R_i = \sum_{i=1}^N C_i = a$, and we have

$$1^\top \boldsymbol{S}^2 1 = \sum_{i,j,\ell} S_{i\ell}S_{\ell j} = \sum_{\ell=1}^N \Big(\sum_{i=1}^N S_{i\ell}\Big)\Big(\sum_{j=1}^N S_{\ell j}\Big) = \sum_{\ell=1}^N C_\ell R_\ell.$$

Moreover, since $\mathrm{diag}(\boldsymbol{S}) = 0$ and $\boldsymbol{S} \geq 0$, we get

$$R_\ell + C_\ell = \sum_{j\neq\ell} S_{\ell j} + \sum_{i\neq\ell} S_{i\ell} \leq \sum_{i,j} S_{ij} = a, \qquad \forall\, \ell \in [N].$$

This yields

$$C_\ell R_\ell \leq \frac{(C_\ell + R_\ell)^2}{4} \leq \frac{a(C_\ell + R_\ell)}{4}, \qquad \forall\, \ell \in [N]. \tag{23}$$

Summing over ℓ gives

$$1^\top \boldsymbol{S}^2 1 = \sum_{\ell=1}^N C_\ell R_\ell \leq \frac{a}{4}\sum_{\ell=1}^N (C_\ell + R_\ell) = \frac{a^2}{2}. \tag{24}$$

It follows that every feasible $\boldsymbol{S}$ satisfies $1^\top \boldsymbol{S}^2 1 \leq a^2/2$. Equality holds when

$$S_{i^\star j^\star} = S_{j^\star i^\star} = \frac{a}{2}, \qquad S_{ij} = 0 \text{ otherwise}, \qquad i^\star \neq j^\star,$$

so the maximizer $\boldsymbol{S}^\star$ has value $a^2/2$. Furthermore, equality in (24) implies equalities in (23) for each $\ell \in [N]$, i.e.,

$$C_\ell R_\ell = \frac{(C_\ell + R_\ell)^2}{4} = \frac{a(C_\ell + R_\ell)}{4}, \qquad \forall\, \ell \in [N].$$

Therefore, for each $\ell \in [N]$, we have

$$C_\ell = R_\ell = 0 \qquad \text{or} \qquad C_\ell = R_\ell = \frac{a}{2}.$$

Let $T := \{\ell \in [N] : R_\ell = a/2\}$. Since $\sum_{\ell=1}^N R_\ell = a$ and $R_\ell \in \{0, a/2\}$ for $\ell \in [N]$, we have $|T| = 2$. Hence, we may write $T = \{i^\star, j^\star\}$ with $i^\star \neq j^\star$. For $\ell \notin T$, both the ℓ-th row and ℓ-th column of $\boldsymbol{S}^\star$ vanish. Hence, the only possible nonzero entries are $S_{i^\star j^\star}^\star$ and $S_{j^\star i^\star}^\star$, and the row-sum constraints give

$$S_{i^\star j^\star}^\star = S_{j^\star i^\star}^\star = \frac{a}{2}.$$

All other entries are zero. This proves the claim.

REFERENCES

[1] M. Newman, *Networks.* Oxford, U.K.: Oxford Univ. Press, 2018.

[2] G. Mateos, S. Segarra, A. G. Marques, and A. Ribeiro, "Connecting the dots: Identifying network structure via graph signal processing," *IEEE Signal. Process. Mag.*, vol. 36, no. 3, pp. 16–43, May 2019.

[3] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data: A signal representation perspective," *IEEE Signal. Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.

[4] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical Lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[5] X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, "Learning Laplacian matrix in smooth graph signal representations," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6160–6173, Dec. 2016.

[6] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. AISTATS*, 2016, pp. 920–929.

[7] P.-L. Loh and M. J. Wainwright, "Structure estimation for discrete graphical models: Generalized covariance matrices and their inverses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 2087–2095.

[8] H. Rue and L. Held, *Gaussian Markov Random Fields: Theory and Applications.* Boca Raton, FL, USA: Chapman and Hall/CRC, 2005.

[9] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the Lasso," *Ann. Statist.*, vol. 34, no. 3, pp. 1436–1462, 2006.

[10] Y. Shen, B. Baingana, and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2503–2516, May 2017.

[11] H.-T. Wai, A. Scaglione, B. Barzel, and A. Leshem, "Joint network topology and dynamics recovery from perturbed stationary points," *IEEE Trans. Signal Process.*, vol. 67, no. 17, pp. 4582–4596, Sep. 2019.

[12] S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network topology inference from spectral templates," *IEEE Trans. Signal Inform. Process. Netw.*, vol. 3, no. 3, pp. 467–483, Sep. 2017.

[13] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE Trans. Signal Process.*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[14] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal. Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[15] Z. Chen, W. Gan, J. Wu, K. Hu, and H. Lin, "Data scarcity in recommendation systems: A survey," *ACM Trans. Recommender Syst.*, vol. 3, no. 3, pp. 1–31, 2025.

[16] Y. Li, M. Purcell, T. Rakotoarivelo, D. Smith, T. Ranbaduge, and K. S. Ng, "Private graph data release: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, pp. 1–39, 2023.

[17] G. Kossinets, "Effects of missing data in social networks," *Social Netw.*, vol. 28, no. 3, pp. 247–268, 2006.

[18] X. Yang, M. Sheng, Y. Yuan, and T. Q. Quek, "Network topology inference from heterogeneous incomplete graph signals," *IEEE Trans. Signal Process.*, vol. 69, pp. 314–327, 2020.

[19] M. O. Jackson, *Social and Economic Networks.* Princeton, NJ, USA: Princeton Univ. Press, 2008.

[20] S. M. Smith et al., "Network modelling methods for FMRI," *NeuroImage*, vol. 54, no. 2, pp. 875–891, 2011.

[21] F. R. Chung, *Spectral Graph Theory.* Providence, RI, USA: AMS, 1997.

[22] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proc. CIKM*, 2001, pp. 25–32.

[23] L. Lovász, *Graphs and Geometry*, vol. 65. Providence, RI, USA: AMS, 2019.

[24] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under Laplacian and structural constraints," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 6, pp. 825–841, Sep. 2017.

[25] S. Kumar, J. Ying, J. V. d M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *J. Mach. Learn. Res.*, vol. 21, no. 22, pp. 1–60, 2020.

[26] F. Nie, X. Wang, M. Jordan, and H. Huang, "The constrained Laplacian rank algorithm for graph-based clustering," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016, pp. 1969–1976.

[27] G. Demange, "Optimal targeting strategies in a network under complementarities," *Games Econ. Behav.*, vol. 105, pp. 84–103, Sep. 2017.

[28] O. Candogan, K. Bimpikis, and A. Ozdaglar, "Optimal pricing in networks with externalities," *Oper. Res.*, vol. 60, no. 4, pp. 883–905, 2012.

[29] J. Gao, B. Barzel, and A.-L. Barabási, "Universal resilience patterns in complex networks," *Nature*, vol. 530, no. 7590, pp. 307–312, 2016.

[30] M. Hong, H.-T. Wai, Z. Wang, and Z. Yang, "A two-timescale stochastic algorithm framework for bilevel optimization: Complexity analysis and application to actor-critic," *SIAM J. Optim.*, vol. 33, no. 1, pp. 147–180, 2023.

[31] H. Shen and T. Chen, "On penalty-based bilevel gradient descent method," in *Proc. ICML*, 2023, pp. 30992–31015.

[32] A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots, "Truncated back-propagation for bilevel optimization," in *Proc. AISTATS*, 2019, pp. 1723–1732.

[33] Z. Zhou and A. M.-C. So, "A unified approach to error bounds for structured convex optimization problems," *Math. Program.*, vol. 165, pp. 689–728, 2017.

[34] M. Kühne, P. D. Grontas, G. De Pasquale, G. Belgioioso, F. Dorfler, and J. Lygeros, "Optimizing social network interventions via hypergradient-based recommender system design," in *Proc. Int. Conf. Mach. Learn.*, PMLR, 2025, pp. 31860–31875.

[35] Z. Xu and V. Tzoumas, "Performance-aware self-configurable multi-agent networks: A distributed submodular approach for simultaneous coordination and network design," 2024, *arXiv:2409.01411*.

[36] M. Hu, H. Chang, B. Ma, and S. Shan, "Learning continuous graph structure with bilevel programming for graph neural networks," in *Proc. IJCAI*, 2022, pp. 3057–3063.

[37] C. Zhang, S. Liu, H.-T. Wai, and A. M.-C. So, "Network games induced prior for graph topology learning," in *Proc. ICASSP*, 2025, pp. 1–5.

[38] R. Ramakrishna, H. T. Wai, and A. Scaglione, "A user guide to low-pass graph signal processing and its applications: Tools and applications," *IEEE Signal. Process. Mag.*, vol. 37, no. 6, pp. 74–85, Nov. 2020.

[39] A. Tjärnberg, T. E. Nordling, M. Studham, and E. L. Sonnhammer, "Optimal sparsity criteria for network inference," *J. Comput. Biol.*, vol. 20, no. 5, pp. 398–408, 2013.

[40] T. Hillerton, D. Seçilmiş, S. Nelander, and E. L. Sonnhammer, "Fast and accurate gene regulatory network inference by normalized least squares regression," *Bioinformatics*, vol. 38, no. 8, pp. 2263–2268, 2022.

[41] Y.-J. Chen, Y. Zenou, and J. Zhou, "The impact of network topology and market structure on pricing," *J. Econ. Theory*, vol. 204, 2022, Art. no. 105491.

[42] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proc. Nat. Acad. Sci.*, vol. 101, no. 14, pp. 4781–4786, 2004.

[43] S. Ciliberti, O. C. Martin, and A. Wagner, "Robustness can evolve gradually in complex regulatory gene networks with varying topology," *PLoS Comput. Biol.*, vol. 3, no. 2, 2007, Art. no. e15.

[44] J. Cai, C. Zhang, and H.-T. Wai, "Optimal pricing for linear-quadratic games with nonlinear intesraction between agents," *IEEE Control Syst. Lett.*, vol. 8, pp. 1631–1636, 2024.

[45] C. Ballester, A. Calvó-Armengol, and Y. Zenou, "Who's who in networks. Wanted: The key player," *Econometrica*, vol. 74, no. 5, pp. 1403–1417, 2006.

[46] Y. Bramoullé, R. Kranton, and M. D'amours, "Strategic interaction and networks," *Amer. Econ. Rev.*, vol. 104, pp. 898–930, 2014.

[47] M. O. Jackson and Y. Zenou, "Games on Networks," in *Handbook of Game Theory with Economic Applications*, Amsterdam, The Netherlands: Elsevier, 2015, vol. 4, pp. 95–163.

[48] N. Allouch, "On the private provision of public goods on networks," *J. Econ. Theory*, vol. 157, pp. 527–552, 2015.

[49] F. Parise and A. Ozdaglar, "A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis," *Games Econ. Behav.*, vol. 114, pp. 47–82, 2019.

[50] M. Belhaj and F. Deroïan, "Competing activities in social networks," *BE J. Econ. Anal. & Policy*, vol. 14, no. 4, pp. 1431–1466, 2014.

[51] H. Sanhedrai, J. Gao, A. Bashan, M. Schwartz, S. Havlin, and B. Barzel, "Reviving a failed network through microscopic interventions," *Nat. Phys.*, vol. 18, no. 3, pp. 338–349, 2022.

[52] C. Meena, C. Hens, S. Acharyya, S. Haber, S. Boccaletti, and B. Barzel, "Emergent stability in complex network dynamics," *Nat. Phys.*, vol. 19, no. 7, pp. 1033–1042, 2023.

[53] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits.* Boca Raton, FL, USA: Chapman and Hall/CRC, 2019.

[54] G. Karlebach and R. Shamir, "Modelling and analysis of gene regulatory networks," *Nat. Rev. Mol. Cell Biol.*, vol. 9, no. 10, pp. 770–780, 2008.

[55] T. S. Gardner, D. Di Bernardo, D. Lorenz, and J. J. Collins, "Inferring genetic networks and identifying compound mode of action via expression profiling," *Science*, vol. 301, no. 5629, pp. 102–105, 2003.

[56] Y. Yuan, G.-B. Stan, S. Warnick, and J. Goncalves, "Robust dynamical network structure reconstruction," *Automatica*, vol. 47, no. 6, pp. 1230–1235, Jun. 2011.

[57] B. Liu et al., "Inducing equilibria via incentives: Simultaneous design-and-play ensures global convergence," in *Proc. NeurIPS*, vol. 35, 2022, pp. 29001–29013.

[58] Y. Zhang et al., "Advancing model pruning via bi-level optimization," in *Proc. NeurIPS*, vol. 35, 2022, pp. 18309–18326.

[59] D. Davis and D. Drusvyatskiy, "Stochastic model-based minimization of weakly convex functions," *SIAM J. Optim.*, vol. 29, no. 1, pp. 207–239, 2019.

[60] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317, New York, NY, USA: Springer, 2009.

[61] K. Yosida, *Functional Analysis*, vol. 123. New York, NY, USA: Springer, 2012.

[62] Y. Leng, X. Dong, J. Wu, and A. Pentland, "Learning quadratic games on networks," in *Proc. ICML*, 2020, pp. 5820–5830.

[63] A. Banerjee, A. G. Chandrasekhar, E. Duflo, and M. O. Jackson, "The diffusion of microfinance," *Science*, vol. 341, no. 6144, 2013, Art. no. 1236498.

[64] S. Gama-Castro et al., "RegulonDB (version 6.0): Gene regulation model of *Escherichia coli* K-12 beyond transcription, active (experimental) annotated promoters and Textpresso navigation," *Nucleic Acids Res.*, vol. 36, no. suppl_1, pp. D120–D124, 2008.

[65] S. Balaji, M. M. Babu, L. M. Iyer, N. M. Luscombe, and L. Aravind, "Comprehensive analysis of combinatorial regulation using the transcriptional regulatory network of yeast," *J. Mol. Biol.*, vol. 360, no. 1, pp. 213–227, 2006.

[66] D. Marbach et al., "Wisdom of crowds for robust gene network inference," *Nat. Methods*, vol. 9, no. 8, pp. 796–804, 2012.