Bilevel Network Learning via Hierarchically Structured Sparsity

Jiayi Fan[1]*, Jingyuan Yang[1]*, Shuangge Ma[2], Mengyun Wu[1]†
[1]School of Statistics and Data Science, Shanghai University of Finance and Economics
[2]Department of Biostatistics, Yale School of Public Health
fanjiayi@stu.sufe.edu.cn, yang.jingyuan@stu.sufe.edu.cn,
shuangge.ma@yale.edu, wu.mengyun@mail.shufe.edu.cn

Abstract

Accurate network estimation serves as the cornerstone for understanding complex systems across scientific domains, from decoding gene regulatory networks in systems biology to identifying social relationship patterns in computational sociology. Modern applications demand methods that simultaneously address two critical challenges: capturing nonlinear dependencies between variables and reconstructing inherent hierarchical structures where higher-level entities coordinate lower-level components (e.g., functional pathways organizing gene clusters). Traditional Gaussian graphical models fundamentally fail in these aspects due to their restrictive linear assumptions and flat network representations. We propose NNBLNet, a neural network-based learning framework for bi-level network inference. The core innovation lies in hierarchical selection layers that enforce structural consistency between high-level coordinator groups and their constituent low-level connections via adaptive sparsity constraints. This architecture is integrated with a compositional neural network architecture that learn cross-level association patterns through constrained nonlinear transformations, explicitly preserving hierarchical dependencies while overcoming the representational limitations of linear methods. Crucially, we establish formal theoretical guarantees for the consistent recovery of both high-level connections and their internal low-level structures under general statistical regimes. Extensive validation demonstrates NNBLNet's effectiveness across synthetic and real-world scenarios, achieving superior F1 scores compared to competitive methods and particularly beneficial for complex systems analysis through its interpretable bi-level structure discovery.

1 Introduction

Network estimation is a fundamental task across many disciplines—such as genetics, finance, and social science—where uncovering the structure of dependencies among variables can yield critical insights into the underlying mechanisms of complex systems [8, 38, 44]. For instance, in genetics, gene regulatory networks can be reconstructed from analyzing multi-patient omics datasets to identify statistically significant edges between genes. In social science, co-authorship networks connect researchers through jointly published papers, with edge weights reflecting collaboration intensity, and the resulting network reveals meaningful patterns of scholarly connection and knowledge flow.

A key characteristic shared across these domains is that variables can often be naturally organized into groups. Such group information is readily available as direct labels in practice: for instance, most genes have well-annotated pathway information in databases like KEGG, which naturally defines

*These authors contributed equally.
†Corresponding author. Email: wu.mengyun@mail.shufe.edu.cn

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

functional groupings, while public social datasets commonly include demographic or professional categories such as gender or research field that provide meaningful group divisions. Under such a grouped organization, connections may occur both within these groups—reflecting functional or structural coherence—and between groups—reflecting higher-order dependencies or cross-functional regulation [26, 6]. Crucially, these systems exhibit hierarchical dependency architectures: dependencies among low-level variables often emerge only when their parent groups share systemic interdependencies. For instance, in genomics, genes within a pathway rarely interact with genes in unrelated pathways unless cross-pathway regulatory mechanisms exist [14, 10]. This hierarchical structure poses significant modeling challenges, as conventional network estimation methods that ignore group-level dependencies often fail to capture emergent system behaviors, yielding fragmentary results that lack systematic interpretability. This motivates bi-level network estimation, which aims to recover hierarchically structured dependencies at both the group and variable levels.

Gaussian Graphical Models (GGMs) are widely used for learning network structures by estimating the inverse covariance matrix under multivariate Gaussian assumptions due to their ability to capture conditional dependencies between variables [39]. Furthermore, a series of methods reformulate the GGM estimation problem into a set of sparse linear regression models, often offering greater computational efficiency [24]. Extensions to bi-level GGMs have also been proposed to capture grouped variable associations [29]. However, GGMs are restricted to modeling linear relationships and rely heavily on Gaussianity, limiting their applicability in real-world data with nonlinear or non-Gaussian structure. To address this gap, a number of model-free or nonparametric approaches have been proposed [13, 36, 32]. While these methods alleviate distributional assumptions, they often struggle to model highly complex interdependencies in modern high-dimensional systems, where traditional kernel-based or graphical techniques lack the representational capacity to capture intricate hierarchical patterns. Neural network (NN) has demonstrated superior performance in capturing nonlinear relationships compared to conventional nonparametric methods [18]. Nevertheless, neural network-based bi-level network estimation remains unexplored.

We propose a neural network–based framework for bi-level network estimation to address the challenge of modeling complex, hierarchical dependencies in grouped variables. This work pioneers the use of neural networks for bi-level network inference, effectively capturing nonlinear interdependencies both within and across groups, beyond the capacity of traditional linear models. The key contributions of our work are:

- Bi-level Network Estimation: We propose a structured estimation framework that recovers hierarchical network architectures through dual-layer selection mechanisms, capturing bi-level dependencies between variables by incorporating group information and identifying group dependencies simultaneously.
- Hierarchical Nonlinear Architecture: We integrate compositional neural network architectures to model cross-level dependencies via constrained nonlinear transformations, explicitly preserving hierarchical structures while overcoming the representational constraints of linear methods.
- Theoretical Guarantees: We formally introduce the notion of *bi-level selection consistency* and establish a rigorous selection consistency result under high-dimensional regimes, ensuring the method to reliably recover the true network structure at both levels.
- Empirical Validation: We validate our method through comprehensive experiments on both synthetic and real-world datasets. The results highlight the advantages of our approach in accurately estimating complex network structures, especially in scenarios where traditional GGM-based methods fail due to nonlinearity or distributional misspecification.

1.1 Notation

Consider n independent observations with p variables partitioned into L predefined groups (Figure 1 A), where the l-th group contains p_l variables such that $\sum_{l=1}^{L} p_l = p$. For the i-th observation, define the continuous measurement vector as $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ip})^\top$. The group membership of the j-th variable is specified by $C_j \in \{1, \ldots, L\}$. We further define two derived quantities: $\boldsymbol{x}_{i,-j} = (x_{i,1}, \ldots, x_{i,j-1}, x_{i,j+1}, \ldots, x_{i,p})^\top$ representing the i-th observation with the j-th variable excluded and $\tilde{\boldsymbol{x}}_{il}$ denoting the subvector containing all variables from group l in the i-th observation. We define the operator $\odot$ as a block-wise product between a vector and

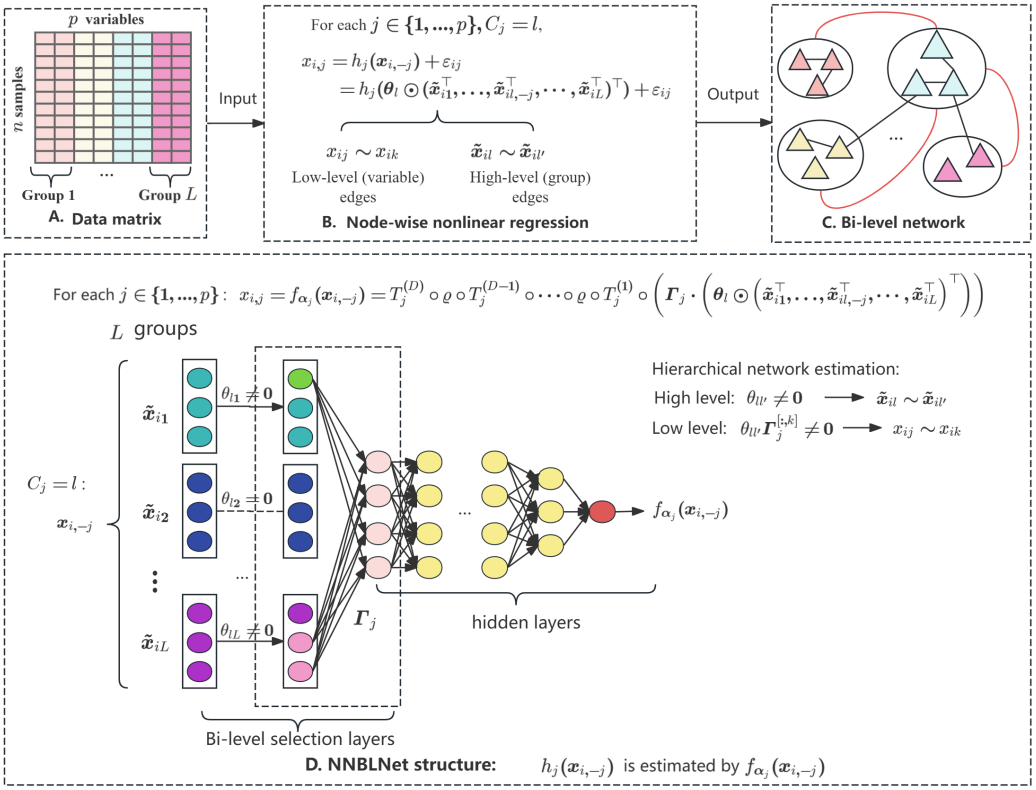

Figure 1: Workflow of the proposed bi-level network estimation. (A) Input $\boldsymbol{x}_i = (x_{i,1}, \cdots, x_{i,p})^\top$'s: n observations with p variables partitioned into L groups. (B) Node-wise nonlinear regression for modeling the relationships between the jth variable $x_{i,j}$ and the others $\boldsymbol{x}_{i,-j}$, as well as those between groups $\tilde{\boldsymbol{x}}_{i,l}$ and $\tilde{\boldsymbol{x}}_{i,l'}$. Here, $C_j = l$ indicates that the jth variable belongs to the lth group. (C) Output: Bi-level network, including both low-level edges and high-level edges. (D) Network structure of the proposed NNBLNet, including the two hierarchical selection layers and multiple hidden layers for accommodating the nonlinear relationships among both low-level variables and high-level groups.

a partitioned vector (or matrix). Specifically, let $\boldsymbol{A} = (a_1, \ldots, a_m)^\top \in \mathbb{R}^{m \times 1}$ be a vector and $\boldsymbol{B} = \left(\left(\boldsymbol{B}^{(1)} \right)^\top, \ldots, \left(\boldsymbol{B}^{(m)} \right)^\top \right)^\top \in \mathbb{R}^{n \times N}$ be a matrix that can be decomposed into m contiguous blocks, i.e., $\boldsymbol{B}^{(l)} \in \mathbb{R}^{n_l \times N}$ for $l = 1, \ldots, m$ with $\sum_{l=1}^m n_l = n$. The block-wise product $\boldsymbol{A} \odot \boldsymbol{B} \in \mathbb{R}^{n \times N}$ is defined as $\boldsymbol{A} \odot \boldsymbol{B} := \left(a_1 \left(\boldsymbol{B}^{(1)} \right)^\top, a_2 \left(\boldsymbol{B}^{(2)} \right)^\top, \ldots, a_m \left(\boldsymbol{B}^{(m)} \right)^\top \right)^\top$. That is, each element a_l scales all entries in the corresponding block $\boldsymbol{B}^{(l)}$.

2 Neural Network-based Bi-level Network Estimation (NNBLNet)

Traditional methods, such as GGMs, typically handle linear conditional dependencies captured via precision matrices. However, realistic systems often exhibit complex, nonlinear associations that linear models may not adequately capture. To overcome these limitations, our framework formalizes nonlinear relationships among variables through node-wise nonlinear regression. Specifically, for the j-th variable with group assignment $C_j = l$, consider

$$x_{i,j} = h_j(\boldsymbol{x}_{i,-j}) + \varepsilon_{ij} \triangleq h_j \left(\left(\tilde{\boldsymbol{x}}_{i1}^\top, \ldots, \tilde{\boldsymbol{x}}_{il,-j}^\top, \cdots, \tilde{\boldsymbol{x}}_{iL}^\top \right)^\top \right) + \varepsilon_{ij}, \tag{1}$$

where $\tilde{\boldsymbol{x}}_{il,-j}$ specifically excludes the j-th variable from its native group l, ε_{ij} is independent and identically distributed sub-Gaussian stochastic noise with mean zero and sub-Gaussian parameter σ,

and $h_j : \mathbb{R}^{p-1} \to \mathbb{R}$ is a sparse nonparametric mapping (Figure 1 B). Model (1) captures the nonlinear relationships between $x_{i,j}$ and all other variables $\boldsymbol{x}_{i,-j}$, enabling more accurate characterization of complex system mechanisms than linear approximations permit. The group information is incorporated in (1) through structured group inputs $\tilde{\boldsymbol{x}}_{il}$. This strategy can also accommodate nonlinear high-level relationships between group l and all other groups, leading to deeper investigation of the underlying structures (Figure 1 C).

For estimating $h_j(\cdot)$ and performing bi-level network estimation, we propose a neural network with two hierarchical selection layers and D hidden layers (Figure 1 D):

$$f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j}) = T_j^{(D)} \circ \varrho \circ T_j^{(D-1)} \circ \cdots \circ \varrho \circ T_j^{(1)} \circ \left(\boldsymbol{\Gamma}_j \cdot \left(\boldsymbol{\theta}_l \odot \left(\tilde{\boldsymbol{x}}_{i1}^\top, \ldots, \tilde{\boldsymbol{x}}_{il,-j}^\top, \cdots, \tilde{\boldsymbol{x}}_{iL}^\top \right)^\top \right) \right), \quad (2)$$

with $\boldsymbol{\alpha}_j = \left\{ \boldsymbol{\theta}_l, \boldsymbol{\Gamma}_j, \{\boldsymbol{\Delta}_j^{(d)}, \boldsymbol{b}_j^{(d)}\}_{d=1}^D \right\}$ being the parameter set. Here, each affine transformation is given by $T_j^{(d)}(\boldsymbol{u}) = \boldsymbol{\Delta}_j^{(d)} \boldsymbol{u} + \boldsymbol{b}_j^{(d)}$ with learnable parameters $\boldsymbol{\Delta}_j^{(d)} \in \mathbb{R}^{w_j^{(d)} \times w_j^{(d-1)}}$ and $\boldsymbol{b}_j^{(d)} \in \mathbb{R}^{w_j^{(d)} \times 1}$, where $w_j^{(d)}$ is the width of the d-th hidden layer. The activation function ϱ is chosen as the ReLU function and $\odot$ is the block-wise product defined in *Notation*.

In (2), we innovatively introduce two selection layers. Specifically, the first selection layer involves the vector $\boldsymbol{\theta}_l = (\theta_{l1}, \ldots, \theta_{ll}, \ldots, \theta_{lL})^\top$, which encodes group-level (high-level) dependencies with $\theta_{ll} = 1$. Simultaneously, the second selection layer involves the matrix $\boldsymbol{\Gamma}_j \in \mathbb{R}^{w_j^{(1)} \times (p-1)}$, which captures specific associations between variable j and the other $p-1$ variables. Hence, by integrating these hierarchical components, the relationships among low-level variables are explicitly characterized by $\boldsymbol{\theta}_l \odot \boldsymbol{\Gamma}_j^\top$, where high-level group dependencies act as latent scaffolds that constrain or enable low-level dependencies.

Denote $\boldsymbol{\alpha} = \left\{ \boldsymbol{\theta}, \left\{ \boldsymbol{\Gamma}_j, \{\boldsymbol{\Delta}_j^{(d)}, \boldsymbol{b}_j^{(d)}\}_{d=1}^D \right\}_{j=1}^p \right\}$ with $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_L)$, and $\boldsymbol{\Gamma}_j^{[:,k]}$ as the k-th column of $\boldsymbol{\Gamma}_j$. Integrating $f_{\boldsymbol{\alpha}_1}(\boldsymbol{x}_{i,-1}), \cdots, f_{\boldsymbol{\alpha}_p}(\boldsymbol{x}_{i,-p})$, we propose a dual-penalized estimator designed to achieve bi-level sparsity for network estimation:

$$\hat{\boldsymbol{\alpha}}_n = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n l(\boldsymbol{\alpha}_j, x_{i,j}, \boldsymbol{x}_{i,-j}) + \lambda_1 \sum_{l<l'} |\theta_{ll'}| + \lambda_2 \sum_{j=1}^p \sum_{k=1}^{p-1} \left\| \boldsymbol{\Gamma}_j^{[:,k]} \right\|, \quad (3)$$

with $l(\boldsymbol{\alpha}_j, x_{i,j}, \boldsymbol{x}_{i,-j}) = \left(x_{i,j} - f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j}) \right)^2$ and $\|\cdot\|$ being the L_2-norm of a vector.

Here, the first term constitutes a quadratic reconstruction loss. The second term implements a Lasso penalty on $\theta_{ll'}$, inducing sparsity through element-wise shrinkage towards zero, which enables automatic high-level edge selection. This foundational assumption of inter-group sparsity aligns with established practices in prior methodological work [5, 29]. This is well illustrated in the context of genomic networks, where pathway-to-pathway connections exhibit natural sparsity since regulatory relationships occur only between specific pathway pairs rather than universally [27]. The third term employs a group Lasso penalty on $\boldsymbol{\Gamma}_j^{[:,k]}$, enforcing simultaneous shrinkage of entire parameter vectors to zero, thereby facilitating low-level edge selection. The tuning parameters λ_1 and λ_2 govern edge selection stringency, with larger values yielding sparser network structures.

Based on $\hat{\boldsymbol{\alpha}}_n$, to mitigate regularization bias and enhance selection consistency, we further introduce an adaptive bi-level sparse estimator:

$$\tilde{\boldsymbol{\alpha}}_n = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \sum_{j=1}^p \frac{1}{n} \sum_{i=1}^n l(\boldsymbol{\alpha}_j, x_{i,j}, \boldsymbol{x}_{i,-j}) + \zeta_1 \sum_{l<l'} \frac{|\theta_{ll'}|}{|\hat{\theta}_{ll'}|^\gamma} + \zeta_2 \sum_{j=1}^p \sum_{k=1}^{p-1} \frac{\left\| \boldsymbol{\Gamma}_j^{[:,k]} \right\|}{\left\| \hat{\boldsymbol{\Gamma}}_j^{[:,k]} \right\|^\gamma}, \quad (4)$$

with ζ_1 and ζ_2 being two tuning parameters for controlling network sparsity and γ being a positive constant. Here, the weights $|\hat{\theta}_{ll'}|^\gamma$ and $\left\| \hat{\boldsymbol{\Gamma}}_j^{[:,k]} \right\|^\gamma$ are introduced to adaptively reduce penalties for connections with larger initial estimates while amplifying the shrinkage for smaller ones, resulting in more accurate edge selection owing to reduced estimation bias.

The final bi-level network is constructed based on $\tilde{\boldsymbol{\alpha}}_n$. Specifically, high-level connections between groups l and l' are established when either $\tilde{\theta}_{ll'} \neq 0$ or $\tilde{\theta}_{l'l} \neq 0$. This high-level connectivity then

informs low-level interactions: an edge forms between node j in group l and node k in group l' if either $\tilde{\theta}_{ll'}\tilde{\mathbf{\Gamma}}_j^{[:,k]} \neq \mathbf{0}$ or $\tilde{\theta}_{l'l}\tilde{\mathbf{\Gamma}}_k^{[:,j]} \neq \mathbf{0}$. The proposed bi-level network estimation strategy can also be extended to accommodate overlapping groups, with the details provided in the Appendix.

3 Statistical Properties

3.1 Approximation Error

Without loss of generality, assume that $\boldsymbol{x}_i \in \mathcal{X} \subset [0,1]^p$. Define the β-Hölder smooth class: $\mathcal{H}^\beta([0,1]^s, B_0) = \left\{ f : [0,1]^s \to \mathbb{R}, \max_{\|\alpha\|_1 \leq \lfloor\beta\rfloor} \|\partial^\alpha f\|_\infty \leq B_0, \max_{\|\alpha\|_1 = \lfloor\beta\rfloor} \sup_{x \neq y} \frac{|\partial^\alpha f(x) - \partial^\alpha f(y)|}{\|x-y\|_2^{\beta - \lfloor\beta\rfloor}} \leq B_0 \right\}$, where $\partial^\alpha = \partial^{\alpha_1} \cdots \partial^{\alpha_s}$ with $\alpha = (\alpha_1, \ldots, \alpha_s)^\top \in \mathbb{N}_0^s$ and $\mathbb{N}_0$ denotes the set of non-negative integers, $\|\alpha\|_1 = \sum_{i=1}^s |\alpha_i|$ and $\lfloor\beta\rfloor$ denotes the largest integer strictly smaller than β. If a function belongs to $\mathcal{H}^\beta([0,1]^s, B_0)$, then all the partial derivatives up to order $\lfloor\beta\rfloor$ exist, and the partial derivatives of order $\lfloor\beta\rfloor$ are $\beta - \lfloor\beta\rfloor$ Hölder continuous.

Asummption 3.1. *For $j = 1, \cdots, p$, the target sparse function h_j resides in the s_j-sparse function class $\mathcal{F}_{s_j}$, where $\mathcal{F}_s = \left\{ h : [0,1]^{p-1} \to [0,1] : \exists\, \bar{h} : [0,1]^s \to [0,1] \in \mathcal{H}^\beta([0,1]^s, B_0) \right.$ s.t. $h(\boldsymbol{x}) = \bar{h}(\tilde{\boldsymbol{x}}), \forall \boldsymbol{x} \in [0,1]^{p-1} \left. \right\}$, with $\tilde{\boldsymbol{x}} \in \mathbb{R}^s$ representing the relevant s-dimensional subvector of $\boldsymbol{x}$.*

Assumption 3.1 posits that the target functions exhibit sparsity and smoothness. This is a common requirement in neural network approximation theory [15, 4]. In addition, the assumption is grounded in reality: for instance, in gene regulatory networks, genes are sparsely rather than fully connected, and nonlinear effects vary smoothly rather than abruptly.

Next, we first establish the approximation capabilities of our proposed sparse ReLU feedforward neural network architecture for sparse nonlinear functions. We define the width of a neural network as the maximum width among its hidden layers.

Theorem 3.2 (Approximation Error)**.** *If $h(\boldsymbol{x}) \in \mathcal{F}_s$, then for any positive integer N and M, there exists a sparse ReLU feedforward neural network $f(\boldsymbol{x})$ with width $W = 38(\lfloor\beta\rfloor + 1)^2 3^s s^{\lfloor\beta\rfloor+1} N \lceil\log_2(8N)\rceil$, depth $D = 21(\lfloor\beta\rfloor + 1)^2 M \lceil\log_2(8M)\rceil + 2s$, and the selection layer parameter $(\boldsymbol{\theta} \odot \mathbf{\Gamma}^\top)^\top$ being able to be rearranged as $[\boldsymbol{u}^{w^{(1)} \times s}, \mathbf{0}^{w^{(1)} \times (p-1-s)}]$ such that:*

1. *Support equivalence: both $h(\boldsymbol{x})$ and $f(\boldsymbol{x})$ are s-sparse functions and have the same support;*
2. *Uniform approximation:*

$$\sup_{\|\boldsymbol{x}\| \leq [0,1]^{p-1}} |f(\boldsymbol{x}) - h(\boldsymbol{x})| \leq 19 B_0 (\lfloor\beta\rfloor + 1)^2 s^{\lfloor\beta\rfloor + (\beta \vee 1)/2} (NM)^{-2\beta/s}. \tag{5}$$

Under Assumption 3.1, Theorem 3.2 indicates that sparser nonlinear functions require smaller network dimensions for accurate approximation—an observation aligning with intuition. Therefore, the target sparse function h_j can be approximated by a sparse ReLU feedforward network with parameter set $\boldsymbol{\alpha}_j^*$, denoted as $f_{\boldsymbol{\alpha}_j^*}$.

3.2 Bi-level Selection Consistency

Let $c_j : \mathbb{N} \to \mathbb{N}$ be the column label mapping where $c_j(k)$ denotes the column of $\mathbf{\Gamma}_j$ corresponding to the k-th variable, that is, $c_j(k) = k$ if $k < j$ and $c_j(k) = k - 1$ if $k > j$. Denote $[p]$ and $[L]$ as the set $\{1, \cdots, p\}$ and $\{1, \cdots, L\}$, respectively, and $\mathcal{A}^c$ as the complement of $\mathcal{A}$. In addition, we formalize three fundamental sets: $\mathcal{A}_j = \left\{ k \in [p] : \theta^*_{C_j, C_k} \neq 0 \text{ and } \mathbf{\Gamma}_j^{*[:, c_j(k)]} \neq 0 \right\}$, $\mathcal{B}_j = \left\{ k \in [p] : \mathbf{\Gamma}_j^{*[:, c_j(k)]} = 0 \right\}$, and $\mathcal{P}_l = \left\{ l' \in [L] : \theta^*_{l,l'} = 0 \right\}$.

Then, we can define the bi-level selection consistency as follows.

Definition 3.3 (Bi-level Selection Consistency)**.** *We say the estimator $(\widetilde{\boldsymbol{\theta}}, \{\widetilde{\mathbf{\Gamma}}_j\}_{j=1}^p)$ achieves bi-level selection consistency if $\forall \delta > 0$, $\exists N_\delta \in \mathbb{N}$ such that $\forall n > N_\delta$:*

1. ***High-level true positives**: $\widetilde{\theta}_{ll'} \neq 0$ for all $(l, l') \in [L] \times \mathcal{P}_l^c$;*

2. ***High-level true negatives**: $\widetilde{\theta}_{ll'} = 0$ for all $(l, l') \in [L] \times \mathcal{P}_l$;*

3. ***Low-level true positives**: $\left(\left(\widetilde{\boldsymbol{\theta}}_{C_j} \odot \widetilde{\boldsymbol{\Gamma}}_j^\top\right)^\top\right)^{[:, c_j(k)]} \neq 0$ for all $(j, k) \in [p] \times \mathcal{A}_j$;*

4. ***Low-level true negatives**: $\left(\left(\widetilde{\boldsymbol{\theta}}_{C_j} \odot \widetilde{\boldsymbol{\Gamma}}_j^\top\right)^\top\right)^{[:, c_j(k)]} = 0$ for all $(j, k) \in [p] \times \mathcal{A}_j^c$;*

with probability at least $1 - \delta$.

Let $\boldsymbol{\alpha}^*$ consist of all $\boldsymbol{\alpha}_1^*, \cdots, \boldsymbol{\alpha}_p^*$ and $\mathcal{W}$ denote the feasible parameter space with $\boldsymbol{\alpha}^* \in \mathcal{W}$. We define the population risk $R(\boldsymbol{\alpha})$ and empirical risk $R_n(\boldsymbol{\alpha})$ associated with the squared error as:

$$R(\boldsymbol{\alpha}) = \sum_{j=1}^{p} R_j(\boldsymbol{\alpha}_j) = \sum_{j=1}^{p} \mathbb{E}\left[\left(f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{-j}) - x_j\right)^2\right], \tag{6}$$

$$R_n(\boldsymbol{\alpha}) = \sum_{j=1}^{p} R_{nj}(\boldsymbol{\alpha}_j) = \frac{1}{n} \sum_{j=1}^{p} \sum_{i=1}^{n} \left(f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j}) - x_{ij}\right)^2. \tag{7}$$

The optimal parameter set is defined as:

$$\mathcal{H}^* = \{\boldsymbol{\alpha} \in \mathcal{W} : R(\boldsymbol{\alpha}) = R(\boldsymbol{\alpha}^*)\}. \tag{8}$$

Despite the intricate geometric structure of $\mathcal{H}^*$, we establish in Lemma A.1 (Appendix) the fundamental equivalence:

$$\boldsymbol{\alpha}_0 \in \mathcal{H}^* \iff f_{\boldsymbol{\alpha}_{0j}} = f_{\boldsymbol{\alpha}_j^*}, \quad \forall j \in \{1, \ldots, p\}. \tag{9}$$

This equivalence implies that both the estimation and selection consistency can be simultaneously attained by controlling the proximity of the parameters to the optimal set $\mathcal{H}^*$. Define $d(\boldsymbol{\alpha}, \mathcal{H}^*) = \inf_{\boldsymbol{\beta} \in \mathcal{H}^*} \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|$.

To establish these theoretical guarantees, we introduce the following assumptions.

Asummption 3.4. *Define $\mathcal{F}$ as a class of ReLU feedforward neural networks $f_{\check{\boldsymbol{\alpha}}} : [0, 1]^{p-1} \to [0, 1]$ with parameter $\check{\boldsymbol{\alpha}}$, depth D, width W, size S (the number of elements in $\check{\boldsymbol{\alpha}}$) and B-Lipschitz continuity. We assume that $f_{\boldsymbol{\alpha}_j} \in \mathcal{F}$ for all $j = 1, 2, \ldots, p$.*

Asummption 3.5. *There exist $c_2 > 0$ and $\nu > 2$ such that $R(\boldsymbol{\beta}) - R(\boldsymbol{\alpha}^*) \geq c_2 d(\boldsymbol{\beta}, \mathcal{H}^*)^\nu$ for all $\boldsymbol{\beta} \in \mathcal{W}$.*

Assumption 3.4 posits the boundedness of the ReLU neural network function class. This assumption holds in practice, as real-world data (e.g., gene expression levels) and weights are naturally bounded by physical constraints. Assumption 3.5 is a technical assumption. For a fixed network with an analytic activation function, it holds and can be justified by Lojasiewicz's inequality [9].

Under Assumptions 3.4 and 3.5, the following convergence properties hold:

Theorem 3.6 (Group Lasso + Lasso Convergence)**.** *Let $p = o(\log n), SD\log(S) = O(n^{\frac{1}{4}}), \lambda_1 = O(n^{-\frac{1}{8}})$, and $\lambda_2 = O(n^{-\frac{1}{8}})$, then there exist $c_3 > 0$ and $c_4 > 0$ such that*

$$d(\hat{\boldsymbol{\alpha}}_n, \mathcal{H}^*) \leq c_3 \left(\frac{\log n}{n^{\frac{1}{8}}}\right)^{\frac{1}{\nu - 1}} \tag{10}$$

and

$$\sum_{j=1}^{p} \sum_{k \in \mathcal{B}_j} \left\|\hat{\boldsymbol{\Gamma}}_j^{[:, c_j(k)]}\right\| + \sum_{l=1}^{L} \sum_{l' \in \mathcal{P}_l} |\hat{\theta}_{l,l'}| \leq c_4 \log n \left(\frac{\log n}{n^{\frac{1}{8}}}\right)^{\frac{1}{\nu - 1}} \tag{11}$$

holds with probability at least $1 - \delta_1$ with $\delta_1 = 4n\left(n^{\frac{1}{4}} + 1\right)^{\log n}\left(32en^{\frac{1}{4}}\right)^{(\log n)n^{\frac{1}{4}}} e^{-\frac{\sqrt{n}\log n}{32}}$.

Theorem 3.7 (Consistency of the adaptive bi-level sparse estimator)**.** *Let* $\gamma > 0, \epsilon > 0, \lambda_1 = O(n^{-\frac{1}{8}}), \lambda_2 = O(n^{-\frac{1}{8}}), \zeta_1 = O\left(n^{-\frac{\gamma}{8(\nu-1)}+\epsilon}\right)$*, and* $\zeta_2 = O\left(n^{-\frac{\gamma}{8(\nu-1)}+\epsilon}\right)$*, then the estimator* $\tilde{\boldsymbol{\alpha}}_n$ *with adaptive bi-level sparse penalty has bi-level selection consistency, and there exists* $c_5 > 0$ *such that*

$$d(\tilde{\boldsymbol{\alpha}}_n, \mathcal{H}^*) \leq c_5 n^{\left(-\frac{\gamma}{8(\nu-1)}+\epsilon\right)/\nu}, \tag{12}$$

holds with probability at least $1 - \delta_1$.

Under Assumption 3.4 and Theorem 3.7, the estimation error can be bounded by leveraging the Lipschitz continuity $|f_{\tilde{\boldsymbol{\alpha}}_j} - f_{\boldsymbol{\alpha}_j^*}| \leq Bc_5 n^{\left(-\frac{\gamma}{8(\nu-1)}+\epsilon\right)/\nu}$. This result implies that the estimation error decays as sample size n increases. Combined Theorem 3.2 and Theorem 3.7, we can easily establish the estimation consistency of $f_{\tilde{\boldsymbol{\alpha}}_j}$ by triangle inequality. That is, $|f_{\tilde{\boldsymbol{\alpha}}_j} - h_j| \leq Bc_5 n^{\left(-\frac{\gamma}{8(\nu-1)}+\epsilon\right)/\nu} + 19B_0(\lfloor\beta\rfloor+1)^2 s^{\lfloor\beta\rfloor+(\beta\vee1)/2}(NM)^{-2\beta/s}$ with probability at least $1 - \delta_1$.

4 Computation

The proximal gradient descent algorithm for training NNBLNet is outlined in Algorithm 1 (see Appendix). Following the convergence conditions in Theorems 3.6 and 3.7, we set $\lambda_k = \zeta_k = c \cdot n^{-1/8}$ for $k = 1, 2$, with c as a tunable constant. This choice follows conventional rate settings in high-dimensional sparse inference [22]. Empirically, we recommend $c = 0.35$, as it achieves satisfactory accuracy across diverse data settings. For finer calibration, these parameters can be selected via cross-validation (see Appendix A.4). Consistent with common practice, we set $\gamma = 1$. Regarding neural network hyperparameters, exploratory experiments (Appendix A.4) showed that a configuration of 1000 training epochs, three hidden layers, and 50 nodes per layer offers an optimal balance between accuracy and computational efficiency. This fixed setup was used in all experiments to ensure reproducibility and consistency, in line with common practice [33].

5 Experiment

To comprehensively validate our methodology, we established a dual evaluation framework encompassing both synthetic benchmarks and real-world networks. The synthetic analysis included systematically constructed networks with predefined nonlinear and linear patterns, enabling a controlled assessment of relationship modeling capabilities. For real-world validation, we analyzed four distinct network types: (1) Friendship– social connections among high school students from a high school; (2) Co-authorship–collaborative relationships in academic publications; (3-4) BRCA and LUAD–gene regulatory networks derived from The Cancer Genome Atlas (TCGA) breast cancer and lung adenocarcinoma data, respectively.

In addition to the proposed NNBLNet, we conducted systematic comparisons with three alternative methodologies: (1) BGSL [7], a Bayesian Gaussian graphical modeling framework with explicit group-structured variable representations; (2) Fair Glasso [25], which enhances sparse precision matrix estimation through group-aware regularization constraints; and (3) DeepGRNCS [19], a multi-task deep learning architecture specializing in joint network inference. Experimental details, including the synthetic settings and introductions to the real-world datasets, are provided in Appendix A.5. To supplement the results presented in the main text, we also conducted a series of sensitivity analyses in Appendix A.6, including an assessment of model generalization capability based on the stability of network estimation and performance comparisons under different sample sizes, group sizes, and group label misclassification rates, as well as downstream analysis of the LUAD dataset.

5.1 Synthetic Networks with Nonlinear and Linear signals

In Table 1, NNBLNet consistently outperformed all baseline methods in nonlinear scenario, achieving the highest F1-score of 0.772. This improvement was attributed to its hierarchical design, which jointly modeled both within-group and between-group edges through bi-level regularization. Fair Glasso and BGSL, which impose sparsity within the GGM framework, yielded comparable results with F1-scores of 0.710 and 0.709, respectively. DeepGRNCS, a deep learning-based method, performed better than classical approaches but fell short of our proposed method, suggesting that

integrating bi-level selection is crucial for recovering multiscale network structure. These results demonstrate the efficacy of NNBLNet in uncovering meaningful networks with greater accuracy and structural coherence.

Under the linear setting, Fair Glasso attained the highest F1-score (0.790), aligning with its design premise for sparse Gaussian graphical models. NNBLNet remained highly competitive (F1 = 0.769), which illustrates its adaptability across both linear and nonlinear data regimes. Although specifically designed for hierarchical nonlinear dependencies, it maintained robust performance even in simpler settings. DeepGRNCS also delivered favorable results, while BGSL achieved a balanced recall-precision trade-off. These findings verify that NNBLNet not only performs stably in conventional linear estimation tasks but also offers flexibility to more complex environments.

Table 1: Performance comparison of different methods for the synthetic network with nonlinear and linear relationships: Mean (SD) over 100 replicates.

Patterns	**Metric**	**NNBLNet**	**Fair Glasso**	**BGSL**	**DeepGRNCS**
Nonlinear	Recall	0.872(0.016)	0.779(0.014)	0.790(0.017)	0.809(0.016)
	Precision	0.693(0.022)	0.653(0.021)	0.644(0.018)	0.681(0.018)
	F1-score	0.772(0.014)	0.710(0.017)	0.709(0.013)	0.731(0.017)
Linear	Recall	0.881(0.018)	0.901(0.016)	0.836(0.016)	0.846(0.016)
	Precision	0.675(0.020)	0.704(0.019)	0.692(0.019)	0.671(0.020)
	F1-score	0.765(0.013)	0.790(0.013)	0.757(0.015)	0.763(0.017)

5.2 Four Real-World Networks

We further evaluated our method on four real-world datasets spanning social networks and biological systems: Friendship, Co-authorship, BRCA, and LUAD. Performance results are summarized in Table 2.

Table 2: Performance comparison of different methods for the four real-world networks.

Dataset	**Metric**	**NNBLNet**	**Fair Glasso**	**BGSL**	**DeepGRNCS**
Friendship	Recall	0.875	0.859	0.798	0.811
	Precision	0.804	0.703	0.735	0.686
	F1	0.838	0.771	0.765	0.745
Co-authorship	Recall	0.712	0.643	0.622	0.655
	Precision	0.678	0.606	0.591	0.610
	F1	0.695	0.619	0.604	0.632
BRCA	Recall	0.764	0.812	0.676	0.742
	Precision	0.618	0.652	0.552	0.601
	F1	0.683	0.723	0.607	0.664
LUAD	Recall	0.641	0.625	0.597	0.654
	Precision	0.526	0.517	0.489	0.539
	F1	0.577	0.566	0.537	0.591

Across all four datasets, NNBLNet consistently achieved either the best or second-best F1-score, demonstrating its versatility and robustness in diverse application domains. On the Friendship network, it significantly outperformed all competitors with an F1-score of 0.838, reflecting its strength in modeling community-driven structures. In the Co-authorship setting, where overlapping communities and latent hierarchies were expected, NNBLNet again led with an F1-score of 0.695, significantly outperforming Fair Glasso (0.619), BGSL (0.604), and DeepGRNCS (0.632). These results indicate the advantage of jointly modeling inter-group and intra-group dependencies.

For the biological datasets, BRCA and LUAD, NNBLNet maintained top-tier performance. While Fair Glasso achieved a slightly higher F1 score in BRCA (0.723), NNBLNet offered more balanced precision-recall trade-offs, especially in LUAD, where it achieved the second highest F1-score

(0.577). The improvement in biological contexts suggests that our method effectively captures complex relationships between molecular factors, including both nonlinear individual factor-level dynamics and pathway-level crosstalk.

Collectively, these results demonstrate that NNBLNet is a highly competitive and generalizable framework for network estimation, performing consistently well across domains with varying complexity and noise characteristics.

5.3 Ablation Study

To validate the necessity of the proposed bi-level architecture, we performed ablation studies comparing our model against two simplified variants: the flat-structured NNNet, which ignores the bi-level design, and Modified-NNNet, which uses group labels but omits intra-group adjacency modeling. As shown in Table 3, NNBLNet consistently outperformed both variants on synthetic and real-world datasets. The significant performance gain confirms that the bi-level structure is essential for achieving higher recall, precision, and F1 scores.

To further isolate the benefits of the bi-level architecture beyond group information alone, we conducted sensitivity analyses in scenarios with extreme sparsity of inter-group edges (Table 4), where the ratio of inter-group edges to total edges (η) was set to 0%, 0.5%, and 1%. The results showed that NNBLNet consistently achieved the highest F1 by balancing recall and precision. In addition, the modified-NNNet failed to capture inter-group edges and exhibited declining recall as η increased, while the NNNet, which ignored inter-group sparsity, produced excessive false positives under sparse settings. These findings confirmed that the bi-level hierarchical sparsity structure is essential for accurate network inference.

Table 3: Performance comparison of NNBLNet, NNNet, and Modified-NNNet across six datasets (mean values and standard deviation for synthetic datasets).

Dataset	Method	Recall	Precision	F1
Nonlinear	NNBLNet	0.872 (0.016)	0.693 (0.022)	0.772 (0.014)
	NNNet	0.844 (0.017)	0.656 (0.019)	0.738 (0.014)
	Modified-NNNet	0.782 (0.020)	0.660 (0.021)	0.716 (0.017)
Linear	NNBLNet	0.881 (0.018)	0.675 (0.020)	0.769 (0.013)
	NNNet	0.857 (0.014)	0.628 (0.017)	0.727 (0.013)
	Modified-NNNet	0.802 (0.018)	0.627 (0.019)	0.704 (0.016)
Friendship	NNBLNet	0.875	0.804	0.838
	NNNet	0.842	0.745	0.790
	Modified-NNNet	0.774	0.734	0.753
Co-authorship	NNBLNet	0.712	0.678	0.695
	NNNet	0.674	0.637	0.649
	Modified-NNNet	0.612	0.636	0.624
BRCA	NNBLNet	0.764	0.618	0.683
	NNNet	0.709	0.563	0.627
	Modified-NNNet	0.655	0.566	0.607
LUAD	NNBLNet	0.641	0.526	0.577
	NNNet	0.613	0.502	0.552
	Modified-NNNet	0.552	0.503	0.527

6 Discussion

This work presents NNBLNet, a neural network framework for bi-level network inference. The method is built upon a key structural prior that represents dependency structures in complex systems as hierarchical. This hierarchical modeling is implemented through two mechanisms: intra-group information sharing, which amplifies weak signals via latent pooling within groups, and inter-group sparse transmission, which gates cross-group connections through switches ($\theta_{ll'}$) to suppress irrelevant noise. NNBLNet represents a paradigm shift in structured network inference by unifying neural network representation learning with hierarchical sparsity constraints. Its innovative hierarchical

Table 4: Performance of NNBLNet, NNNet, and Modified-NNNet under sparse inter-group edges (mean and standard deviation over 100 replicates).

Setting	η	Method	Recall	Precision	F1
Nonlinear	0%	NNBLNet	0.752 (0.018)	0.684 (0.020)	0.716 (0.017)
		NNNet	0.744 (0.019)	0.619 (0.023)	0.676 (0.018)
		Modified-NNNet	0.737 (0.018)	0.682 (0.020)	0.708 (0.016)
	0.5%	NNBLNet	0.760 (0.017)	0.696 (0.019)	0.727 (0.016)
		NNNet	0.752 (0.018)	0.616 (0.022)	0.677 (0.017)
		Modified-NNNet	0.717 (0.019)	0.673 (0.021)	0.694 (0.017)
	1%	NNBLNet	0.774 (0.016)	0.715 (0.018)	0.743 (0.014)
		NNNet	0.764 (0.018)	0.615 (0.023)	0.681 (0.019)
		Modified-NNNet	0.696 (0.020)	0.663 (0.022)	0.679 (0.019)
Linear	0%	NNBLNet	0.740 (0.019)	0.672 (0.021)	0.704 (0.018)
		NNNet	0.732 (0.020)	0.607 (0.024)	0.664 (0.019)
		Modified-NNNet	0.724 (0.019)	0.671 (0.021)	0.696 (0.017)
	0.5%	NNBLNet	0.750 (0.018)	0.684 (0.020)	0.715 (0.017)
		NNNet	0.741 (0.019)	0.605 (0.023)	0.666 (0.018)
		Modified-NNNet	0.706 (0.020)	0.662 (0.021)	0.682 (0.018)
	1%	NNBLNet	0.766 (0.017)	0.702 (0.019)	0.733 (0.015)
		NNNet	0.753 (0.019)	0.605 (0.024)	0.671 (0.019)
		Modified-NNNet	0.681 (0.021)	0.652 (0.022)	0.666 (0.020)

selection layer explicitly captures bi-level dependencies: local associations between individual variables and global coordination among groups. This architecture effectively models complex nonlinear relationships while maintaining interpretability. Theoretically, we establish a bridge between neural modeling and statistical guarantees by proving estimation consistency and exact bi-level selection consistency. Empirical evaluations and real-data analyses demonstrate NNBLNet's effectiveness, showing it achieves superior F1 scores compared to competing methods.

6.1 Limitation

This study has several limitations that point to valuable directions for future research. Theoretically, our analysis relies on the assumption of sub-Gaussian noise, which, while common in statistical literature, is often difficult to verify in practical applications. Despite this limitation, empirical comparisons against authoritative ground-truth networks demonstrate that our method still achieves competitive network reconstruction accuracy, thereby offering partial validation of its practical effectiveness. Moreover, our method relies on pre-defined group labels, and performance may decline if these are noisy or incomplete. Future extensions could jointly infer group memberships and associations or incorporate unsupervised techniques (e.g., spectral clustering) when prior labels are unavailable, enhancing robustness and applicability across diverse scenarios.

6.2 Broader Impact

NNBLNet bridges neural network learning and hierarchical sparse inference to enable interpretable bi-level network discovery across scientific domains, from identifying gene-pathway interplay in disease mechanisms to modeling individual-group dynamics in social systems. Its adaptive sparsity design balances predictive power with mechanistic interpretability, offering actionable insights for precision medicine and policy-making.

Acknowledgments and Disclosure of Funding

We thank the Area Chair and the anonymous reviewers for their insightful feedback, which was instrumental in strengthening this paper. This research was supported by the National Natural Science Foundation of China (12071273); MOE Project of Humanities and Social Sciences (25YJCZH291); Shanghai Rising-Star Program (22QA1403500); Shanghai Science and Technology Development

Funds (23JC1402100); Shanghai Research Center for Data Science and Decision Technology; National Institutes of Health (CA204120); and National Science Foundation (2209685).

References

[1] M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

[2] P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019.

[3] T. Chen, L. M. Ashwood, O. Kondrashova, A. Strasser, G. Kelly, and K. D. Sutherland. Breathing new insights into the role of mutant p53 in lung cancer. *Oncogene*, 44(3):115–129, 2025.

[4] Y. Chen, Q. Gao, F. Liang, and X. Wang. Nonlinear variable selection via deep neural networks. *Journal of Computational and Graphical Statistics*, 30(2):484–492, 2021.

[5] L. Cheng, L. Shan, and I. Kim. Multilevel gaussian graphical model for multilevel networks. *Journal of Statistical Planning and Inference*, 190:1–14, 2017.

[6] I. Choi and W. C. Kim. A multifaceted graph-wise network analysis of sector-based financial instruments' price-based discrepancies with diverse statistical interdependencies. *The North American Journal of Economics and Finance*, 75:102316, 2025.

[7] A. Colombi, R. Argiento, L. Paci, and A. Pini. Learning block structured graphs in gaussian graphical models. *Journal of Computational and Graphical Statistics*, 33(1):152–165, 2024.

[8] L. Cowen, T. Ideker, B. J. Raphael, and R. Sharan. Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics*, 18(9):551–562, 2017.

[9] V. C. Dinh and L. S. Ho. Consistent feature selection for analytic deep neural networks. *Advances in Neural Information Processing Systems*, 33:2420–2431, 2020.

[10] R. Feng, J. Gong, L. Wu, L. Wang, B. Zhang, G. Liang, H. Zheng, and H. Xiao. Mapk and hippo signaling pathways crosstalk via the raf-1/mst-2 interaction in malignant melanoma. *Oncology Reports*, 38(2):1199–1205, 2017.

[11] A. Halama, S. Zaghlool, G. Thareja, S. Kader, W. Al Muftah, M. Mook-Kanamori, H. Sarwath, Y. A. Mohamoud, N. Stephan, S. Ameling, et al. A roadmap to the molecular human linking multiomics with population traits and diabetes subtypes. *Nature Communications*, 15(1):7111, 2024.

[12] A. Herreros-Pomares, J. D. de Maya-Girones, S. Calabuig-Fariñas, R. Lucas, A. Martínez, J. M. Pardo-Sánchez, S. Alonso, A. Blasco, R. Guijarro, M. Martorell, et al. Lung tumorspheres reveal cancer stem cell-like properties and a score with prognostic impact in resected non-small-cell lung cancer. *Cell Death & Disease*, 10(9):660, 2019.

[13] V. A. Huynh-Thu, A. Irrthum, L. Wehenkel, and P. Geurts. Inferring regulatory networks from expression data using tree-based methods. *PloS One*, 5(9):e12776, 2010.

[14] J.-W. Jang, M.-K. Kim, and S.-C. Bae. Reciprocal regulation of yap/taz by the hippo pathway and the small gtpase pathway. *Small GTPases*, 11(4):280–288, 2020.

[15] Y. Jiao, G. Shen, Y. Lin, and J. Huang. Deep nonparametric regression on approximate manifolds: Nonasymptotic error bounds with polynomial prefactors. *The Annals of Statistics*, 51(2):691–716, 2023.

[16] M. Kanehisa, M. Furumichi, Y. Sato, Y. Matsuura, and M. Ishiguro-Watanabe. Kegg: biological systems database as a model of the real world. *Nucleic Acids Research*, 53(D1):D672–D677, 2025.

[17] N. M. LaMarche, S. Hegde, M. D. Park, B. B. Maier, L. Troncoso, J. Le Berichel, P. Hamon, M. Belabed, R. Mattiuz, C. Hennequin, et al. An il-4 signalling axis in bone marrow drives pro-tumorigenic myelopoiesis. *Nature*, 625(7993):166–174, 2024.

[18] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[19] Y. Lei, X.-T. Huang, X. Guo, K. Hang Katie Chan, and L. Gao. Deepgrncs: deep learning-based framework for jointly inferring gene regulatory networks across cell subpopulations. *Briefings in Bioinformatics*, 25(4):bbae334, 2024.

[20] A. Lewkowycz and G. Gur-Ari. On the training dynamics of deep networks with l_2 regularization. *Advances in Neural Information Processing Systems*, 33:4790–4799, 2020.

[21] B. Liu, Y. Wei, Y. Zhang, and Q. Yang. Deep neural networks for high dimension, low sample size data. In *IJCAI*, volume 2017, pages 2287–2293, 2017.

[22] H. Liu and L. Wang. TIGER: A tuning-insensitive approach for optimally estimating Gaussian graphical models. *Electronic Journal of Statistics*, 11(1):241 – 294, 2017.

[23] G. Mao, Z. Pang, K. Zuo, Q. Wang, X. Pei, X. Chen, and J. Liu. Predicting gene regulatory links from single-cell rna-seq data using graph neural networks. *Briefings in Bioinformatics*, 24(6):bbad414, 2023.

[24] N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, 34(3):1436 – 1462, 2006.

[25] M. Navarro, S. Rey, A. Buciulea, A. G. Marques, and S. Segarra. Fair glasso: Estimating fair graphical models with unbiased statistical behavior. *Advances in Neural Information Processing Systems*, 2024.

[26] M. A. Reyna, D. Haan, M. Paczkowska, L. P. Verbeke, M. Vazquez, A. Kahraman, S. Pulido-Tamayo, J. Barenboim, L. Wadi, P. Dhingra, et al. Pathway and network analysis of more than 2500 whole cancer genomes. *Nature Communications*, 11(1):729, 2020.

[27] F. Sanchez-Vega, M. Mina, J. Armenia, W. K. Chatila, A. Luna, K. C. La, S. Dimitriadoy, D. L. Liu, H. S. Kantheti, S. Saghafinia, et al. Oncogenic signaling pathways in the cancer genome atlas. *Cell*, 173(2):321–337, 2018.

[28] L. Seninge, I. Anastopoulos, H. Ding, and J. Stuart. Vega is an interpretable generative model for inferring biological network activity in single-cell transcriptomics. *Nature Communications*, 12(1):5684, 2021.

[29] L. Shan, Z. Qiao, L. Cheng, and I. Kim. Joint estimation of the two-level gaussian graphical models across multiple classes. *Journal of Computational and Graphical Statistics*, 29(3):562–579, 2020.

[30] B. Shekar and G. Dagnew. L1-regulated feature selection and classification of microarray cancer data using deep learning. In *Proceedings of 3rd International Conference on Computer Vision and Image Processing: CVIP 2018, Volume 2*, pages 227–242. Springer, 2020.

[31] D. Stewart, M. Javadi, M. Chambers, C. Gunsolly, G. Gorski, and R. C. Borghaei. Interleukin-4 inhibition of interleukin-1-induced expression of matrix metalloproteinase-3 (mmp-3) is independent of lipoxygenase and pparγ activation in human gingival fibroblasts. *BMC Molecular Biology*, 8(1):12, 2007.

[32] C. Su, Z. Xu, X. Shan, B. Cai, H. Zhao, and J. Zhang. Cell-type-specific co-expression inference from single cell rna-sequencing data. *Nature Communications*, 14(1):4846, 2023.

[33] T. Sun, Y. Wei, W. Chen, and Y. Ding. Genome-wide association study-based deep learning for survival prediction. *Statistics in Medicine*, 39(30):4605–4620, 2020.

[34] Y. Sun, Q. Song, and F. Liang. Consistent sparse deep learning: Theory and computation. *Journal of the American Statistical Association*, 117(540):1981–1995, 2022.

[35] D. Szklarczyk, K. Nastou, M. Koutrouli, R. Kirsch, F. Mehryary, R. Hachilif, D. Hu, M. E. Peluso, Q. Huang, T. Fang, et al. The string database in 2025: protein networks with directionality of regulation. *Nucleic Acids Research*, 53(D1):D730–D737, 2025.

[36] X. Wang, D. Choi, and K. Roeder. Constructing local cell-specific networks from single-cell data. *Proceedings of the National Academy of Sciences*, 118(51):e2113178118, 2021.

[37] Y. Wang, T. Ran, Y. Li, L. Tian, L. Yang, Z. Liu, and B. Yao. Identification of jun gene and cellular microenvironment in response to pd-1 blockade treatment in lung cancer patients via single-cell rna sequencing. *Aging (Albany NY)*, 16(12):10348, 2024.

[38] R. Xu, C. Mi, R. Mierzwiak, and R. Meng. Complex network construction of internet finance risk. *Physica A: Statistical Mechanics and its Applications*, 540:122930, 2020.

[39] M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

[40] M. Zhang, Y. Sun, and F. Liang. Sparse deep learning for time series data: theory and applications. *Advances in Neural Information Processing Systems*, 36:54509–54539, 2023.

[41] T. Zhang and H. Zou. Sparse precision matrix estimation via lasso penalized D-trace loss. *Biometrika*, 101(1):103–120, 2014.

[42] Z. Zhang, H. Li, S. Jiang, R. Li, W. Li, H. Chen, and X. Bo. A survey and evaluation of web-based tools/databases for variant analysis of tcga data. *Briefings in Bioinformatics*, 20(4):1524–1541, 2019.

[43] Y. Zhou, X. Wang, H. Li, M. Fan, M. Wang, W. Gao, Z. Shi, J. Wang, J. Tan, and F. Liu. A positive feedback loop between serpinh1 and mmp-9/tgf-β1 promotes lung adenocarcinoma progression. *Cell Death & Differentiation*, pages 1–16, 2025.

[44] W. Zou, X. Hu, Z. Pan, C. Li, Y. Cai, and M. Liu. Exploring the relationship between social presence and learners' prestige in mooc discussion forums using automated content analysis and social network analysis. *Computers in Human Behavior*, 115:106582, 2021.

A Appendix / Supplemental Material

A.1 Related Works

In recent years, the explosive growth of high-dimensional data, such as high-throughput omics data, has generated unprecedented volumes of complex data, thereby elevating the importance of network estimation research. Among the various existing approaches, Gaussian Graphical Models (GGMs) have emerged as one of the most widely used techniques. In the GGM framework, network structures are inferred by estimating a sparse precision matrix (i.e., the inverse of the covariance matrix), which encodes conditional dependencies among variables conditional on all others [41, 11]. This estimation procedure can be formulated as a series of sparse node-wise linear regressions. Compared to marginal or unconditional correlation-based methods (e.g., Pearson correlation), this conditional strategy offers a more holistic view of system-level dependencies, potentially leading to more meaningful interpretations.

In real-world scenarios, networks often exhibit a bi-level hierarchical structure. This structure implies that some variables are organized into higher-level groups, with lower-level variables nested within these groups. For example, a gene pathway consists of multiple genes that collaborate to perform a specific cellular or physical function. In this context, pathways represent the higher-level groups, while the individual genes within those pathways are the lower-level variables. It is also important to note that these groups are not independent of each other. To tackle hierarchical network estimation, several GGM-based methods have been developed. Cheng et al. (2017) [5] introduced a multilevel Gaussian graphical model for nested data structures. Shan et al. (2020) [29] proposed a framework for joint estimation of two-level GGMs across multiple classes. Colombi et al. (2024) [7] focused on learning block-structured graphical models using variable groupings. Notably, Fair Glasso [25] specifically leverages group information to estimate graphical models with provably unbiased statistical behavior, addressing fairness concerns in network inference. However, these GGM based methods can only capture linear dependencies and may ignore complex nonlinear relationship in the real world.

Deep neural networks (DNNs) and related machine learning models have gained widespread attention due to their strong capacity for nonlinear approximation and representation learning, particularly in high-dimensional settings [21]. These models excel at uncovering complex associations within large datasets, making them particularly appealing for network inference tasks. With the increasing availability of large-scale high-dimensional data, deep learning has become a cornerstone for network estimation. Researchers have recently developed several deep learning-based methods that aim to reconstruct latent networks. These methods harness the expressive power of neural networks to model intricate dependencies between variables, often using architectures such as pre-trained deep neural networks [19], Variational Autoencoders (VAEs) [28] and Graph Convolutional Networks (GCNs) [23]. While promising, these approaches face several important limitations. First, most are supervised and require labeled data—such as group-specific regulatory annotations or curated databases—which are expensive and time-consuming to acquire. Consequently, the size and diversity of training datasets remain limited. Moreover, deep learning methods often lack theoretical guarantees and interpretability, which hampers their adoption in sensitive scientific domains.

To improve interpretability, the research community has proposed a variety of strategies aimed at making deep learning models more transparent and reliable. Among these, regularization-based approaches have shown particular promise. By incorporating additional constraints (e.g., L1 or L2 penalties), regularization can limit model complexity, promote sparsity, and enhance feature selection [30, 20]. This is especially valuable for network inference, where identifying the most influential variables (e.g., hub nodes) is often a primary goal. In scientific applications, regularization enhances both statistical reliability and practical interpretability. Recent theoretical work further supports the effectiveness of regularization for consistent variable selection in deep models [34, 40], providing a rigorous foundation for interpretable deep learning-based network estimation. These theoretical insights not only deepen our understanding of model behavior but also inform the development of robust and generalizable algorithms.

A.2 Extension to Overlapping Groups

In many scientific applications, variables may simultaneously belong to multiple groups, resulting in overlapping group structures. For example, in genomics, the same gene may participate in several

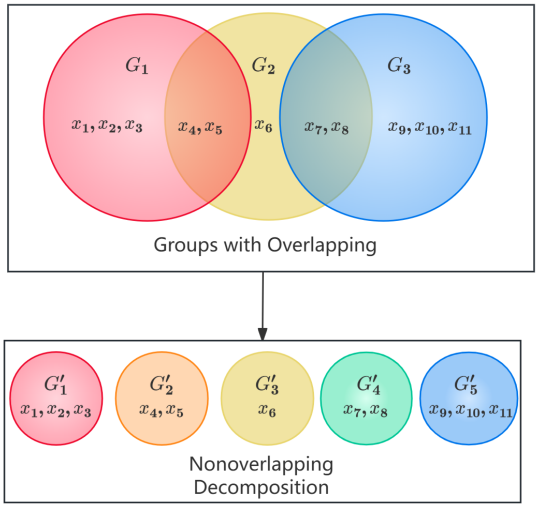

Figure 2: Illustration of overlapping and disjoint group structures. The upper panel shows three pathways G_1, G_2, G_3 with shared genes, leading to overlapping groups. The lower panel demonstrates the decomposition of these overlapping pathways into non-overlapping subgroups $(G'_1, \ldots, G'_5)$, which allows group-level dependencies to be defined in a consistent and interpretable manner.

biological pathways, so different pathways can share common genes. Such overlaps complicate the definition of group-level dependencies, as conventional methods typically assume disjoint group memberships.

To illustrate, consider three pathways denoted by G_1, G_2, and G_3 as shown in Figure 2. Each contains five genes: $G_1 = \{1, 2, 3, 4, 5\}, G_2 = \{4, 5, 6, 7, 8\}, G_3 = \{7, 8, 9, 10, 11\}$. Here, G_1 and G_2 overlap on genes $\{4, 5\}$, G_2 and G_3 overlap on $\{7, 8\}$, while G_1 and G_3 have no common elements. To resolve overlaps, we decompose the original groups into disjoint subgroups as shown in Figure 2: $G'_1 = \{1, 2, 3\}, G'_2 = \{4, 5\}, G'_3 = \{6\}, G'_4 = \{7, 8\}, G'_5 = \{9, 10, 11\}$. Accordingly, the original pathways can be represented as $G_1 = \{G'_1, G'_2\}, G_2 = \{G'_2, G'_3, G'_4\}, G_3 = \{G'_4, G'_5\}$. We denote $\theta'_{ll'} = 0$ whenever two disjoint subgroups G'_l and $G'_{l'}$ are conditionally independent. Based on this decomposition, we define the conditional dependency rules among the original pathways:

- Non-overlapping groups: For instance, G_1 and G_3 have no overlap. If $\theta'_{14} = \theta'_{15} = \theta'_{24} = \theta'_{25} = 0$, then G_1 and G_3 are conditionally independent, as all variables in G_1 are independent of those in G_3. Otherwise, they are conditionally dependent.
- Overlapping groups: For adjacent groups such as G_1 and G_2, if $\theta'_{13} \neq 0$ or $\theta'_{14} \neq 0$, they are conditionally dependent, since at least one unique variable in G_1 depends on a unique variable in G_2. If instead $\theta'_{13} = \theta'_{14} = 0$, but at least one of $\theta'_{12}, \theta'_{23}, \theta'_{24}$ is nonzero, dependency arises through the common subgroup. Otherwise, G_1 and G_2 are conditionally independent.

This decomposition preserves interpretability by mapping subgroup-level dependencies back to the original overlapping groups, while avoiding inflated false discoveries due to redundant memberships.

A.3 Proof of Statistical Properties

A.3.1 Proof of Theorem 3.2

Since $h(\boldsymbol{x}) \in \mathcal{F}_s$, there exists a $\bar{h}(\tilde{\boldsymbol{x}}) \in \mathcal{H}^\beta([0,1]^s, B_0)$ such that $h(\boldsymbol{x}) = \bar{h}(\tilde{\boldsymbol{x}})$. Refer to Corollary 3.1 in [15], there exists a function $\bar{f}(\tilde{\boldsymbol{x}})$ implemented by a ReLU network with width $W = 38(\lfloor\beta\rfloor + 1)^2 3^s s^{\lfloor\beta\rfloor+1} N \lceil\log_2(8N)\rceil$ and depth $D = 21(\lfloor\beta\rfloor + 1)^2 M \lceil\log_2(8M)\rceil + 2s$ such that

$$|\bar{h}(\tilde{\boldsymbol{x}}) - \bar{f}(\tilde{\boldsymbol{x}})| \leq 19 B_0 (\lfloor\beta\rfloor + 1)^2 s^{\lfloor\beta\rfloor + (\beta \vee 1)/2} (NM)^{-2\beta/s}, \tilde{\boldsymbol{x}} \in [0,1]^s. \tag{13}$$

Knowing that $\left(\boldsymbol{\theta} \odot \boldsymbol{\Gamma}^\top\right)^\top$ can be rearranged as $\left(\boldsymbol{u}^{w^{(1)} \times s}, \mathbf{0}^{w^{(1)} \times (p-1-s)}\right)$, we have $\left(\boldsymbol{\theta} \odot \boldsymbol{\Gamma}^\top\right)^\top \boldsymbol{x} = \boldsymbol{u}\tilde{\boldsymbol{x}}$. Thus, for any low-dimensional Relu network $\bar{f}(\tilde{\boldsymbol{x}})$ there always exists a high-dimensional sparse ReLU network $f(\boldsymbol{x})$ satisfying $\bar{f}(\tilde{\boldsymbol{x}}) = f(\boldsymbol{x})$. This completes the proof.

A.3.2 Supporting Lemmas

Lemma A.1. *1. There exists $c_0 > 0$ such that $\theta_{\boldsymbol{\alpha} C_j, C_k} \geq c_0$ and $\left\|\boldsymbol{\Gamma}_{\boldsymbol{\alpha}_j}^{[:, c_j(k)]}\right\| \geq c_0$ for all $k \in \mathcal{A}_j$ and $\boldsymbol{\alpha} \in \mathcal{H}^*$.*

2. Denote $\phi(\boldsymbol{\alpha})$ the vector obtained from $\boldsymbol{\alpha}$ by setting $\boldsymbol{\Gamma}_{\boldsymbol{\alpha}_j}^{[:, c_j(k)]} = 0$ for all $(j,k) \in \bigcup_{j=1}^p \{j\} \times \mathcal{B}_j$ and $\theta_{\boldsymbol{\alpha} l, l'} = 0$ for all $(l, l') \in \bigcup_{l=1}^L \{l\} \times \mathcal{P}_l$. For $\boldsymbol{\alpha} \in \mathcal{H}^*$, $\phi(\boldsymbol{\alpha})$ also belongs to $\mathcal{H}^*$.

Proof. By Theorem 3.2, we establish the uniform convergence:

$$\left|h_j(\boldsymbol{x}_{-j}) - f_{\boldsymbol{\alpha}_j^*}(\boldsymbol{x}_{-j})\right| \to 0 \quad \text{as } W, D \to \infty. \tag{14}$$

This implies that $f_{\boldsymbol{\alpha}_j^*}(\boldsymbol{x}_{-j}) \to \mathbb{E}[x_j | \boldsymbol{x}_{-j}]$ serves as the unique minimizer of $R_j(\boldsymbol{\alpha}_j)$. Consequently, the composite risk functional $R(\boldsymbol{\alpha}^*) = \sum_{j=1}^p R_j(\boldsymbol{\alpha}_j^*)$ attains its global minimum through coordinate-wise optimization. For any competing parameter $\boldsymbol{\alpha}_0 \in \mathcal{H}^*$, this construction ensures:

$$R_j(\boldsymbol{\alpha}_{0j}) = R_j(\boldsymbol{\alpha}_j^*), \quad \forall j = 1, \ldots, p. \tag{15}$$

The fundamental inequality

$$R_j(\boldsymbol{\alpha}_j^*) = \min_g \mathbb{E}\left[(x_j - g(\boldsymbol{x}_{-j}))^2\right] \leq \min_{\alpha_j \in \mathcal{W}_j} R_j(\boldsymbol{\alpha}_j) = R_j(\boldsymbol{\alpha}_{0j}), \tag{16}$$

holds with equality if and only if $f_{\boldsymbol{\alpha}_j} = f_{\boldsymbol{\alpha}_j^*}$ almost surely, where g can be any measurable function from the input space to the real numbers. Therefore, the identifiability condition

$$\boldsymbol{\alpha}_0 \in \mathcal{H}^* \iff f_{\boldsymbol{\alpha}_{0j}} = f_{\boldsymbol{\alpha}_j^*}, \quad \forall j = 1, \ldots, p, \tag{17}$$

follows necessarily.

1: Assuming that no such c_0 exists, there exist $\boldsymbol{\alpha}_0 \in \mathcal{H}^*$ and $k \in \mathcal{A}_j$ such that $\left(\boldsymbol{\theta}_{\boldsymbol{\alpha}_0 C_j} \odot \boldsymbol{\Gamma}_{\boldsymbol{\alpha}_0 j}^\top\right)^{\top[:, c_j(k)]} = 0$. This means $f_{\boldsymbol{\alpha}_{0j}} = f_{\boldsymbol{\alpha}_j^*}$ does not depend on the related variable x_k, which is a contradiction.

2: Denote $\phi(\boldsymbol{\alpha}_j)$ the sub-vector of $\phi(\boldsymbol{\alpha})$ corresponding to $\boldsymbol{\alpha}_j$. Since $\boldsymbol{\alpha} \in \mathcal{H}^*$, we have $f_{\boldsymbol{\alpha}_j^*}(\boldsymbol{x}_{\mathcal{B}_j}, \boldsymbol{x}_{\mathcal{B}_j^c}) = f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{\mathcal{B}_j}, \boldsymbol{x}_{\mathcal{B}_j^c}) = f_{\boldsymbol{\alpha}_j}(\mathbf{0}, \boldsymbol{x}_{\mathcal{B}_j^c}) = f_{\phi(\boldsymbol{\alpha}_j)}(\boldsymbol{x}_{\mathcal{B}_j}, \boldsymbol{x}_{\mathcal{B}_j^c})$, which implies $\phi(\boldsymbol{\alpha}) \in \mathcal{H}^*$. □

Lemma A.2. *Let $p = o(\log n)$, $SD \log S = O(n^{\frac{1}{4}})$, then there exists $c_1 > 0$ such that*

$$|R_n(\boldsymbol{\alpha}) - R(\boldsymbol{\alpha})| \leq c_1 \frac{\log n}{n^{\frac{1}{4}}}, \quad \forall \boldsymbol{\alpha} \in \mathcal{W}, \tag{18}$$

holds with probability at least $1 - \delta_1$ with $\delta_1 = 4n\left(n^{\frac{1}{4}} + 1\right)^{\log n}\left(32en^{\frac{1}{4}}\right)^{(\log n) n^{\frac{1}{4}}} e^{-\frac{\sqrt{n}\log n}{32}}$.

Proof. Let $l_{f_j}(\boldsymbol{x}_i) = \left(x_{ij} - f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j})\right)^2$ and $l_f(\boldsymbol{x}_i) = \sum_{j=1}^p l_{f_j}(\boldsymbol{x}_i)$. For $x \in \mathcal{X}^n$, denote $R_n(f_x) = \frac{1}{n}\sum_{i=1}^n l_f(\boldsymbol{x}_i)$ and $R(f) = \mathbb{E}\left[l_f(\boldsymbol{x})\right]$.

Define

$$Q = \left\{\boldsymbol{x} \in \mathcal{X}^n : \exists f_{\boldsymbol{\alpha}_1} \cdots f_{\boldsymbol{\alpha}_p} \in \mathcal{F} \text{ s.t. } |R(f) - R_n(f_x)| \geq \varepsilon\right\}, \tag{19}$$

and

$$R = \left\{(\boldsymbol{r}, \boldsymbol{s}) \in \mathcal{X}^n \times \mathcal{X}^n : \exists f_{\boldsymbol{\alpha}_1} \cdots f_{\boldsymbol{\alpha}_p} \in \mathcal{F} \text{ s.t. } |R_n(f_r) - R_n(f_s)| \geq \frac{\varepsilon}{2}\right\}. \tag{20}$$

Since $\left\{|R(f) - R_n(f_r)| \geq \varepsilon \text{ and } |R(f) - R_n(f_s)| < \frac{\varepsilon}{2}\right\} \subset \left\{|R_n(f_r) - R_n(f_s)| \geq \frac{\varepsilon}{2}\right\}$, we have

$$\begin{aligned}\mathbb{P}(R) &\geq \mathbb{P}\left\{\exists f_{\boldsymbol{\alpha}_1} \cdots f_{\boldsymbol{\alpha}_p} \in \mathcal{F} \text{ s.t. } |R(f) - R_n(f_r)| \geq \varepsilon \text{ and } |R(f) - R_n(f_s)| < \frac{\varepsilon}{2}\right\} \\ &= \int_Q \mathbb{P}\left\{s : \exists f_{\boldsymbol{\alpha}_1} \cdots f_{\boldsymbol{\alpha}_p} \in \mathcal{F}, |R(f) - R_n(f_r)| \geq \varepsilon \text{ and} |R(f) - R_n(f_s)| < \varepsilon/2\right\} d\mathbb{P}(r).\end{aligned} \tag{21}$$

Noting that $f_{\boldsymbol{\alpha}_j}$ maps into $[0,1]$ and $x_{ij} \in [0,1]$, we have $l_{f_j}(\boldsymbol{x}_i) \in [0,1]$ and $l_f(\boldsymbol{x}_i) \in [0,p]$. Hoeffding's inequality for bounded random variables shows that

$$\mathbb{P}\left(|R(f) - R_n(f_s)| \leq \frac{\varepsilon}{2}\right) \geq 1 - \exp\left(-\frac{n\varepsilon^2}{4p}\right) \geq \frac{1}{2}, \tag{22}$$

for $n > \frac{4p}{\varepsilon^2}$. Thus, $\mathbb{P}(Q) \leq 2\mathbb{P}(R)$ for $n > \frac{4p}{\varepsilon^2}$.

In order to bound $\mathbb{P}(R)$, the technique of permutation and reduction to a finite class is useful. Denote T_m the set of permutations on $\{1, 2, \ldots, 2n\}$ that switch elements i and $n+i$, for i in some subset of $\{1, 2, \ldots, 2n\}$. A permutation σ is chosen uniformly at random from T_m. By Lemma 4.5 of [1],

$$\mathbb{P}(R) = \mathbb{E}\mathbb{P}(\sigma \boldsymbol{x} \in R) \leq \max_{\boldsymbol{x} \in \mathcal{X}^{2n}} \mathbb{P}(\sigma \boldsymbol{x} \in R). \tag{23}$$

where the expectation is over $\boldsymbol{x}$ and the probability is over permutations σ.

Denote $\mathcal{L}_{\mathcal{F}} = \left\{l_f(\boldsymbol{x}) = \sum_{j=1}^p \left(x_j - f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{-j})\right)^2 : f_{\boldsymbol{\alpha}_j} \in \mathcal{F}, j = 1, \ldots, p\right\}$. Let $N\left(\varepsilon, \mathcal{F}, L^1(P_n)\right)$ be the covering number of $\mathcal{F}$ under the empirical $L^1(P_n)$ metric with radius ε. For a given sequence $x = (x_1, \ldots, x_n) \in \mathcal{X}^n$, let $\mathcal{F}|_x = \{(f(x_1), \ldots, f(x_n)) : f \in \mathcal{F}\}$ be the subset of $\mathbb{R}^n$. Define the uniform covering number

$$\mathcal{N}_n\left(\varepsilon, \mathcal{F}, L^1(P_n)\right) = \max\left\{\mathcal{N}\left(\varepsilon, \mathcal{F}|_x, L^1(P_n)\right) : x \in \mathcal{X}^n\right\}. \tag{24}$$

Suppose that $\boldsymbol{x} \in \mathcal{X}^{2n}$ and let $\mathcal{T}$ be a minimal $\frac{\varepsilon}{8}$-cover for $\mathcal{L}_{\mathcal{F}}|_x$ with respect to the $L^1(P_n)$ metric. Pick $\mathcal{G} \subset \mathcal{F}$ such that $\mathcal{T} = \mathcal{L}_{\mathcal{G}}|_x$ and $|\mathcal{G}| = |\mathcal{T}|$. Let $\boldsymbol{r} \in \mathcal{X}^n$ and $\boldsymbol{s} \in \mathcal{X}^n$ so that $\boldsymbol{x} = (\boldsymbol{r}, \boldsymbol{s})$. Suppose that $f \in \mathcal{F}$ satisfies $|R_n(f_r) - R_n(f_s)| \geq \frac{\varepsilon}{2}$ and there exists $g \in \mathcal{G}$ such that $\frac{1}{2n}\sum_{i=1}^{2n} |l_f(\boldsymbol{x}_i) - l_g(\boldsymbol{x}_i)| < \frac{\varepsilon}{8}$. Then,

$$\begin{aligned}|R_n(g_r) - R_n(g_s)| &= \left|\frac{1}{n}\sum_{i=1}^n l_g(\boldsymbol{x}_i) - \frac{1}{n}\sum_{i=n+1}^{2n} l_g(\boldsymbol{x}_i)\right| \\ &= \left|\frac{1}{n}\sum_{i=1}^n (l_g(\boldsymbol{x}_i) - l_f(\boldsymbol{x}_i)) - \frac{1}{n}\sum_{i=n+1}^{2n} (l_g(\boldsymbol{x}_i) - l_f(\boldsymbol{x}_i)) + R_n(f_r) - R_n(f_s)\right| \\ &\geq |R_n(f_r) - R_n(f_s)| - \\ &\quad \left|\frac{1}{n}\sum_{i=1}^n (l_g(\boldsymbol{x}_i) - l_f(\boldsymbol{x}_i)) - \frac{1}{n}\sum_{i=n+1}^{2n} (l_g(\boldsymbol{x}_i) - l_f(\boldsymbol{x}_i))\right| \\ &\geq |R_n(f_r) - R_n(f_s)| - \frac{1}{n}\sum_{i=1}^{2n} |l_g(\boldsymbol{x}_i) - l_f(\boldsymbol{x}_i)| \\ &> \varepsilon/4.\end{aligned} \tag{25}$$

Thus,

$$\begin{aligned}\mathbb{P}(\sigma\boldsymbol{x}\in R) &\le \mathbb{P}\left(\exists g\in\mathcal{G}:\left|\frac{1}{n}\sum_{i=1}^{n}\left(l_g(\boldsymbol{x}_{\sigma(i)})-l_g(\boldsymbol{x}_{\sigma(n+i)})\right)\right|\ge\varepsilon/4\right)\\ &\le |\mathcal{G}|\max_{g\in\mathcal{G}}\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}\left(l_g(\boldsymbol{x}_{\sigma(i)})-l_g(\boldsymbol{x}_{\sigma(n+i)})\right)\right|\ge\varepsilon/4\right)\\ &= |\mathcal{G}|\max_{g\in\mathcal{G}}\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}\left|l_g(\boldsymbol{x}_i)-l_g(\boldsymbol{x}_{n+i})\right|\epsilon_i\right|\ge\varepsilon/4\right)\\ &\le |\mathcal{G}|2\exp\left(-\frac{\varepsilon^2 n}{32p}\right),\end{aligned} \tag{26}$$

where each ϵ_i is independently and uniformly drawn from $\{-1,1\}$.

Noting that

$$\begin{aligned}\|l_f-l_{f'}\|_{L^1(P_n)} &= \frac{1}{2n}\sum_{i=1}^{2n}|l_f(\boldsymbol{x}_i)-l_{f'}(\boldsymbol{x}_i)|\\ &= \frac{1}{2n}\sum_{i=1}^{2n}\left|\sum_{j=1}^{p}\left([x_{ij}-f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j})]^2-[x_{ij}-f'_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j})]^2\right)\right|\\ &= \frac{1}{2n}\sum_{i=1}^{2n}\left|\sum_{j=1}^{p}(f_{\boldsymbol{\alpha}_j}-f'_{\boldsymbol{\alpha}_j})(f_{\boldsymbol{\alpha}_j}+f'_{\boldsymbol{\alpha}_j}-2x_{ij})\right|\\ &\le \frac{1}{n}\sum_{i=1}^{2n}\sum_{j=1}^{p}|f_{\boldsymbol{\alpha}_j}-f'_{\boldsymbol{\alpha}_j}|,\end{aligned} \tag{27}$$

we can translate a cover of $\mathcal{L}_{\mathcal{F}}|_x$ into a cover of the function space $\mathcal{F}$, that is,

$$|\mathcal{G}| = N\left(\frac{\varepsilon}{8},\mathcal{L}_{\mathcal{F}}|_x,L^1(P_n)\right)\le N_{2n}\left(\frac{\varepsilon}{16},\mathcal{F}^p,L^1(P_n)\right)\le N_{2n}\left(\frac{\varepsilon}{16p},\mathcal{F},L^1(P_n)\right)^p. \tag{28}$$

For a class of functions $\mathcal{F}$, the pseudo dimension, denoted by $Pdim(\mathcal{F})$, is a natural measure of its complexity. According to [1], $Pdim(\mathcal{F})$ is the largest integer m for which there exists $(x_1,\ldots,x_m,y_1,\ldots,y_m)\in\mathcal{X}^m\times\mathbb{R}^m$ such that for any $(b_1,\ldots,b_m)\in\{0,1\}^m$ there exists $f\in\mathcal{F}$ such that $\forall i: f(x_i)>y_i \iff b_i=1$. Using Theorem 18.4 in [1], we can give an upper bound on the covering number by $Pdim(\mathcal{F})$. Suppose that $Pdim(\mathcal{F})=d$, we have

$$\mathcal{N}_n\left(\varepsilon,\mathcal{F},L^1(P_n)\right)\le e(d+1)\left(\frac{2e}{\varepsilon}\right)^d. \tag{29}$$

Moreover, based on Theorems 3 and 6 in [2], the pseudo dimension of ReLU feedforward neural network space is bounded as

$$c\cdot SD\log(S/D)\le \mathrm{Pdim}(\mathcal{F})\le C\cdot SD\log(S). \tag{30}$$

Thus,

$$\begin{aligned}\mathbb{P}(Q) &\le 4N_{2n}\left(\frac{\varepsilon}{16p},\mathcal{F},L^1(P_n)\right)^p\exp\left(-\frac{\varepsilon^2 n}{32p}\right)\\ &\le 4e^p\left(SD\log(S)+1\right)^p\left(\frac{32pe}{\varepsilon}\right)^{pSD\log(S)}\exp\left(-\frac{\varepsilon^2 n}{32p}\right).\end{aligned} \tag{31}$$

Let $p=o(\log n),\varepsilon=O\left(\frac{\log n}{n^{\frac{1}{4}}}\right)$ and $SD\log S=O(n^{\frac{1}{4}})$, it is obvious that $n>\frac{4p}{\varepsilon^2}$ and $\delta_1=4e^p\left(SD\log(S)+1\right)^p\left(\frac{32pe}{\varepsilon}\right)^{pSD\log(S)}\exp\left(-\frac{\varepsilon^2 n}{32p}\right)\lesssim 4n\left(n^{\frac{1}{4}}+1\right)^{\log n}\left(32en^{\frac{1}{4}}\right)^{(\log n)n^{\frac{1}{4}}}e^{-\frac{\sqrt{n}\log n}{32}}\to 0$. This completes the proof. □

Lemma A.3. *There exists a constant M_{δ_2} such that with probability at least $1-\delta_2$, the empirical risk function $R_n(\boldsymbol{\alpha})$ is $M_{\delta_2}\sqrt{p}$-Lipschitz.*

Proof. For the empirical risk:

$$\begin{aligned}
|R_n(\boldsymbol{\alpha}) - R_n(\boldsymbol{\beta})| &= \left| \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}\left[\left(x_{ij} - f_{\boldsymbol{\alpha}_j}(x_{i,-j})\right)^2 - \left(x_{ij} - f_{\boldsymbol{\beta}_j}(x_{i,-j})\right)^2\right]\right| \\
&= \left| \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}\left(f_{\boldsymbol{\alpha}_j}(x_{i,-j}) - f_{\beta_j}(x_{i,-j})\right)\left(f_{\boldsymbol{\alpha}_j}(x_{i,-j}) + f_{\boldsymbol{\beta}_j}(x_{i,-j}) - 2x_{ij}\right)\right| \\
&\le \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n}\left|f_{\boldsymbol{\alpha}_j}(x_{i,-j}) - f_{\beta_j}(x_{i,-j})\right| \\
&\quad \left(\left|f_{\boldsymbol{\alpha}_j}(x_{i,-j}) - f_{\boldsymbol{\alpha}_j^*}(x_{i,-j})\right| + \left|f_{\boldsymbol{\beta}_j}(x_{i,-j}) - f_{\boldsymbol{\alpha}_j^*}(x_{i,-j})\right| + 2\left|\varepsilon_{ij}\right|\right) \\
&\le \frac{1}{n}\sum_{j=1}^{p}\sum_{i=1}^{n} B\|\boldsymbol{\alpha}_j - \boldsymbol{\beta}_j\| \cdot (2 + 2|\varepsilon_{ij}|) \\
&= B\sum_{j=1}^{p}\|\boldsymbol{\alpha}_j - \boldsymbol{\beta}_j\|\left(2 + \frac{2}{n}\sum_{i=1}^{n}|\varepsilon_{ij}|\right).
\end{aligned} \tag{32}$$

For a zero-mean sub-Gaussian random variable ε_{ij}, we know that there exists a constant C_1 such that $\mathbb{E}|\varepsilon_{ij}| \le C_1\sigma$. Applying Markov's inequality:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i=1}^{n}|\varepsilon_{ij}| > C_{\delta_2}\right) \le \frac{\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}|\varepsilon_{ij}|}{C_{\delta_2}}. \tag{33}$$

Choosing $C_{\delta_2} = C_1\sigma/\delta_2$, we obtain

$$|R_n(\boldsymbol{\alpha}) - R_n(\boldsymbol{\beta})| \le B(2 + 2C_{\delta_2})\sqrt{p}\|\boldsymbol{\alpha} - \boldsymbol{\beta}\| = M_{\delta_2}\sqrt{p}\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|, \tag{34}$$

with probability at least $1-\delta_2$. □

A.3.3 Proof of Theorem 3.6

Define $\boldsymbol{\beta}_n = \operatorname{argmin}_{\boldsymbol{\alpha}\in\mathcal{H}^*}\|\hat{\boldsymbol{\alpha}}_n - \boldsymbol{\alpha}\|$. Let $L(\boldsymbol{\alpha}) = \sum_{j=1}^{p}\sum_{k=1}^{p-1}\left\|\boldsymbol{\Gamma}_j^{[:,k]}\right\| + \sum_{l<l'}|\theta_{ll'}|$ and $\lambda = \max(\lambda_1, \lambda_2)$. Since $L(\boldsymbol{\alpha})$ is Lipschitz, that is,

$$\begin{aligned}
L(\boldsymbol{\beta}_n) - L(\hat{\boldsymbol{\alpha}}_n) &= \sum_{j=1}^{p}\sum_{k=1}^{p-1}\left\|\boldsymbol{\Gamma}_{\boldsymbol{\beta}_{nj}}^{[:,k]}\right\| + \sum_{l<l'}|\theta_{\boldsymbol{\beta}_n ll'}| - \sum_{j=1}^{p}\sum_{k=1}^{p-1}\left\|\boldsymbol{\Gamma}_{\hat{\boldsymbol{\alpha}}_{nj}}^{[:,k]}\right\| - \sum_{l<l'}|\theta_{\hat{\boldsymbol{\alpha}}_n ll'}| \\
&\le \sum_{j=1}^{p}\sum_{k=1}^{p-1}\left\|\boldsymbol{\Gamma}_{\boldsymbol{\beta}_{nj}}^{[:,k]} - \boldsymbol{\Gamma}_{\hat{\boldsymbol{\alpha}}_{nj}}^{[:,k]}\right\| + \sum_{l<l'}|\theta_{\boldsymbol{\beta}_n ll'} - \theta_{\hat{\boldsymbol{\alpha}}_n ll'}| \\
&\le \sqrt{p(p-1)}\left\|\boldsymbol{\Gamma}_{\boldsymbol{\beta}_n} - \boldsymbol{\Gamma}_{\hat{\boldsymbol{\alpha}}_n}\right\| + \frac{L(L+1)}{2}\|\boldsymbol{\theta}_{\boldsymbol{\beta}_n} - \boldsymbol{\theta}_{\hat{\boldsymbol{\alpha}}_n}\| \\
&\le C\sqrt{p(p-1)}\|\boldsymbol{\beta}_n - \hat{\boldsymbol{\alpha}}_n\|.
\end{aligned} \tag{35}$$

Combined with Lemma A.2, we have

$$\begin{aligned}
c_2\|\boldsymbol{\beta}_n - \hat{\boldsymbol{\alpha}}_n\|^\nu = c_2 d(\hat{\boldsymbol{\alpha}}_n, \mathcal{H}^*)^\nu &\le R(\hat{\boldsymbol{\alpha}}_n) - R(\boldsymbol{\beta}_n) \\
&\le 2c_1\frac{\log n}{n^{\frac{1}{4}}} + \lambda\left(L(\boldsymbol{\beta}_n) - L(\hat{\boldsymbol{\alpha}}_n)\right) \\
&\le 2c_1\frac{\log n}{n^{\frac{1}{4}}} + \lambda C\sqrt{p(p-1)}\|\boldsymbol{\beta}_n - \hat{\boldsymbol{\alpha}}_n\|,
\end{aligned} \tag{36}$$

holds with probability at least $1-\delta_1$.

Applying Young's inequality,

$$\begin{aligned}\lambda C\sqrt{p(p-1)}\|\boldsymbol{\beta}_n-\hat{\boldsymbol{\alpha}}_n\| &\le \frac{1}{\nu}\left(\frac{(c_2\nu)^{1/\nu}}{2}\|\boldsymbol{\beta}_n-\hat{\boldsymbol{\alpha}}_n\|\right)^{\nu}+\frac{\nu-1}{\nu}\left(\frac{2C\sqrt{p(p-1)}}{(c_2\nu)^{1/\nu}}\lambda\right)^{\nu/(\nu-1)}\\ &\le \frac{c_2}{2}\|\boldsymbol{\beta}_n-\hat{\boldsymbol{\alpha}}_n\|^{\nu}+C_\nu\left(\lambda\sqrt{p(p-1)}\right)^{\nu/(\nu-1)},\end{aligned} \tag{37}$$

yielding $\|\boldsymbol{\beta}_n-\hat{\boldsymbol{\alpha}}_n\| \le C'\left(\left(\lambda\sqrt{p(p-1)}\right)^{\frac{\nu}{\nu-1}}+\frac{\log n}{n^{\frac{1}{4}}}\right)^{\frac{1}{\nu}}$. Let $p=o(\log n), SD\log(S)=O(n^{\frac{1}{4}}), \lambda=O(n^{-\frac{1}{8}})$. Since $\nu>2$, $1<\frac{\nu}{\nu-1}<2$, there exists $c_3>0$ such that $d(\hat{\boldsymbol{\alpha}}_n,\mathcal{H}^*)\le c_3\left(\frac{\log n}{n^{\frac{1}{8}}}\right)^{\frac{1}{\nu-1}}$.

Let K denote the inactive regularization components of L. Since K is Lipschitz and $K(\phi(\boldsymbol{\alpha}))=K(\boldsymbol{\alpha})$, we have

$$\begin{aligned}\lambda_1\sum_{l=1}^{L}\sum_{l'\in\mathcal{P}_l}|\hat{\theta}_{l,l'}|+\lambda_2\sum_{j=1}^{p}\sum_{k\in\mathcal{B}_j}\left\|\hat{\boldsymbol{\Gamma}}_j^{[:,c_j(k)]}\right\| &\le R_n(\phi(\boldsymbol{\beta}_n))-R_n(\hat{\boldsymbol{\alpha}}_n)+\lambda[K(\phi(\boldsymbol{\beta}_n))-K(\hat{\boldsymbol{\alpha}}_n)]\\ &\le 2c_1\frac{\log n}{n^{\frac{1}{4}}}+R(\phi(\boldsymbol{\beta}_n))-R(\hat{\boldsymbol{\alpha}}_n)\\ &\quad+\lambda[K(\boldsymbol{\beta}_n)-K(\hat{\boldsymbol{\alpha}}_n)]\\ &\le 2c_1\frac{\log n}{n^{\frac{1}{4}}}+\lambda C\sqrt{p(p-1)}\|\boldsymbol{\beta}_n-\hat{\boldsymbol{\alpha}}_n\|.\end{aligned} \tag{38}$$

Similarly, since $\nu>2$, $0<\frac{1}{\nu-1}<1$, there exists $c_4>0$ such that $\sum_{j=1}^{p}\sum_{k\in\mathcal{B}_j}\left\|\hat{\boldsymbol{\Gamma}}_j^{[:,c_j(k)]}\right\|+\sum_{l=1}^{L}\sum_{l'\in\mathcal{P}_l}|\hat{\theta}_{l,l'}|\le c_4\log n\left(\frac{\log n}{n^{\frac{1}{8}}}\right)^{\frac{1}{\nu-1}}$. This completes the proof.

A.3.4 Proof of Theorem 3.7

By Theorem 3.6 and Lemma A.3, we have that for all $(j,k)\in\bigcup_{j=1}^{p}\{j\}\times\mathcal{A}_j$, with probability $1-\delta_1$, $\hat{\boldsymbol{\Gamma}}_j^{[:,c_j(k)]}$ and $\hat{\theta}_{C_j,C_k}$ are bounded away from zero as $n\to\infty$. Our analysis considers two connection structures: variable-level connection and group-level connection. The fundamental dependency principle requires that variables from statistically independent groups must exhibit no conditional dependence. That is, $\theta^*_{C_j,C_k}=0$ implies $\boldsymbol{\Gamma}_j^{*[:,c_j(k)]}=0$. Let $M(\boldsymbol{\alpha})=\sum_{j=1}^{p}\sum_{k=1}^{p-1}\frac{\left\|\boldsymbol{\Gamma}_j^{[:,k]}\right\|}{\left\|\hat{\boldsymbol{\Gamma}}_j^{[:,k]}\right\|^{\gamma}}+\sum_{l<l'}\frac{|\theta_{ll'}|}{|\hat{\theta}_{ll'}|^{\gamma}}$ and $\zeta=\max(\zeta_1,\zeta_2)$. Thus,

$$M(\boldsymbol{\alpha}^*)=\sum_{j=1}^{p}\sum_{k\notin\mathcal{B}_j}\frac{\left\|\boldsymbol{\Gamma}_j^{*[:,c_j(k)]}\right\|}{\left\|\hat{\boldsymbol{\Gamma}}_j^{[:,c_j(k)]}\right\|^{\gamma}}+\sum_{l=1}^{L}\sum_{l'\notin\mathcal{P}_l}\frac{|\theta^*_{ll'}|}{|\hat{\theta}_{ll'}|^{\gamma}}<\infty, \tag{39}$$

and

$$c_2 d(\tilde{\boldsymbol{\alpha}}_n,\mathcal{H}^*)^{\nu}\le 2c_1\frac{\log n}{n^{\frac{1}{4}}}+\zeta\left(M(\boldsymbol{\alpha}^*)-M(\hat{\boldsymbol{\alpha}}_n)\right)\le 2c_1\frac{\log n}{n^{\frac{1}{4}}}+\zeta M(\boldsymbol{\alpha}^*). \tag{40}$$

Let $\zeta=O\left(n^{-\frac{\gamma}{8(\nu-1)}+\epsilon}\right)$, there exists $c_5>0$ such that $d(\tilde{\boldsymbol{\alpha}}_n,\mathcal{H}^*)\le c_5 n^{\left(-\frac{\gamma}{8(\nu-1)}+\epsilon\right)/\nu}\to 0$ with probability $1-\delta_1$. Thus, by Lemma A.1, $\tilde{\theta}_{C_j,C_k}$ and $\tilde{\boldsymbol{\Gamma}}_j^{[:,c_j(k)]}$ are bounded away from zero for all $k\in\mathcal{A}_j$ and large enough n.

To prove true negativity, we can separately prove that $\widetilde{\boldsymbol{\Gamma}}_j^{[:,c_j(k)]}=0$ for all $(j,k)\in\bigcup_{j=1}^{p}\{j\}\times\mathcal{B}_j$ and $\widetilde{\theta}_{l,l'}=0$ for all $(l,l')\in\bigcup_{l=1}^{L}\{l\}\times\mathcal{P}_l$. We establish the result by contradiction. Suppose there

exist some j and $k \in \mathcal{B}_j$ with $\tilde{\mathbf{\Gamma}}_j^{[:,c_j(k)]} \neq \mathbf{0}$. Define $\boldsymbol{g}_n$ the vector obtained from $\tilde{\boldsymbol{\alpha}}_n$ by setting the $\tilde{\mathbf{\Gamma}}^{[:,c_j(k)]}$ component to 0, then we have $R_n(\tilde{\boldsymbol{\alpha}}_n) + \zeta_2 \frac{\left\|\tilde{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|}{\left\|\hat{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|^\gamma} \leq R_n(\boldsymbol{g}_n)$. By Lemma A.3, there exists M_{δ_2} such that

$$\zeta_2 \frac{\left\|\tilde{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|}{\left\|\hat{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|^\gamma} \leq R_n(\boldsymbol{g}_n) - R_n(\tilde{\boldsymbol{\alpha}}_n) \leq M_{\delta_2}\sqrt{p}\|\boldsymbol{g}_n - \tilde{\boldsymbol{\alpha}}_n\| = M_{\delta_2}\sqrt{p}\left\|\tilde{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|, \tag{41}$$

with probability at least $1 - \delta_2$. Since $\tilde{\mathbf{\Gamma}}^{[:,c_j(k)]} \neq 0$, we deduce that $\zeta_2 \frac{1}{\left\|\hat{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|^\gamma} \leq M_{\delta_2}\sqrt{p}$. This contradicts Theorem 3.6, which proves that for n large enough $\zeta_2 \frac{1}{\left\|\hat{\mathbf{\Gamma}}_j^{[:,c_j(k)]}\right\|^\gamma} \geq \zeta_2 c_4^{-\gamma} (\log n)^{\frac{-\gamma\nu}{\nu-1}} n^{\frac{\gamma}{8(\nu-1)}} \geq 2M_{\delta_2}\sqrt{p}$, with probability at least $1 - \delta_1$. Thus, by Bonferroni inequality, we have $\tilde{\mathbf{\Gamma}}^{[:,c_j(k)]} = 0$ for all $k \in \mathcal{B}_j$ with probability at least $1 - \delta_1 - \delta_2$. Similarly, it can be inferred that $\tilde{\theta}_{ll'} = 0$ for all $l' \in \mathcal{P}_l$ with probability at least $1 - \delta_1 - \delta_2$. This completes the proof.

A.4 Computation

A.4.1 Algorithm

Algorithm 1: Two-Stage Proximal Gradient Descent for NNBLNet

Input: Data $\{\boldsymbol{x}_i\}_{i=1}^n$, learning rate η, regularization parameters $\zeta_1 = \lambda_1, \zeta_2 = \lambda_2$, power γ, number of epochs T, tolerance ϵ

Output: Estimated parameters $\{\boldsymbol{\theta}, \mathbf{\Gamma}_j, \{\Delta_j^{(l)}\}_{l=1}^D\}_{j=1}^p$

Stage 1: Initial Estimation (non-adaptive)

Initialize $\boldsymbol{\theta}$, $\mathbf{\Gamma}_j$, and $\Delta_j^{(l)}$ for all j and l

for $i = 1$ **to** T **do**

  for $j = 1$ **to** p **do**

    Compute predictions: $\hat{x}_{i,j} = f_{\boldsymbol{\alpha}_j}(\boldsymbol{x}_{i,-j})$

    Compute gradients: $\nabla_{\mathbf{\Gamma}_j}$, $\nabla_{\boldsymbol{\theta}}$, $\nabla_{\Delta_j^{(l)}}$ for all l

    Gradient step:

      $\mathbf{\Gamma}_j^{\text{tmp}} \leftarrow \mathbf{\Gamma}_j - \eta\nabla_{\mathbf{\Gamma}_j}$

      $\boldsymbol{\theta}^{\text{tmp}} \leftarrow \boldsymbol{\theta} - \eta\nabla_{\boldsymbol{\theta}}$

      $\Delta_j^{(l)} \leftarrow \Delta_j^{(l)} - \eta\nabla_{\Delta_j^{(l)}}$ for all l

    Proximal update:

      $\mathbf{\Gamma}_j^{[:,k]} \leftarrow \left(1 - \frac{\eta\zeta_1}{\|\mathbf{\Gamma}_j^{[:,k],\text{tmp}}\|}\right)_+ \mathbf{\Gamma}_j^{[:,k],\text{tmp}}$

      $\theta_{ll'} \leftarrow \text{sign}(\theta_{ll'}^{\text{tmp}}) \cdot \max(|\theta_{ll'}^{\text{tmp}}| - \eta\zeta_2, 0)$

Store estimates $\hat{\mathbf{\Gamma}}, \hat{\boldsymbol{\theta}}$ and compute adaptive weights:

$w_{jk}^{(1)} \leftarrow \frac{1}{\|\hat{\mathbf{\Gamma}}_j^{[:,k]}\|^\gamma + \epsilon}$, $w_{ll'}^{(2)} \leftarrow \frac{1}{|\hat{\theta}_{ll'}|^\gamma + \epsilon}$

Stage 2: Adaptive Estimation

Reinitialize $\mathbf{\Gamma}_j$ and $\boldsymbol{\theta}$, keep $\Delta_j^{(l)}$ from Stage 1 (or reinitialize optionally)

for $i = 1$ **to** T **do**

  for $j = 1$ **to** p **do**

    Compute predictions and gradients as in Stage 1

    Gradient step and update $\Delta_j^{(l)}$ as before

    Proximal update with adaptive weights:

      $\mathbf{\Gamma}_j^{[:,k]} \leftarrow \left(1 - \frac{\eta\zeta_1 w_{jk}^{(1)}}{\|\mathbf{\Gamma}_j^{[:,k],\text{tmp}}\|}\right)_+ \mathbf{\Gamma}_j^{[:,k],\text{tmp}}$

      $\theta_{ll'} \leftarrow \text{sign}(\theta_{ll'}^{\text{tmp}}) \cdot \max(|\theta_{ll'}^{\text{tmp}}| - \eta\zeta_2 w_{ll'}^{(2)}, 0)$

A.4.2 Selection of the Regularization Parameters

Based on the convergence requirements specified in Theorems 3.6 and 3.7, we established the parameter configuration $\lambda_k = \zeta_k = cn^{-1/8}$ for $k = 1, 2$. Computationally, $n^{-1/8}$ yields values between 0.46 and 0.26 for sample sizes ranging from 500 to 50,000. To determine the optimal c, in Table 5, we evaluated F1-score performance across values $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 1, 1.5\}$ at different sample sizes. This theoretically derived value achieved stable performance across varying n at $c = 0.35$. We therefore use $c = 0.35$ in practical implementations to maintain computational efficiency without compromising on accuracy.

Table 5: F1 score of NNBLNet with different sample sizes and values of c under linear and nonlinear scenarios

		Sample size n					
Scenario	c	500	1000	2000	5000	10000	20000
Nonlinear	0.05	0.659 (0.033)	0.666 (0.026)	0.673 (0.023)	0.674 (0.022)	0.670 (0.019)	0.676 (0.017)
	0.10	0.695 (0.032)	0.704 (0.028)	0.713 (0.022)	0.716 (0.018)	0.719 (0.016)	0.717 (0.014)
	0.15	0.720 (0.030)	0.729 (0.025)	0.731 (0.019)	0.738 (0.017)	0.735 (0.014)	0.742 (0.013)
	0.20	0.738 (0.024)	0.749 (0.021)	0.754 (0.017)	0.759 (0.015)	0.760 (0.012)	0.758 (0.011)
	0.25	0.756 (0.022)	0.763 (0.017)	0.769 (0.015)	0.772 (0.013)	0.771 (0.011)	0.773 (0.010)
	0.30	0.764 (0.017)	0.767 (0.015)	0.772 (0.013)	0.777 (0.011)	0.778 (0.010)	0.780 (0.009)
	0.35	0.769 (0.019)	0.772 (0.014)	0.775 (0.012)	0.778 (0.009)	0.776 (0.009)	0.778 (0.008)
	0.40	0.765 (0.020)	0.768 (0.017)	0.775 (0.013)	0.777 (0.011)	0.781 (0.008)	0.779 (0.008)
	0.45	0.762 (0.021)	0.763 (0.019)	0.767 (0.014)	0.770 (0.013)	0.773 (0.011)	0.771 (0.010)
	0.50	0.748 (0.022)	0.755 (0.018)	0.757 (0.015)	0.762 (0.014)	0.761 (0.013)	0.765 (0.011)
	1.00	0.708 (0.027)	0.716 (0.023)	0.724 (0.019)	0.729 (0.018)	0.728 (0.016)	0.731 (0.014)
	1.50	0.670 (0.036)	0.677 (0.029)	0.688 (0.024)	0.695 (0.022)	0.693 (0.018)	0.698 (0.016)
Linear	0.05	0.645 (0.035)	0.652 (0.028)	0.660 (0.025)	0.661 (0.023)	0.657 (0.020)	0.664 (0.018)
	0.10	0.682 (0.033)	0.691 (0.029)	0.701 (0.023)	0.704 (0.019)	0.707 (0.017)	0.705 (0.015)
	0.15	0.707 (0.031)	0.716 (0.026)	0.718 (0.021)	0.726 (0.018)	0.723 (0.016)	0.730 (0.014)
	0.20	0.724 (0.026)	0.735 (0.022)	0.741 (0.018)	0.747 (0.016)	0.747 (0.014)	0.746 (0.012)
	0.25	0.742 (0.024)	0.751 (0.019)	0.757 (0.016)	0.761 (0.014)	0.760 (0.012)	0.763 (0.011)
	0.30	0.751 (0.019)	0.755 (0.016)	0.760 (0.014)	0.765 (0.012)	0.766 (0.011)	0.768 (0.010)
	0.35	0.760 (0.021)	0.765 (0.016)	0.769 (0.014)	0.772 (0.011)	0.771 (0.010)	0.773 (0.009)
	0.40	0.752 (0.021)	0.755 (0.018)	0.760 (0.015)	0.764 (0.012)	0.767 (0.010)	0.766 (0.009)
	0.45	0.746 (0.022)	0.749 (0.020)	0.754 (0.015)	0.757 (0.014)	0.760 (0.012)	0.759 (0.011)
	0.50	0.732 (0.023)	0.741 (0.019)	0.744 (0.016)	0.749 (0.015)	0.748 (0.014)	0.752 (0.012)
	1.00	0.695 (0.028)	0.703 (0.024)	0.711 (0.020)	0.716 (0.019)	0.715 (0.017)	0.718 (0.015)
	1.50	0.657 (0.037)	0.664 (0.030)	0.675 (0.025)	0.682 (0.023)	0.681 (0.019)	0.686 (0.017)

For applications requiring finer calibration, the regularization parameters may alternatively be selected via cross-validation. Specifically, to avoid extensive grid search, we unified the four sparsification parameters as a single λ_0 based on theoretical analysis indicating their identical asymptotic order. Certainly, should that be necessary, we could alternatively assume distinct parameters for the four components and conduct a grid search. We then implemented a five-fold cross-validation procedure: datasets were partitioned into training and validation sets, where models were trained using objective functions (3) and (4) to obtain $f_{\alpha_j}(\boldsymbol{x}_{i,-j})$ for $j = 1, \cdots, p$. The validation loss $L = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} \left(x_{i,j} - f_{\alpha_j}(\boldsymbol{x}_{i,-j})\right)^2$ was evaluated across candidate λ_0 values $(0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.5, 1.0)$, with the minimizer selected as optimal. Performance results are summarized in Table 6. This approach yielded slightly improved F1 scores compared to our original results but incurred higher computational complexity.

A.4.3 Selection of the Hyperparameters

Regarding the hyperparameters of the neural network, we evaluated different configurations using two simulated datasets. Specifically, we investigated the effect of the number of hidden layers: (2, 3, 5, 8) and the number of units per hidden layer: (25, 50, 100). The F1-score results (summarized in Table 7) indicate minimal performance differences across configurations. To optimize computational efficiency while maintaining competitive performance, and in line with established practices in the field, we ultimately set the network architecture to 3 hidden layers with 50 units per layer.

Table 6: Performance of NNBLNet with the optimal λ_0 identified by five-fold cross-validation across six datasets (mean values and standard deviations for synthetic datasets).

Dataset	Recall	Precision	F1
Nonlinear	0.878 (0.016)	0.702 (0.021)	0.776 (0.014)
Linear	0.893 (0.017)	0.695 (0.019)	0.781 (0.013)
Friendship	0.888	0.820	0.853
Co-authorship	0.732	0.698	0.715
BRCA	0.779	0.634	0.700
LUAD	0.658	0.545	0.596

Table 7: F1 score of NNBLNet with different hyperparameters under linear and nonlinear scenarios

Layers	Units	Scenario Type	
		Nonlinear	Linear
2	25	0.743(0.019)	0.739(0.018)
	50	0.755(0.018)	0.751(0.017)
	100	0.748(0.018)	0.742(0.018)
3	25	0.765(0.016)	0.760(0.015)
	50	0.772(0.014)	0.769(0.013)
	100	0.768(0.016)	0.765(0.015)
5	25	0.769(0.018)	0.765(0.017)
	50	0.773(0.015)	0.770(0.014)
	100	0.769(0.017)	0.766(0.017)
8	25	0.768(0.019)	0.764(0.018)
	50	0.771(0.016)	0.768(0.016)
	100	0.765(0.018)	0.762(0.018)

A.4.4 Computation Cost

All experiments were conducted on a workstation equipped with an Intel Core i7-800H Processor, an Nvidia Tesla A40 GPU, and 64GB of RAM. Table 8 compares the computational time and peak memory usage of our method against the baselines across varying values of n and p. To ensure a fair comparison, all methods were executed on a single CPU core. The results indicate that Fair Glasso is the most computationally efficient method by a significant margin. In contrast, NNBLNet, BGSL, and DeepGRNCS involve substantially computational costs, which escalate with the number of nodes. This higher cost is attributed to the iterative training of neural networks and, for BGSL, the additional overhead from MCMC sampling. Although NNBLNet is slower than the linear-based methods, it is faster than the other neural network-based approach, DeepGRNCS. Overall, NNBLNet demonstrates favorable scalability.

A.5 Experiment Details

A.5.1 Synthetic Data Setting

We simulated a dataset with $n = 1000$ samples and $p = 100$ variables. The 100 nodes (variables) were partitioned into $L = 10$ groups (blocks), each containing 10 nodes. For each block, we generated a Barabási-Albert network structure with a maximum node degree of 4.

Connections were introduced between each pair of adjacent groups, i.e., group 1 is connected to group 2, group 3 is connected to group 4, $\cdots$, and group 9 is connected to group 10. For each such pair, node-level links were established such that the i-th node in the first group was connected to the i-th node in the second group (e.g., x_1 to x_{11}, x_2 to x_{12}, etc.).

Within each group, the features were generated using a recursive formula inspired by the Barabási-Albert network property. First, nodes were sorted by degree in descending order: $j_1, \ldots, j_{10}$. For the node j_1 with the largest degree, x_{ij_1} was generated from a standard normal distribution. For

Table 8: Average computation time (in minutes, the first value in parentheses) and peak memory usage (MB, the second value in parentheses) of different methods across various sample sizes (n) and dimensions (p)

n	p	NNBLNet	BGSL	Fair Glasso	DeepGRNCS
500	50	(9.5, 61.8)	(6.3, 61.8)	(0.0, 30.3)	(15.7, 240.4)
	100	(21.8, 307.1)	(11.0, 90.3)	(0.0, 45.1)	(49.6, 1120.7)
	200	(46.2, 709.5)	(14.5, 149.2)	(0.1, 69.1)	(100.9, 4490.8)
1000	50	(12.2, 158.2)	(8.1, 65.7)	(0.0, 31.4)	(20.1, 338.8)
	100	(28.0, 308.9)	(14.1, 91.1)	(0.1, 45.7)	(63.6, 1148.2)
	200	(59.2, 719.6)	(18.6, 152.3)	(0.1, 70.2)	(129.4, 4602.9)
2000	50	(15.6, 179.7)	(10.4, 70.1)	(0.2, 34.2)	(25.7, 399.2)
	100	(35.8, 428.4)	(18.0, 98.0)	(0.3, 52.2)	(79.0, 1652.6)
	200	(75.8, 929.2)	(23.8, 171.9)	(0.4, 89.9)	(150.0, 6397.4)
5000	50	(19.5, 215.8)	(13.0, 74.8)	(0.3, 36.6)	(32.2, 528.7)
	100	(44.8, 498.9)	(22.6, 109.2)	(0.5, 56.3)	(95.0, 1952.7)
	200	(94.7, 1102.5)	(29.8, 182.5)	(0.7, 101.8)	(185.0, 7598.6)
10000	50	(22.6, 241.0)	(15.0, 80.9)	(0.5, 38.5)	(37.2, 599.5)
	100	(51.8, 579.2)	(26.1, 114.7)	(0.8, 60.1)	(110.0, 2248.9)
	200	(109.5, 1247.9)	(34.4, 193.3)	(1.1, 109.7)	(215.0, 8799.1)
20000	50	(26.3, 257.6)	(16.8, 85.3)	(0.7, 39.8)	(42.4, 633.2)
	100	(57.3, 609.8)	(30.8, 127.2)	(1.0, 64.4)	(121.2, 2540.5)
	200	(123.9, 1335.2)	(39.7, 208.8)	(1.4, 117.9)	(242.2, 9008.7)

each subsequent node j_l ($l = 2, \ldots, 10$) with neighbors N_{j_l} in the same group, the feature x_{ij_l} was simulated as:

$$x_{ij_l} = \sum_{j_k \in N_{j_l}, k<l} f_k(x_{ij_k}) + \varepsilon_{ij_l}, \tag{42}$$

where for the nonlinear case: $f_k(\cdot)$ incorporated polynomial term x^3, interaction term $0.5x_1x_2$, $0.2\exp x$, and $\sin x$, and for the linear case: $f_k(x) = z \cdot x$ with $z \sim \mathcal{N}(2, 1)$. The error term ε_{ij_l} was generated from $\mathcal{N}(0, 0.01)$.

For each connected group pair (e.g., group 1 and group 2), we first generated the features of the first group (e.g., group 1) using the procedure described above. Then, for the second group (e.g., group 2), we generated each node's feature using a bi-level formulation. Specifically, we first generated a group-internal signal $x'_{ij_l} = \sum_{j_k \in N_{j_l}, k<l} f_k(x_{ij_k})$, then incorporated a signal from the connected node in the previously generated group, $g(x)$, along with a group-connection coefficient $\theta^* \sim \mathcal{N}(5, 1)$ as follows:

$$x_{ij_l} = \theta^* \left(x'_{ij_l} + g(x_{ij'})\right) + \varepsilon_{ij_l}, \tag{43}$$

where j' denotes the index of the connected node in the first group, and $g(x) = x^3$ for the nonlinear case and $g(x) = x$ for the linear case.

In the synthetic data generated by the aforementioned procedure, inter-group connections accounted for 37.0% of the total network edges. To better visualize the connectivity patterns, we quantified the network's structural properties using two metrics. The intra-group density of a group C_k is defined as the proportion of observed edges among all possible edges within the group:

$$\text{IntraDensity}(C_k) = \frac{2 \cdot \sum_{i<j,\, i,j \in C_k} A_{ij}}{|C_k|(|C_k| - 1)}.$$

Similarly, the inter-group density between two distinct groups C_k and C_l ($k \neq l$) is given by the proportion of observed edges across the groups:

$$\text{InterDensity}(C_k, C_l) = \frac{\sum_{i \in C_k,\, j \in C_l} A_{ij}}{|C_k| \cdot |C_l|}.$$

For the synthetic network, the mean intra- and inter-group densities were 37.8% and 10.0%, respectively. These metrics help characterize the strength of connections within and between groups, providing a quantitative basis for analyzing the network structure.

For both linear and nonlinear patterns, the performance was assessed over 100 independent simulation replicates, with results for recall, precision, and F1-score summarized by their mean and standard deviation.

A.5.2 Real-world Datasets

Table 9 summarizes the key features of the real-world datasets. Additional details are provided below.

Table 9: Summary of real-world datasets.

Dataset	Nodes (No.)	Edges (No.)	Samples (No.)	Groups (No.)
Friendship	Students (311)	1009	Interactions (47127)	Gender (2)
Co-authorship	Authors (130)	525	Keywords (1903)	Publication type (6)
BRCA	mRNA (73)	763	Patients (1099)	Pathway (4)
LUAD	CNA (98)	700	Patients (507)	Pathway (13)

• **Friendship**. This network dataset captures social interactions among students across nine classes at a high school in Marseille, France, recorded over five consecutive days in December 2013. Following standard practice in contact network analysis, we constructed a ground-truth graph where nodes correspond to students and weighted edges reflect aggregated interaction frequencies. In line with [25], node attributes were assigned based on gender, and the signals were generated by grouping the interactions into sets of four. The Friendship dataset is available at `http://www.sociopatterns.org/datasets/high-school-contact-and-friendship-networks/`.

• **Co-authorship**. This network dataset originates from ACM conference proceedings and includes 17,431 unique authors, 122,499 publications, and 1,903 technical keywords. We focused on a representative subset of authors, where nodes correspond to individual researchers. Demographic attributes were assigned according to authors' predominant conference categories, determined by their maximum publication frequency. The ground-truth network was constructed through co-authorship detection: edges were added between authors who co-published at least one paper. To generate network signals, we quantified authors' keyword usage patterns by calculating normalized frequencies of specific technical terms across their publications. The Co-authorship dataset is available at `https://dl.acm.org/`.

• **BRCA**. The BRCA dataset is derived from The Cancer Genome Atlas (TCGA) and comprises mRNA gene expression profiles from 1,099 breast cancer patients. We selected 73 genes involved in four key biological pathways: B Cell Receptor Complexes, Caspase Cascade, G1 And S Phases, and MMP Cytokine Connection, based on prior domain knowledge and pathway annotations from the KEGG database [16]. Nodes in the network represent these genes, and group structure is defined by their pathway annotations [42]. A biologically grounded reference network was constructed using curated interaction data from the STRING database [35], which integrates multiple evidence sources such as experimental data, co-expression, and pathway information. An undirected edge was placed between two genes if a high-confidence interaction was reported in STRING, reflecting known regulatory or functional associations. For network estimation, each patient's expression profile was treated as an input signal across the 73 genes.

• **LUAD**. The LUAD dataset is also sourced from TCGA and contains copy number alteration (CNA) profiles for 507 lung adenocarcinoma patients. We focused on 98 CNAs implicated in 13 distinct biological pathways (see Figure 3 for details), with pathway annotations obtained from the KEGG database, offering a more complex grouping structure compared to BRCA. Each CNA is treated as a node, and group membership is determined by pathway assignment. As with BRCA, the reference network was constructed using functional interaction information from the STRING database. Each patient's CNA profile was used as input for structure learning, with the goal of recovering sparse and modular dependencies.

BRCA and LUAD expression data were obtained from the R package *cdgsr*, pathway information from the KEGG database was obtained using *msigdbr*, and interaction information from STRING was

obtained via *STRINGdb*. The structures of the four benchmark networks reveal distinct connectivity patterns. Specifically, the proportion of inter-group connections is 47.9% (Friendship), 36.2% (Co-authorship), 60.8% (BRCA), and 74.4% (LUAD), while the corresponding intra- and inter-group density pairs are (37.8%, 10.0%); (2.5%, 2.0%); (11.1%, 3.2%); and (53.7%, 20.9%). These metrics collectively highlight the structural heterogeneity across the datasets.

A.6 Sensitive Analysis

A.6.1 Generalization Capability

To evaluate the generalization capability of the proposed method, we assessed the stability of its network estimates. We repeatedly drew 90% subsets of the data from two synthetic and four real-world datasets and re-estimated the networks 100 times. The Jaccard index between each re-estimated network and the original full-dataset network was then calculated. The resulting indices (mean and standard deviation) were as follows: Nonlinear: 0.913 (0.052), Linear: 0.932 (0.047), Friendship: 0.889 (0.061), Co-authorship: 0.921 (0.055), BRAC: 0.902 (0.058), and LUAD: 0.874 (0.064). These consistently high values indicate that the proposed method yields stable network structures across diverse datasets, confirming its strong generalization capability.

A.6.2 Performance across Varying Sample Sizes and Group Sizes

To assess the robustness of our method to sample and group size specifications, we evaluated its performance under varying sample sizes ($n = 500, 1000, 2000, 5000$) with a fixed group size of ten, and under varying group sizes (ranging from 5-10, 5-20, and 5-30 individuals) with a fixed sample size of 1000. For groups containing 20 or 30 members, the network structure was simplified to a star module with node 1 as the hub connected to all other nodes. Based on the balanced baseline setting (with a ratio of group size 1:1), these configurations reflected increasing levels of group size imbalance, with approximate ratios of 2:1, 4:1 and 6:1, respectively. The last case represents the most pronounced disparity, where the largest group is six times larger than the smallest.

Results are summarized in Tables 10 and 11. Consistent with expectations, larger sample sizes enhanced the network recovery accuracy of all methods; nevertheless, NNBLNet consistently demonstrated superior performance. According to Table 11, NNBLNet maintained highly stable performance across moderate group size imbalances, with only a slight decrease in recall, precision, and F1 score observed under the most extreme 6:1 condition.

Table 10: Performance comparison of different methods for the nonlinear synthetic network with varying sample sizes: Mean (SD) over 100 replicates.

n	Method	Recall	Precision	F1
500	NNBLNet	0.861 (0.022)	0.679 (0.027)	0.769 (0.019)
	Fair Glasso	0.756 (0.021)	0.639 (0.026)	0.693 (0.020)
	BGSL	0.764 (0.022)	0.631 (0.024)	0.692 (0.019)
	DeepGRNCS	0.782 (0.021)	0.662 (0.023)	0.717 (0.018)
1000	NNBLNet	0.872 (0.016)	0.693 (0.022)	0.772 (0.014)
	Fair Glasso	0.779 (0.014)	0.653 (0.021)	0.710 (0.017)
	BGSL	0.790 (0.017)	0.644 (0.018)	0.709 (0.013)
	DeepGRNCS	0.809 (0.016)	0.681 (0.018)	0.731 (0.017)
2000	NNBLNet	0.879 (0.014)	0.707 (0.019)	0.775 (0.012)
	Fair Glasso	0.787 (0.013)	0.660 (0.018)	0.717 (0.015)
	BGSL	0.795 (0.015)	0.652 (0.017)	0.717 (0.014)
	DeepGRNCS	0.822 (0.014)	0.695 (0.017)	0.752 (0.015)
5000	NNBLNet	0.887 (0.011)	0.716 (0.015)	0.778 (0.009)
	Fair Glasso	0.795 (0.011)	0.667 (0.015)	0.725 (0.012)
	BGSL	0.802 (0.012)	0.660 (0.014)	0.724 (0.011)
	DeepGRNCS	0.838 (0.011)	0.708 (0.014)	0.767 (0.012)

A.6.3 Performance when Group Label Misclassification

Even when genuine group labels (e.g., pathway information) are available, misclassification may still occur and impair model performance. To assess the impact of such label errors, we randomly

Table 11: Performance comparison of different methods for the nonlinear synthetic network with varying group imbalance ratios: Mean (SD) over 100 replicates.

Ratio	Method	Recall	Precision	F1
1:1	NNBLNet	0.872 (0.016)	0.693 (0.022)	0.772 (0.014)
	Fair Glasso	0.779 (0.014)	0.653 (0.021)	0.710 (0.017)
	BGSL	0.790 (0.017)	0.644 (0.018)	0.709 (0.013)
	DeepGRNCS	0.809(0.016)	0.681(0.018)	0.731(0.017)
2:1	NNBLNet	0.874 (0.017)	0.691 (0.023)	0.771 (0.015)
	Fair Glasso	0.773 (0.015)	0.649 (0.022)	0.706 (0.018)
	BGSL	0.782 (0.018)	0.640 (0.020)	0.704 (0.015)
	DeepGRNCS	0.802 (0.017)	0.670 (0.019)	0.730 (0.016)
4:1	NNBLNet	0.869 (0.018)	0.688 (0.024)	0.769 (0.016)
	Fair Glasso	0.766 (0.016)	0.642 (0.023)	0.701 (0.018)
	BGSL	0.770 (0.019)	0.635 (0.021)	0.698 (0.016)
	DeepGRNCS	0.812 (0.017)	0.684 (0.019)	0.734 (0.016)
6:1	NNBLNet	0.854 (0.020)	0.675 (0.026)	0.753 (0.017)
	Fair Glasso	0.755 (0.018)	0.634 (0.024)	0.692 (0.019)
	BGSL	0.756 (0.020)	0.627 (0.023)	0.687 (0.017)
	DeepGRNCS	0.808 (0.018)	0.677 (0.020)	0.727 (0.017)

scrambled node group labels in simulated data under nonlinear setting, using misclassification rates of 10%, 20%, and 30%, with each scenario repeated 100 times. We evaluated three group-aware methods—NNBLNet, Fair Glasso, and BGSL—under these conditions. As summarized in Table 12, all three methods exhibit performance degradation as label error increases. Nonetheless, NNBLNet maintains relatively higher F1 scores and consistently outperforms both BGSL and Fair Glasso, demonstrating its robustness to moderate levels of label misclassification in real-world applications.

Table 12: Performance comparison of different methods for the nonlinear synthetic network with varying levels of group label misclassification: Mean (SD) over 100 replicates.

Error rate	Method	Recall	Precision	F1
0%	NNBLNet	0.872 (0.016)	0.693 (0.022)	0.772 (0.014)
	Fair Glasso	0.779 (0.014)	0.653 (0.021)	0.710 (0.017)
	BGSL	0.790 (0.017)	0.644 (0.018)	0.709 (0.013)
10%	NNBLNet	0.821 (0.022)	0.598 (0.029)	0.691 (0.019)
	Fair Glasso	0.756 (0.023)	0.564 (0.031)	0.662 (0.020)
	BGSL	0.742 (0.024)	0.553 (0.032)	0.651 (0.021)
20%	NNBLNet	0.765 (0.026)	0.541 (0.034)	0.633 (0.022)
	Fair Glasso	0.702 (0.027)	0.509 (0.035)	0.602 (0.023)
	BGSL	0.688 (0.028)	0.496 (0.036)	0.588 (0.024)
30%	NNBLNet	0.702 (0.031)	0.489 (0.039)	0.574 (0.026)
	Fair Glasso	0.648 (0.032)	0.455 (0.040)	0.540 (0.027)
	BGSL	0.632 (0.033)	0.442 (0.041)	0.526 (0.028)

A.7 Downstream Analysis of the LUAD dataset

Based on the low-level network inferred by NNBLNet, the top three hub genes with the highest degrees are TP53, JUN, and CD44. All three are strongly supported by existing literature in the context of lung adenocarcinoma (LUAD) and lung cancer biology. Specifically, TP53 is one of the most frequently mutated tumor suppressor genes in LUAD, with mutation rates often exceeding 40–50%. Its loss or mutation contributes to genomic instability, aggressive tumor behavior, and poor prognosis in lung cancer [3]. JUN has been identified as a gene associated with response to PD-1 blockade, suggesting its potential as a biomarker for immunotherapy efficacy in non-small cell lung cancer [37]. CD44, a recognized cell surface glycoprotein and cancer stem cell marker, is commonly used to identify stem-like subpopulations in lung cancer [12].

Based on NNBLNet analysis, the most significant pathway-level connection was observed between "MMP Cytokine Connection" and "IL4 Receptor in B Lymphocytes", with extensive gene

connectivity indicating functional cross-talk. IL-4 experimentally regulates MMP expression; for instance, it suppresses IL-1–induced MMP-3 transcription in human fibroblasts by inhibiting AP-1 promoter binding [31]. In LUAD, MMPs are markedly overexpressed and contribute to ECM remodeling, tumor invasion, and metastasis [43], while IL-4 signaling modulates immune activity and tumor–microenvironment crosstalk [17]. This supports a model wherein IL-4/IL-4R signaling influences ECM dynamics in LUAD by modulating MMP pathway activity, potentially via transcriptional regulation. As shown above, the analysis further validates that the NNBLNet method can yield biologically meaningful networks.

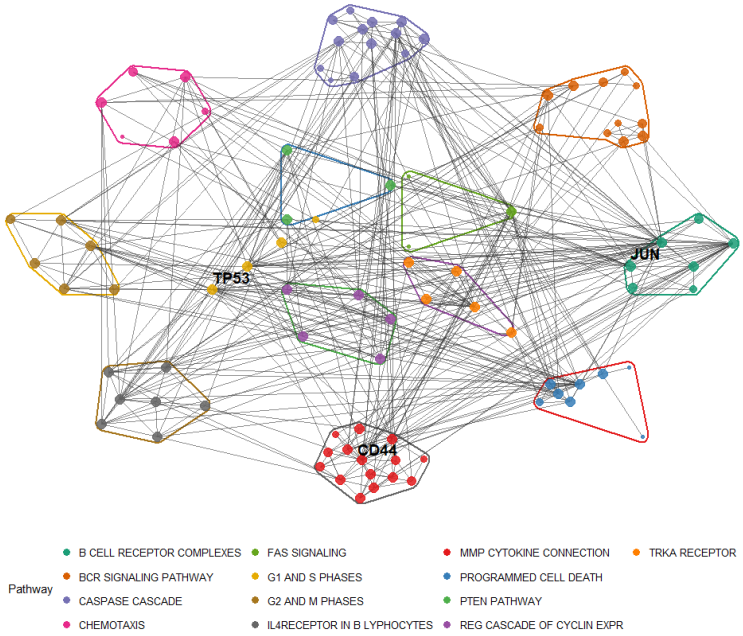

Figure 3: Network reconstruction of the LUAD dataset using NNBLNet. Nodes are color-coded by annotated pathways, and the top three hub genes are labeled.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 1 summarizes our main contributions, experimental and theoretical results are presented in Section 3 and Section 5 respectively.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are discussed in Section 6.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The assumptions and theories are provided in Section 3 and a complete proof is provided in Appendix A.3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 provides a concise overview of our experimental setup, while Appendix A.5 contains a comprehensive description. The implementation code is included in the supplementary material and is publicly accessible on GitHub at `https://github.com/mengyunwu2020/NNBLNet`.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The supplementary material includes the code used for the experiments presented in this paper. Links to all publicly available datasets, as well as instructions for generating the synthetic datasets used, are provided in Appendix A.5.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The hyperparameter choices and optimization algorithm are described in Appendix A.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The results of the synthetic simulations reported in Sections 5.1 are averaged over 100 independent trials, with both the mean and standard error computed. However, we observed consistently low variance across these realizations, and thus omit error bars in the plots for visual clarity. In Section 5.2 , the experiments are conducted on real-world datasets using all available samples and statistical significance tests are not applicable.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide a summary of the compute resources used for the experiments in Appendix A.4.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impacts of this work are discussed in Section 6.2.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The datasets used in the experiments are properly cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.