Hidden Markov Networks for Heterogeneous Graph Structure Learning

Anonymous authors
Paper under double-blind review

Abstract

An important task in graph machine learning is inferring the graph structure (when it is not readily available) from observed data such that it captures well the intrinsic relationships between the data entities. Such a task relies on constructing a data-generating process capable of modeling the interaction between node features and the underlying graph structure. While significant advancements have been made in graph learning technologies for homogeneous graphs, many real-world graphs exhibit heterogeneous patterns where nodes and edges have multiple types. This poses a challenge in graph learning given the difficulty of modeling the interaction between node features, nodes/edge types, and graph structure in heterogeneous graphs with a proper generative process. To this end, this paper introduces the first approach for heterogeneous graph structure learning (HGSL) from a probabilistic perspective. We first propose a novel statistical model for the data-generating process of heterogeneous graph features, namely hidden Markov networks for heterogeneous graphs (H2MN), by accounting for the heterogeneity of nodes and edges. We then formalize the problem of HGSL as a maximum a-posterior estimation problem parameterized by such a model, and derive an alternating optimization method to obtain a solution together with a theoretical justification of the optimization conditions. Finally, we conduct extensive experiments on both synthetic and real-world datasets, which demonstrate that our proposed method consistently outperforms the baselines in terms of edge identification and edge weight recovery.

1 Introduction

Graphs are a powerful and ubiquitous representation of complex relational data, which is capable of capturing pairwise relationships between entities. Graphs occur frequently in real datasets such as social networks Salami et al. (2017), e-commerce Wang et al. (2019) and financial transactions Liu et al. (2018), in addition to various scientific and technological areas. However, a meaningful graph is not always readily available from the data Dong et al. (2015), and sometimes the graph observed is not always clean Chen et al. (2015). Inferring the underlying graph structure from observed signals living in the nodes is, therefore, an important problem in the field of both graph machine learning (GML) Chami et al. (2022); Hamilton et al. (2017) and graph signal processing (GSP) Dong et al. (2019), Dong et al. (2020); Ortega et al. (2018). In GML models, e.g., graph neural networks, graph structure learning (GSL) is typically plugged in as an extra component to refine the graph topology, and empower the prediction ability of models Zaripova et al. (2023), Battiloro et al. (2023). In GSP, graph structure learning serves as a crucial technique to assist in exploring the intrinsic relationships between entities, and make many algorithms possible to work, such as spectral clustering Dong et al. (2015), cooperation game Rossi et al. (2022), graph anomaly detection, etc.

Depending on whether a graph is inferred with or without extra supervision signals such as downstream tasks, graph structure learning (GSL) methods are divided into two classes - model-based and learning-based GSL (add citations). Both Graph Structure Learning Dong et al. (2015); Pu et al. (2021a); Kalofolias (2016a); Pu et al. (2021b) assume that the features/signals on the graphs should admit certain regularity or smoothness, which is modelled by an underlying data-generating process. In homogeneous graphs, the common choices of the data-generating process are Ising models Ising (1925), Gaussian Graphical Models Yuan & Lin (2007)

or Pair-wise Markov Random Fields (Markov Networks Park et al. (2017)), where the features are assumed to be emitted from a colored Gaussian distribution that is uniquely determined by the graph structure. Then the graph structure learning problem is solved by the inverse covariance (precision) matrix estimation on the graph features Yuan & Lin (2007); Friedman et al. (2008); Banerjee et al. (2008); Ravikumar et al. (2008). The following studies improve from the optimization problem formalized by the precision matrix estimation to directly obtain the graph Laplacian matrix Dong et al. (2015), guarantee a better valid graph structure Kalofolias (2016b) and derive relaxed optimization for better convergence Egilmez et al. (2017).

Despite the success of the aforementioned methods in a variety of homogeneous graph structure learning tasks where nodes and edges belong to a single type, there remains a significant lack of insights on how to solve graph structure learning tasks in heterogeneous graphs. Real-world graphs often exhibit heterogeneous patterns Wang et al. (2020), with multiple types of nodes and edges representing different kinds of entities and relationships. For instance, in a network representing a recommender system Fu et al. (2020), nodes can have distinct types (e.g. users and items), and edges can represent different types of relations (e.g. like/dislike for user-item edges or following/being followed for user-user edges). Other examples include social networks, academic networks Lv et al. (2021); Gao et al. (2009), and knowledge graphs Bollacker et al. (2008); Dettmers et al. (2018), Mahdisoltani et al. (2015).

The design of the data-generating process for heterogeneous graphs requires proper modeling of the interaction among the graph structure and the node features from different types of nodes/edges. To this end, we extend the pair-wise Markov Random Field Park et al. (2017) to the scenario of heterogeneous graphs by considering the potential function depending on the node and edge types. This potential function is rooted in an assumption that the features of nodes in a hyperedge are highly correlated by the features of the hyperedge connecting them; See Figure for the illustration of a hypergraph obeying this assumption. In this design, the feature distribution of nodes and hyperedges on a hypergraph is modelled as a colored multivariate Gaussian distribution whose covariance matrix is uniquely determined by the given hypergraph structure. To further alleviate the problem of missing hyperedge features, we show that the summation of node-hyperedge potential function is bounded by the summation of node-node potential function, thus an estimator is developed through an approximation of the node-wise potential. This estimator is defined as the sum of the maximum l_2 distances between any two node features in each hyperedge of a hypergraph.

Such a point of view is helpful as it allows one to justify why a particular clustering algorithm design is deemed suitable based on the statistical properties of the data, i.e., the structures and randomness inherent in random graph models. Existing GSL methods are not well-suited for heterogeneous graphs, as the smoothness assumption does not generalize well when diverse node and edge types are present due to the following *limitations*: 1) All nodes and edges are treated equally without considering the different node and/or edge types that may affect the way the edges are formed; 2) In classical GSL frameworks all signal dimensions are considered contributing to the smoothness prior of the signal while in practice this may not be the case. We refer to appendix D.7 for further details on limitations and motivation.

This paper aims to address the problem of heterogeneous graph structure learning (HGSL) through the following contributions.

1. **Design of data-generating process for heterogeneous graphs.** We first develop a novel generative model for the heterogeneous graph signals.
2. **Problem Formulation** We then solve the HGSL problem through the maximum a-posterior (MAP) estimation of the adjacency tensor that captures the structure of the HG with different node/edge types.
3. **Efficient Algorithm.** We inherently interpret the regularity that an HG should follow to make the algorithm work through the parameters of our proposed generative models.
4. **Real-world Applications.** Finally, extensive experiments were conducted to prove the efficacy of our algorithm.

Our method generalizes GSL approaches to heterogeneous graphs and is applicable to a wide range of graphs, including bipartite, multi-relational, and knowledge graphs.

2 Related Work

Broadly speaking there are two methods to solve the problem of graph structure learning - model-based and learning-based approaches.

Model-based Graph Structure Learning The model-based methods learn the graph structure by solving a regularized convex optimization, whose objective function reflects the inductive bias on the structure-data interactions and the regularizers encode the structure prior Banerjee et al. (2008); Lu (2010); Friedman et al. (2008); Lake & Tenenbaum (2010); Slawski & Hein (2015); Egilmez et al. (2017); Deng & So (2020); Kumar et al. (2019a); Kalofolias (2016a); Dong et al. (2015); Berger et al. (2020). The initial attempts of such methods live in the context of probabilistic graphical models, where the . solve an ℓ_1 regularised log-likelihood maximisation to obtain a sparse precision matrix Banerjee et al. (2008); Lu (2010); Friedman et al. (2008). Precision matrices are further restricted to Laplacian matrices Lake & Tenenbaum (2010); Slawski & Hein (2015); Egilmez et al. (2017); Deng & So (2020), while more complex structural priors such as community structure are explored Kumar et al. (2019a). In the field of graph signal processing, graph signals (data on nodes) can be seen as outcomes of diffusion processes on graphs and reconstruct a graph from signals according to the diffusion model Thanou et al. (2017); Shafipour et al. (2017); Segarra et al. (2017). The authors in Kalofolias (2016a); Dong et al. (2015); Berger et al. (2020) formulate the graph-data fidelity term using the quadratic Laplacian term which measures global smoothness of graph signals. Since structural priors such as sparsity and non-negative edge weights are non-differentiable, They often rely on iterative algorithms which require specific designs, including proximal gradient descent Rolfs et al. (2012), alternating direction method of multiplier (ADMM) Boyd et al. (2011), block-coordinate decent methods Friedman et al. (2008) and primal-dual splitting Kalofolias (2016b). Our model unrolls a primal-dual splitting as an inductive bias, and it allows for regularisation on both the edge weights and node degrees.

Leanring-based Graph Structure Learning Recently, learning-based methods have been proposed to introduce more flexibility to model structural priors in graph learning. GLAD Shrivastava et al. (2020) unrolls an alternating minimisation algorithm for solving an ℓ_1 regularised log-likelihood maximisation of the precision matrix. It also allows the hyperparameters of the ℓ_1 penalty term to differ element-wisely, which further enhances the representation capacity. Deep-Graph Belilovsky et al. (2017) uses a CNN architecture to learn a mapping from empirical covariance matrices to the binary graph structure. We will show in later sections that a lack of task-specific inductive bias leads to an inferior performance in recovering structured graphs.

Data-generating Process for Graphs (DGPG). The pair-wise exponential Markov random field (PE-MRF) is widely used in modeling the stochastic dynamics of the overall system. It can capture the high-dimensional feature generation of nodes on a graph, where the node features can either be discrete labels $\mathbf{Y}$ or continuous signals $\mathbf{X}$. The former is known as Ising models Ising (1925)) and the second is known as Gaussian Graph Models Yuan & Lin (2007). Moreover, if the random variables $\mathbf{X}$ and $\mathbf{Y}$ are thought to have a hierarchy dependency structure as in fig. 1, the model names hidden Markov networks (or hidden Markov random fields) Liu et al. (2014). Another highly related area is statistical random graph model, e.g. random graph models cite a few, contextual stochastic blocking models **????**. Though they could share a highly similar PDF, DGM and random graph models are fundamentally different in the sense of generation sequence, which we left the discussion in appendix D

Heterogeneous Graph Structure Prior The inductive bias can be injected into the model by multiple ways, for example, from the loss function, the architectural design, training methods, etc. Our method is inspired by such a group of priors. In Heterogeneous graph representation learning, the node and edge embeddings are trained through a relation-aware function. The relational graph convolution networks Schlichtkrull et al. (2018) modify the message passing to be relation-wise, which assumes a smoothness concerning each relation type. Wang et al. (2022) consider a relation-wise smoothness to design multi-relational Graph neural networks. Relational Markov Networks Taskar et al. (2007) and introduce a probabilistic formulation for the knowledge base systems, however, such a work considers the relation types solely as an entity and fail to reflect the pair-wise connection on the nodes. This

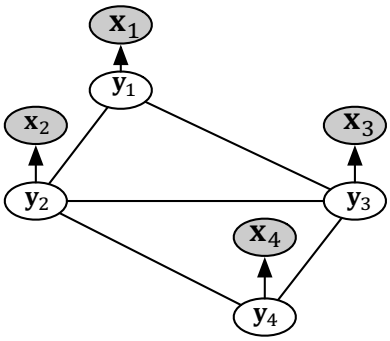


Figure 1: The graphical models for Hidden Markov Networks

3 Preliminaries

Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{W}\}$ denote a graph with edges $\mathcal{E}$ and nodes $\mathcal{V}$. The graph structure is represented by the weighted matrix $\mathbf{W} = \{w_{uv}\}_{u,v\in\{1:N\}} \in \mathbb{R}_+^{N\times N}$ and the graph Laplacian matrix is further defined as $\mathbf{L} = \mathbf{D} - \mathbf{W} \in \mathbb{R}^{N\times N}$, with $\mathbf{D} = \mathrm{diag}(\mathbf{W1})$ being the degree matrix and $d_v \equiv \mathbf{D}_{vv}$. Each node $v \in \mathcal{V}$ is associated with a series of node features, where the features can either be a signal vector $\mathbf{x}_v \in \mathbb{R}^K$ with K as the signal dimension or a label for its class, represented by a one-hot vector $\mathbf{y}_v \in \mathbb{R}^C$ with C the number of classes. It is possible to stack the feature vector on each node together and obtain $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times K}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N]^T \in \mathbb{R}^{N\times C}$.

3.1 Data-generating Process of Graph Features

Our work focuses on generalizing hidden Markovian networks (HMN) Ghahramani (2001) to facilitate the learning of graph structure. HMN assumes the hidden variable $\mathbf{Y}$ to be generated from a colored Gaussian distribution whose covariance matrix is parameterized by the graph structure $\mathbf{W}$ and the signals $\mathbf{X}$ emitted from the hidden variables with a linear transformation. Considering a graphical model for feature generation as in fig. 1 and taking cliques up to rank 2, the joint probability distribution of $\mathbf{X}$ and $\mathbf{Y}$ is

$$\begin{aligned} P(\mathbf{X}, \mathbf{Y}; \mathbf{W}, \mathbf{V}) &= P(\mathbf{Y}; \mathbf{W}) P(\mathbf{X} \mid \mathbf{Y}; \mathbf{V}) \\ &= \frac{1}{Z(\mathbf{W})} \prod_v \psi_1(\mathbf{y}_v) \prod_{u,v} \psi_2(\mathbf{y}_v, \mathbf{y}_u; w_{uv}) \prod_{i=1}^{N} P(\mathbf{x}_v \mid \mathbf{y}_v; \mathbf{V}) \,. \end{aligned} \tag{1}$$

ψ_1 and ψ_2 are respectively the potential functions for cliques of order 1 (the nodes themselves) and order 2 (the pair-wise potential living on edges). $Z(\mathbf{W})$ is the partition function[1]. It is clear that a properly defined data-generating process for a graph consists of three components, 1) the node-wise potential $\psi_1(\cdot)$ defined on nodes $v \in \mathcal{V}$, 2) the edge-wise potential $\psi_2(\cdot,\cdot)$ defined over each edge $\{u, v\} \in \mathcal{E}$, and 3) the emission process $P(\mathbf{x}_v \mid \mathbf{y}_v; \mathbf{V}), \forall v$.

Tracing back to Zhu et al. (2003) and Lake & Tenenbaum (2010), the node-wise and pair-wise potential are selected to be $\psi_1(\mathbf{y}_v) = \exp(-(d_v + \nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2)$ with $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm and $\psi_2(\mathbf{y}_u, \mathbf{y}_v; w_{uv}) = \exp(w_{uv}\mathbf{y}_u^\top \mathbf{y}_v)$ respectively. The overall likelihood function for $\mathbf{Y}$ is then,

$$\begin{aligned} P(\mathbf{Y}; \mathbf{W}) &= \frac{1}{Z} \prod_v \psi_1(\mathbf{y}_v) \prod_{v,u} \psi_2(\mathbf{y}_v, \mathbf{y}_u) \\ &= \frac{1}{Z} \prod_v \exp(-(d_v + \nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2) \prod_{v,u} \exp(w_{uv}\mathbf{y}_u^\top \mathbf{y}_v) \end{aligned} \tag{2}$$

To transit from the hidden variable $\mathbf{Y}$ to $\mathbf{X}$, the emission function is considered a multi-variate Gaussian Liu et al. (2014) which has a linear relationship, specifically $\mathbf{x}_v \mid \mathbf{y}_v \sim \mathcal{N}(\mathbf{V}\mathbf{y}_v, \Sigma_{\mathbf{x}})$. We denote $\mathbf{V}$ as the

[1] In what follows we will simply use Z.

linear transformation matrix and Σ_x the marginal covariance matrix for $\mathbf{x}_v$, which is shared across all nodes $v \in \mathcal{V}$. To further simplify the notation, we make the first assumption that the determinant of covariance matrix on $\mathbf{X}$ marginal is relatively small to the colored multi-variate Gaussian distribution that emits $\mathbf{Y}$, i.e., $\det(\Sigma_{\mathbf{x}}) \ll \det(\Lambda^{-1})$. This assumption assures that the marginal noise on $\mathbf{x}$ is not so strong. Thus, by marginalizing out $\mathbf{y}_v$, we have the likelihood function as

$$P(\{\mathbf{x}_v\} \mid \mathbf{W}, \mathbf{V}) = \frac{1}{Z} \prod_v \exp(-(\nu + d_v)\|\mathbf{V}^\dagger \mathbf{x}_v\|_{\mathrm{F}}^2) \cdot \prod_{u,v} \exp\left\{w_{uv} \operatorname{tr}(\mathbf{x}_u^\top \mathbf{V}^{\dagger\top} \mathbf{V}^\dagger \mathbf{x}_v)\right\}, \tag{3}$$

where $\mathbf{V}^\dagger$ is the Moore–Penrose Pseudo inverse of matrix $\mathbf{V}$. We left the detailed derivation of the likelihood function in appendix A.1.

3.2 Model-based Graph Structure Learning

The hidden Markov networks in section 3.1 can be further used to infer the graph structure from the node features Park et al. (2017). Mathematically, graph structure learning can be considered as a Maximum A-posteriori (MAP) estimation problem for the graph structure $\mathbf{W}$ and the linear transformation matrix $\mathbf{V}$[2], where

$$\mathbf{W}^*, \mathbf{V}^{\dagger *} = \arg\max_{\mathbf{W}, \mathbf{V}^\dagger} \log P(\mathbf{W}, \mathbf{V}^\dagger \mid \mathbf{X}) = \arg\max_{\mathbf{W}, \mathbf{V}^\dagger} \log P(\mathbf{X} \mid \mathbf{W}, \mathbf{V}^\dagger) + \log P(\mathbf{W}) + \log P(\mathbf{V}^\dagger) \tag{4}$$

Current graph structure learning (GSL) algorithms Dong et al. (2015); Pu et al. (2021a); Kumar et al. (2019b) can be considered as a special case of the above MAP problem when $\mathbf{V}^\dagger$ is rectangular orthogonal (semi-orthogonal) Matrix. In such a scenario, the likelihood function of eq. (3) reduces to the following form,

$$P(\{\mathbf{x}_v\} \mid \mathbf{W}) = \frac{1}{Z} \prod_v \exp(-(\nu + d_v)\|\mathbf{x}_v\|_{\mathrm{F}}^2) \cdot \prod_{u,v} \exp\left\{w_{uv} (\mathbf{x}_u^\top \mathbf{x}_v)\right\}, \tag{5}$$

This equates to a Gaussian Graphical Model Yuan & Lin (2007) with precision matrix $\Lambda = \mathbf{L} + \nu\mathbf{I}$, as shown in appendix A.1. The only parameters left are the weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j \in \{1:N\}}$ representing the graph structure. Then the problem becomes, $\mathbf{W}^* = \arg\max_{\mathbf{W}} \log P(\mathbf{X} \mid \mathbf{W}) + \log P(\mathbf{W})$. The log-likelihood of the signal $\mathbf{X}$ is used for convenient optimization,

$$\begin{aligned}
\log\left[P\left(\mathbf{X} \mid \mathbf{W}\right)\right] &= \sum_{v=1}^{N} -(d_v + \nu)\|\mathbf{x}_v\|_{\mathrm{F}}^2 + \sum_{u,v} w_{uv} \mathbf{x}_u^\top \mathbf{x}_v - \log Z(\mathbf{W}) \\
&= -\nu \sum_{u=1}^{N} \|\mathbf{x}_u\|_{\mathrm{F}}^2 - \sum_{u,v} \mathbf{L}_{uj} \mathbf{x}_u^\top \mathbf{x}_v - \log Z(\mathbf{W}) \\
&= -\nu \operatorname{tr}((\mathbf{X}^\top \mathbf{X}) - \operatorname{tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) + \log\det(\mathbf{L} + \nu\mathbf{I}) + \text{Constant},
\end{aligned} \tag{6}$$

where the last step is derived given the only term related to the optimization goal in $Z(\mathbf{W})$ is the log-determinant of the covariance matrix. ν works as the factor balancing the node-wise and pair-wise effects. The pair-wise potentials are scaled by parameter $\mathbf{L}_{uv}$ which encodes the connection strength. In graph signal processing, the second term, in the quadratic form of the graph Laplacian, is called the "smoothness" of the signal on the graph Zhou & Schölkopf (2004), which can be measured in terms of pair-wise difference of node signals,

$$\operatorname{tr}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) = \frac{1}{2} \sum_{\{u,v\}} w_{uv} \|\mathbf{x}_u - \mathbf{x}_v\|_{\mathrm{F}}^2, \tag{7}$$

given that $\sum_v w_{uv} = \mathbf{D}_u$. In other words, if two signal vectors x_u and x_v from a smooth set reside on two well-connected nodes with large w_{uv}, they are expected to have a small dissimilarity.

However, the computationally demanding log-determinant term makes the optimization problem difficult to solve Kalofolias (2016a). Thus, researcher in Egilmez et al. (2017) consider a relaxed optimization on

[2] ν is considered as a hyperparameter.

the lower bound instead. Using $-\sum_v \log\left(\text{diag}(\Lambda)_v\right) \leq -\log\det(\Lambda)$ and substituting the log-likelihood with eq. (6), we can further derive the optimization objective as,

$$\begin{aligned}
&\arg\max_{\mathbf{W}} \log P(\mathbf{X} \mid \mathbf{W}) + \log P(\mathbf{W}) \\
\approx &\arg\max_{\mathbf{L}} -\nu \operatorname{tr}((\mathbf{X}^\top \mathbf{X}) - \operatorname{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) + \mathbf{1}^\top \log\left(\text{diag}(\mathbf{L} + \nu\mathbf{I})_v\right) - \log P(\mathbf{W}) \\
= &\arg\min_{\mathbf{W}} \sum_{ij} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathrm{F}}^2 + \mathbf{1}^\top \log\left(\mathbf{W} \cdot \mathbf{1}\right) - \log P(\mathbf{W}) \\
\equiv &\arg\min_{\mathbf{W}} S(\mathbf{X}, \mathbf{W}) + \Omega(\mathbf{W})
\end{aligned} \tag{8}$$

The training objective consists of graph-signal fidelity term $S(\mathbf{X}, \mathbf{W}) \equiv \sum_{ij} w_{ij} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathrm{F}}^2$ and a structural regularizer $\Omega(\mathbf{W}) \equiv \mathbf{1}^\top \log\left(\mathbf{W} \cdot \mathbf{1}\right) - \log P(\mathbf{W})$. This suggests that inferring the graph topology through MAP estimation parameterized by PE-MRF is equivalent to the GSI algorithms through minimizing smoothness Dong et al. (2015). The graph-signal fidelity term is exactly the smoothness measuring the variation of signals on a graph Thus, finding the optimal weight matrix that minimizes graph-signal fidelity term $S(\mathbf{X}, \mathbf{W})$ facilitates to learn a graph having signals varying slowly along edges Chen et al. (2015). The optimization framework for the GSI problem Dong et al. (2015) has the following format,

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} S(\mathbf{X}, \mathbf{W}) + \Omega(\mathbf{W}), \tag{9}$$

is branched into multiple variants, which include graph Lasso Friedman et al. (2007) which utilizes an ℓ_1 regularizer to promote the sparsity of learned graph, and Kalofolias (2016b) which considers a log-barrier to enhance the connectivity of a graph, etc. In these studies, the log determinant terms are omitted as maximizing the nominator in eq. (2) already formalizes a valid MLE problem.

Remark 1 *The impact of assumptions.*

To transit to the hidden Markov Networks Ghahramani (2001) in Gaussian Graphical Models Yuan & Lin (2007), where $P(\mathbf{X}_{:k} \mid \Theta) \sim \mathcal{N}(0, \Lambda^{-1})$, two important assumptions are made: 1) Each feature dimension in $\mathbf{X}$ contributes equally to the structure estimation, so $\mathbf{V} = \alpha \mathbf{1}_{K \times C}$ becomes a simple scaling matrix and 2) the variance of $\mathbf{x}$, Σ_x is considered to be relatively small to the feature generation covariance across all the nodes, so the conditional covariance matrix of $\mathbf{X}$ is simply $\alpha\Sigma$. These two assumptions can simplify the algorithm, but also introduce limitations to the model. Specifically, fixing the emission matrix $\mathbf{V}$ to be orthonormal makes the algorithm fail to adaptatively focus on the most useful feature sub-spaces of graph signals when learning the graphs. We will address such a limitation in the design of new generative models for graph signals in section 4

4 Data-generating Process for Heterogeneous Graphs

In what follows, we will first define the data-generating process for heterogeneous graphs, and then propose an algorithm to solve the HGSL problem.

4.1 Heterogeneous Graphs.

Heterogeneous graphs consider different types of nodes and edges. A (undirected) heterogeneous graph Xiao et al. (2019), $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ is associated with a node type set $\mathcal{A}$ and an edge type set $\mathcal{R}$. Each node $v \in \mathcal{V}$ has a node type $\phi(v) \in \mathcal{A}$ and each edge $e \in \mathcal{E}$ is assigned with a relation type $r \in \mathcal{R}$. Since heterogeneous graphs are multi-relational, an edge in the graph is an undirected triplet $e = \{v, u, r\}$, which means node v and u are connected by relation type r. Thus, we represent the connections on the graph as a weighted 3-D tensor $\mathbb{W} = \{w_{vur}\} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|}$, with $|\cdot|$ the set cardinality. For each node pair, $\mathbb{W}_{vu:}$ is a $|\mathcal{R}|$-long vector with only one nonzero entry indicating the edge weight and type. Most of the heterogeneous graph datasets Lv et al. (2021); Fu et al. (2020); Guo et al. (2023); Hu et al. (2020a); Zhang & Chen (2020) condition edge types on the node types, i.e., if node u is type 'user' and v is 'item', the relation type r can only be 'like' or 'dislike'. We follow this convention to reduce the degree of freedom in determining relation types. Last, the

node signals on a heterogeneous graph can be represented by a function $f : \mathcal{V} \to \mathbb{R}^K$, which assigns a vector $\mathbf{x}_v \in \mathbb{R}^K$ to node $v \in \mathcal{V}$. emphasize the difference of $\mathbf{y}_v$ in this case to the homogeneous graph case. Some datasets Lv et al. (2021); Traud et al. (2011); Guo et al. (2023) have a type-specific dimension for each node type. The study will focus on the cases with unified dimension sizes but the methods can be generalized to any heterogeneous graph by projecting the signals into a universal $\mathbb{R}^K$ with extra linear modules. Note that we do not consider edge signals encoding edge attributes and leave it for future work.

The relation-wise connectivity matrix In HGs, we need to consider various node and edge types and their impact on the node labels. Specifically, a node v will first have type $\phi(v)$, and then within each type, its class is represented by a one-hot vector $\mathbf{y}_v \in \mathcal{Y}_{\phi(v)} \in \mathbb{R}^{C_{\phi(v)}}$, where $\mathcal{Y}_{\phi(v)}$ is the class space for node type $\phi(v)$ and $C_{\phi(v)}$ is the number of classes. For instance, nodes can have types "actor" and "movie" in a movie review graph. Within each type, nodes are classified into genres like "adventure" or "action". For edges, we further have a subset of relation types within any paired node types, $\mathcal{R}_{\phi(v),\phi(u)}$, that generates edge type r. For each r, we define a probability matrix $\mathbf{B}_r \in \mathbb{R}^{|C_{\phi(v)}|\times|C_{\phi(u)}|}$, where each entry of the matrix encodes the probability of two nodes belonging to specific classes to be connected with r. Following the example of the movie review dataset, actors and movies in the same genre are more likely (with high probability) to be connected by a "star in" edge type. To make sure that $\mathbf{B}_r$ formalizes a valid probability matrix, the ℓ_1 (sum of the entries) is $\|\mathbf{B}_r\|_1 = 1$.

4.2 Hidden Markov Networks for Heterogeneous Graphs

While the current research advancement of statistical graph models is unable to model the heterogeneous graphs, we first generalize the PE-MRF to heterogeneous graphs by redefining the potential functions and name such model hidden Markov networks for heterogeneous graphs (H2MN).

The model With such an understanding, we extend a graphical model for HG as in fig. 2. Similar to the hidden Markov networks introduced, the likelihood can be decomposed into the node-wise potential $\psi(\mathbf{y}_v)$, the pair-wise potential $\psi(\mathbf{y}_v, \mathbf{y}_u \mid \mathbf{B}_r)$ and the emission process whose density function is $P(\mathbf{x}_v \mid \mathbf{y}_v)$. For the sake of brevity, we first consider the joint probability function of node signals and labels, which yields

$$P(\{\mathbf{x}_v\}, \{\mathbf{y}_v\} \mid \mathbb{W}, \{\mathbf{B}_r\}) = \frac{1}{Z} \prod_v \psi\left(\mathbf{y}_v\right) \prod_{u,v,r} \psi\left(\mathbf{y}_u, \mathbf{y}_v \mid \mathbf{B}_r, w_{uvr}\right) \prod_v P\left(\mathbf{x}_v \mid \mathbf{y}_v\right). \tag{10}$$

Similar to the HMN for homogeneous graphs, the generation density of $\{\mathbf{y}_v\}$ in HGs is defined as,

$$P(\{\mathbf{y}_v\} \mid \mathbb{W}, \{\mathbf{B_r}\}) \propto \exp\left(-\sum_v (d_v + \nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2 + \sum_{u \neq v, r} w_{vur} \mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v\right), \tag{11}$$

To link label variable $\mathbf{y}_v$ with the signal variable $\mathbf{x}_v$, we assume that the conditional distribution of $P(\mathbf{x}_v \mid \mathbf{y}_v)$ has a mean that is a linear function of $\mathbf{y}_v$, and a covariance $\sigma^2\mathbf{I}$ that is independent of $\mathbf{y}_v$, which gives,

$$\mathbf{x}_v \mid \mathbf{y}_v \sim \mathcal{N}\left(\mathbf{x}_v \mid \mathbf{V}_{\phi(v)}(\mathbf{y}_v - \mu_{\mathbf{y}_v}), \sigma^2\mathbf{I}\right) \tag{12}$$

where the liner transformation matrix $\mathbf{V}_{\phi(v)} \in \mathbb{R}^{K\times|C_{\phi(u)}|}$ from $\mathbf{y}_v$ to $\mathbf{x}_v$ is assumed to be determined by the node type $\phi(v)$. We then substitute eq. (24) and eq. (12) into eq. (10), marginalize out $\{\mathbf{y}_v\}$ and obtain the overall likelihood for the graph signals,

$$\begin{aligned} &P(\{\mathbf{x}_v\} \mid \mathbb{W}, \{\mathbf{B}_r\}) \propto \\ &\prod_v \exp(-(\nu + d_v)\|\mathbf{U}_{\phi_v}\mathbf{x}_v\|_{\mathrm{F}}^2) \cdot \prod_{u,v,r} \exp\left\{w_{uvr}\,\mathrm{tr}(\mathbf{x}_u^\top \mathbf{U}_{\phi(u)}^\top \mathbf{B}_r \mathbf{U}_{\phi(v)}\mathbf{x}_v)\right\} \end{aligned} \tag{13}$$

where $\mathbf{U}_{\phi(u)} \in \mathbb{R}^{|C_{\phi(u)}|\times K}$ is the Moore–Penrose Pseudo inverse of matrix $\mathbf{V}_{\phi(u)}$ which satisfies $\mathbf{V}_{\phi(u)}\mathbf{U}_{\phi(u)}\mathbf{V}_{\phi(u)} = \mathbf{V}_{\phi(u)}$ and $\mathbf{U}_{\phi(u)}\mathbf{V}_{\phi(u)}\mathbf{U}_{\phi(u)} = \mathbf{U}_{\phi(u)}$. The detailed derivations can be found in appendix A.3

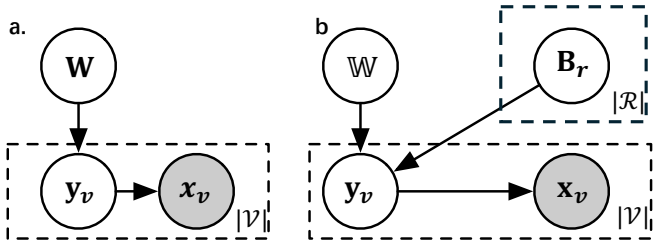


Figure 2: The graphical models for a) Ising model and b) ours. The shadowed variable is observable.

5 Heterogeneous Graph Structure Learning

In different heterogeneous graph datasets, the observables are different. For example, in IMDB and ACM datasets, the node features only contain labels for node classes, $\mathbf{Y}$. In the city traffic network (add the citation here), no node class is observed while the traffic volume is viewed as the signals $\mathbf{X}$. In this section, we first dive into the case when only the node signals $\mathbf{x}_v$ are observable, while leaving the generalization to other scenarios in appendix B.1.

5.1 Problem Formulation

The problem of heterogeneous graph structure learning (HGSL) is formally stated as follows. Given nodes $\{v\}_{v\in\mathcal{V}}$ with corresponding type $\{\phi(v)\}_{v\in\mathcal{V}}$ and associated features $\{\mathbf{z}_v\}_{v\in\mathcal{V}}$, together with a potential relation type set $\mathcal{R}$, we aim to learn a weighted and undirected HG $\mathcal{G}$ represented by tensor $\mathbb{W}$ that encodes the identification and types of edges, and any associated parameters Θ is learned as bi-product. Specifically, the node features $\mathbf{z}_v$ contain all the observable information, which can be labels $\{\mathbf{y}_v\}$, signals $\{\mathbf{x}_v\}$ or a combination of both. For simplicity, we focus on learning the graph structure from signals $\{\mathbf{x}_v\}$ as the initial attempt, and then dive into extension later in **??**

The maximum a-posteriori estimation formulation. In order to properly specify the generative process of heterogeneous graphs, we will also need some parameters, denoted as Θ, where we leave the introduction in section 4. The problem aims at learning heterogeneous graph structure from features living on nodes together with the optimal parameters. Statistically, this can be viewed as an inference problem, where, we wish to find the optimal graph structure $\mathbb{W}^*$ that generates the graph signals and parameters Θ^*,

$$\begin{aligned}\mathbb{W}^*, \Theta^* &= \arg\max_{\mathbb{W},\Theta} \log P(\mathbb{W}, \Theta \mid \{\mathbf{x}_v\}) \\ &= \arg\max_{\mathbb{W},\Theta} \log P(\{\mathbf{x}_v\} \mid \mathbb{W}, \Theta) - \Omega(\mathbb{W}) - \Omega(\Theta).\end{aligned} \tag{14}$$

5.2 Reparametrization

The likelihood function follows eq. (13) and the parameters Θ contains the graph structure $\mathbb{W}$, the connectivity matrix $\{\mathbf{B}_r, \forall r \in \mathbb{R}\}$ and the emission matrix $\{\mathbf{U}_{\phi(v)}, \forall \phi(v) \in \mathcal{A}\}$. For further simplify the estimation, we reparmeterize $\mathbf{E}_r \equiv (\mathbf{U}_{\phi(u)}^\top \mathbf{B}_r \mathbf{U}_{\phi(v)})^{\frac{1}{2}}$, where we specify a relation-wise matrix $\mathbf{E}_r$ that depends only on r since $\phi(u)$ and $\phi(v)$ can be directly determined when $r \in \mathcal{R}_{\phi(v),\phi(u)}$ is given. Thus, the likelihood can be re-formalized as,

$$\begin{aligned}P(\{\mathbf{x}_v\} \mid \mathbb{W}, \{\mathbf{E}_r\}) \propto &\Pi_v \exp(-(\nu + d_v)\|\mathbf{E}_r \mathbf{x}_v\|_{\mathrm{F}}^2) \\ &\cdot \Pi_{u,v,r} \exp\left\{\beta w_{uvr} \operatorname{tr}(\mathbf{x}_u^\top \mathbf{E}_r^\top \mathbf{E}_r \mathbf{x}_v)\right\}.\end{aligned} \tag{15}$$

The reparameterization is up to a scalar,

$$\beta^{-1} = \frac{\bar{d}}{|\mathcal{V}|} \frac{\|\mathbf{U}_{\phi(v)}\mathbf{x}_v\|_{\mathrm{F}}^2}{\|\operatorname{vec}(\mathbf{B}_r)^{1/2}\mathbf{U}_{\phi(v)}\mathbf{x}_v\|_{\mathrm{F}}^2}.$$

Algorithm 1: HGSL with Alternative Weight Update

Input: Node signals $\{\mathbf{x}_v\}_{v=1}^{|\mathcal{V}|}$ and types $\{\phi(v)\}_{v=1}^{|\mathcal{V}|}$;
Relation type set $\mathcal{R}$; Maximum steps T.
Output: The graph weight $\mathbf{w}$; Embeddings $\mathbf{e}_r, \forall r \in \mathcal{R}$;
Init: Initialize $\mathbf{e}_r^0 = \mathbf{1}^{\mathbf{T}}/K, \forall r \in \mathcal{R}$; Initialize $\mathbf{w}^0$ randomly ;
while $t < T$ **do**
`/* Graph structure learning step */`
Calculate the smoothness vector $\mathbf{z}$ based on eq. (54);
Optimize $\mathbf{w}^{t+1}$ based on eq. (19);
`/* Relation embedding update Step */`
if Update Method == Gradient Descent **then**
$\mathbf{e}_r^{t+1} = \mathrm{GD}(\mathbf{e}_r^t)$;
else if Update Method == Iterative Reweighting **then**
Update $\mathbf{e}_r^{t+1}$ based on eq. (21);
end if;
t = t+1;
end

According to the operation that $\mathbf{x}_v$ is normalized as a unit vector w.r.t ℓ_2 norm, we can derive the terms inside the exponential function as,

$$\begin{aligned}
&\sum_{u,v,r} \beta w_{uvr} \operatorname{tr}(\mathbf{x}_u^\top \mathbf{E}_r^\top \mathbf{E}_r \mathbf{x}_v) - (\nu + d_v)\|\mathbf{E}_r \mathbf{x}_v\|_{\mathrm{F}}^2 \\
=& -\sum_{u,v,r} w_{uvr}\|\mathbf{E}_r \mathbf{x}_u - \mathbf{E}_r \mathbf{x}_v\|_{\mathrm{F}}^2 - \beta'(w_{vur} + \frac{\nu}{\beta'})\|\mathbf{E}_r \mathbf{x}_v\|_{\mathrm{F}}^2 \\
=& -\sum_{u,v,r} w_{uvr}\|\mathbf{E}_r \mathbf{x}_u - \mathbf{E}_r \mathbf{x}_v\|_{\mathrm{F}}^2 - \beta'(w_{vur} + \frac{\nu}{\beta'})\|\mathbf{E}_r\|_{\mathrm{F}}^2
\end{aligned} \tag{16}$$

where $\beta' = \beta - 2$. According to the representation theory, $\mathbf{E}_r \mathbf{x}_v$ can be fully represented by a vector element-wise product, where $\mathbf{E}_r \mathbf{x}_v = \mathbf{e}_r \odot \mathbf{x}_v$, thus the training objective follows if $\mathbf{x}_v$ and $\mathbf{x}_u$ are from the same measurable space,

$$\underset{\mathbb{W},\mathbf{E}}{\arg\min} \quad \sum_{v \le u,r} w_{vur}\|\mathbf{e}_r \odot (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 + \beta' \sum_{v \le u,r} w_{vur}\|\mathbf{e}_r\|_{\mathrm{F}}^2 + \Omega(\mathbb{W}) + \Omega(\mathbf{E}). \tag{17}$$

With a bit overuse of notation, the relation-wise vectors $\mathbf{e}_r$ is stacked as $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_{|\mathcal{R}|}]$ and $\nu\|\mathbf{e}_r\|_{\mathrm{F}}^2$ is absorbed into the regularizer $\Omega(\mathbf{E})$. It is intuitive to see that the first term, $\sum_{u,v,r} w_{vur}\|\mathbf{e}_r \odot (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2$, serves similarly to the smoothness objective in vanilla graph structure learning literature Pu et al. (2021a); Dong et al. (2015) as in eq. (7) if we consider the relation embedding $\mathbf{e}_r$ as a way to reweight signals across different dimensions. Inspired by classical GSL methods Kalofolias (2016a); Dong et al. (2015), we choose $\Omega(\mathbb{W})$ to consist of a log barrier regularizer on node degrees and an indicator function on $\mathbb{W}$ to make the weights positive. The ℓ_2 regularizer is applied to $\mathbf{E}$.

5.3 Algorithm Design of Heterogeneous Graph Structure Learning

Since the problem requires learning both the *graph structures* $\mathbb{W}$ and *relation embeddings* $\mathbf{E}$, we adopt an alternating optimization scheme to solve it: $\mathbb{W}$ is optimized with $\mathbf{E}$ fixed, and $\mathbf{E}$ is optimized with $\mathbb{W}$ fixed. For notation consistency, we will use $\mathcal{E}'$ to denote the possible connections in a heterogeneous graph. Note that this is not the combination of all possible nodes and relation types that shapes a $|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|$ space as node pairs with certain types can only be connected by specific edge types.

Graph structure learning step. The first sub-optimization problem is to find $\mathbb{W}$ that minimizes the training objective in eq. (17) with a fixed $\mathbf{E}$. Since $\mathbf{e}_r$ has norm 1, the problem reduces to

$$\begin{aligned} &\arg\min_{\mathbb{W}} \quad \sum_{v\leq u,r} w_{vur}\|\mathbf{e}_r \odot (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 + \beta' w_{vur} + \Omega(\mathbb{W}) \\ =&\arg\min_{\mathbb{W}} \quad \langle \mathbb{W}, \|(\mathbf{X}\otimes\mathbf{1} - \mathbf{1}\otimes\mathbf{X})\otimes\mathbf{E}\|_F^2\rangle + \Omega(\mathbb{W}) \\ &\text{s.t. } [\mathbb{W}]_{vur} = [\mathbb{W}]_{uvr} \geq 0, \text{ for } v\neq u, \forall r \end{aligned} \tag{18}$$

where $\odot$ is the element-wise product,$\langle\cdot,\cdot\rangle$ and $\otimes$ are the tensor inner product and outer product, respectively. It is intuitive to see that the first term, $\sum_{u,v,r} w_{vur}\|\mathbf{e}_r \odot (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2$, serves similarly to the smoothness objective in vanilla graph structure learning literature Pu et al. (2021a); Dong et al. (2015) if we consider the relation embedding $\mathbf{e}_r$ as a way to reweight signals across different dimensions.

Since we focus on undirected graphs (i.e., $\mathbb{W}$ symmetric), we only need to focus on the upper triangle part of the tensor, i.e., the vectorized weights $\mathbf{w} \in \mathbb{R}_+^{|\mathcal{V}|\cdot(|\mathcal{V}|-1)\cdot|\mathcal{R}|/2}$. We reparameterize the training task as in Pu et al. (2021a) and obtain $\sum_{\{v,u,r\}\in\mathcal{E}'} w_{vur}\|\mathbf{e}_r \circ (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 = \|\mathbf{w}\odot\mathbf{z}\|$, where $\mathbf{z} \in \mathbb{R}_+^{|\mathcal{V}|\cdot(|\mathcal{V}|-1)\cdot|\mathcal{R}|/2}$ is the half-vectorization of the tensor $\|(\mathbf{X}\otimes\mathbf{1} - \mathbf{1}\otimes\mathbf{X})\otimes\mathbf{E}\|_F^2$. In addition, inspired by classical GSL methods Kalofolias (2016a); Dong et al. (2015), we choose $\Omega(\mathbb{W})$ to consist of a log barrier regularizer on node degrees and a ℓ_1 norm regularizer on the weights to promote the connectivity and the sparsity of the graph respectively. With this, the objective is reformulated as,

$$\arg\min_{\mathbf{w}} \|\mathbf{w}\odot\mathbf{z}\| - \alpha\mathbf{1}^\top\log(\mathcal{T}\mathbf{w}) + \beta^2\|\mathbf{w}\|_1 + \mathcal{I}_{\mathbf{w}>0} \tag{19}$$

and $\mathcal{T}$ is a linear operator that transforms $\mathbf{w}$ into the vector of node degrees such that $\mathcal{T}\mathbf{w} = (\sum_{r=1}^{|\mathcal{R}|}\mathbb{W}_{::r})\cdot\mathbf{1}$. There $\beta^2\|\mathbf{w}\|_1$ comes from the factor that $\mathbf{e}_r$ has a ℓ_2 norm 1. We follow the literature and solve eq. (19) by alternating direction method of multipliers (ADMM) Pu et al. (2021a) or primal-dual splitting algorithms (PDS) Kalofolias (2016a).

Relation embedding update step The second sub-problem takes care of the optimization of relation embeddings $\mathbf{e}_r$. Deriving from eq. (17), we obtain

$$\arg\min_{\{\mathbf{e}_r\}} \quad \sum_{vur} w_{vur}\|\mathbf{e}_r \odot (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 + \beta' w_{vur}\|\mathbf{e}_r\|_{\mathrm{F}}^2 + \Omega(\mathbf{E}). \tag{20}$$

Intuitively, $\mathbf{e}_r$ can be directly optimized by gradient descent, which we name as HGSL-GD. However, in our experiments, we found this unstable and easily stuck into a sub-optimum. Thus, a more efficient and stable solution is developed through analytically solving eq. (20) w.r.t. $\mathbf{e}_r$ with $\beta' = 1$. The detailed derivation is left in appendix A.4. We call this algorithm iterative reweighting (HGSL-IR) as intuitively this solution assigns higher weights for relation r to dimensions that express higher dimension-wise similarity on the graph learned at iteration t (i.e., w_{vur}^t), which is formulated as

$$\mathbf{e}_{r,k}^{t+1} = \frac{\sum_{\{v,u,r\}\in\mathcal{E}_r'} w_{vur}^t\ \mathbf{x}_{v,k}\cdot\mathbf{x}_{u,k}}{\sqrt{(\sum_{\{v,u,r\}\in\mathcal{E}_r'}\mathbf{x}_{v,k}^2)(\sum_{\{v,u,r\}\in\mathcal{E}_r'}\mathbf{x}_{u,k}^2)}} \tag{21}$$

where $\mathcal{E}_r'$ is the set of possible edges of type r. In each iteration, we normalize relation embeddings to be unit vectors in terms of the ℓ_2 norm. The whole process is in algorithm 1.

5.4 Algorithm Analysis

The probability density function in DGPs for homogeneous graphs (eq. (1)) and heterogeneous graphs (eq. (10)) both can be decomposed into node-wise potential function, edge-wise potential function and feature emission function. Thus, it is natural to conduct the algorithm analysis from these three perspectives. The node-wise and edge-wise potential potentials impact the generation of node labels through the relation-wise connectivity matrix $\{\mathbf{B}_r\}$ and the emission function controls the node feature generation through $\mathbf{U}_{\phi(v)}$. In what follows, we will show how $\{\mathbf{B}_r\}$ is related to the homophily of a heterogeneous graph, and how $\mathbf{U}_{\phi(v)}$ impacts the distinguishability of the edges and their types.

The homophily of a heterogeneous graph: The concept of homophily in networks was first proposed in McPherson et al. (2001), suggesting that a connection between similar entities occurs at a higher probability than among dissimilar entities. In graph with only two classes, the most accurate formulation in the statistical community would be having a higher probability of observing an edge between two nodes belonging to the same communities. Such a concept gives $P(\mathbf{W}_{ij} = 1 \mid Y_i = Y_j) > P(\mathbf{W}_{ij} = 1 \mid Y_i \neq Y_j)$, where Y_i and Y_j are the labels for nodes i and j, and the homophily ratio is defined as the ratio of these two values.

Such a concept can be generalized to heterogeneous graphs. Considering the Cartesian product of node type combination space $t \in \mathcal{T} = \mathcal{A} \times \mathcal{A}$, a possible relation type as $r \in \mathbb{R}_{\phi(u)\phi(v)}$ and then we denote an element of possible node label combinations within two node types as $\tau \in \mathcal{Y}_{\phi(u)} \times \mathcal{Y}_{\phi(v)}$. Then we define a "representative" connection between two nodes within a combination of node types, that gives the highest probability of connection $\hat{p}_{r^*,\tau^*}$. Then, the sufficient condition for the optimization problem to have meaningful solutions is that $\forall r, \tau$,

$$p_{r,\tau^*} - \sum_{\tau \neq \tau^*} p_{r,\tau} > 0. \tag{22}$$

The inequality serves as the proper definition of "homophily" in a heterogeneous graphs. In appendix C.3 we prove that algorithm 1 above inequality is sufficient for guaranteeing a meaningful solution[3] and the homophily ratio can be defined as $p_{r,\tau^*}/(\sum_{\tau \neq \tau^*} p_{r,\tau})$. However, in practice the above value is difficult to compute due to the common missing records of node labels. Instead, when processing the datasets, people usually retain one type of "Target node" and record the label of that node type. Thus, we further show in appendix C.3 that the homophily ratio used in the literature Guo et al. (2023) can instead work as an measurement.

[**Relaxed Homophily Ratio (RHR)**] measures how similar the connected nodes are in a heterogeneous graph, which is related to the smoothness of the signal on a graph. It is defined based on the meta-paths set induced by the graph Wang et al. (2020). A meta-path Φ is a path following a specific sequence of node and relation types like $\mathcal{A}_1 \xrightarrow{\mathcal{R}_1} \mathcal{A}_2 \xrightarrow{\mathcal{R}_2} \cdots \xrightarrow{\mathcal{R}_L} \mathcal{A}_L$. When $\mathcal{A}_1 = \mathcal{A}_L$, the graph induced by Φ is a homogeneous graph $\mathcal{G}_\Phi$ with edges being the meta-path defined by Φ. This helps us to define the HR:

$$\mathrm{HR}\left(\mathcal{G}_\Phi\right) = \frac{\sum_{\{i,j\} \in \mathcal{E}_\Phi} \mathbb{I}\left(y_i = y_j\right)}{|\mathcal{E}_\Phi|}, \tag{23}$$

where y_i and y_j are the node labels for downstream classification tasks. This homophily was empirically studied in Guo et al. (2023) but was not theoretically justified. Here we work as the first to connect the empirical study and the theory behind it. The homophily is also closely related to the meta-path-based method which considers the network schema Liu et al. (2018), given that the random walk between same-typed nodes is always considered in a meta-path.

The Emission Function. Another important aspect of estimating heterogeneous graphs from the node feature is how much the node feature $\mathbf{X}$ can represent the true label $\mathbf{Y}$ behind. This requires us to have some further assumptions on the transformation matrix $\phi(v)$. Intuitively, the node features for various node types should be unified into the same measurable space. In many datasets, such as ACM and IMDB, this is not difficult since the features are just the keywords of academic work and movie captions respectively. However, in other datasets, e.g. DBLP this is more difficult given that the features of venue and terms are not unified into the same space. For such datasets, a more reliable task would be link prediction, in which one gets the opportunity to learn the feature transformation a priori.

[**Smoothest-Dimension Overlapping Ratio (SDOR).**] If two relation types r and r' have a high SDOR, the relation embeddings learned in eq. (21) would be similar, thus solving **??** lead to a result with equal $\mathbb{W}_{::r}$ and $\mathbb{W}_{::r'}$. This makes two relation types indistinguishable in the learned graph.

We assume that edges with distinct relation types are associated with smoothness evaluated in different signal dimensions, and the smoothest dimensions are the most representative ones. This motivates us to define

[3] Avoiding non-convergence solution such as infinitely maximizing $\mathbb{W}$, or trivial solutions such that $\mathbb{W} = 0$

Smoothest-Dimension Overlapping Ratio (SDOR): for each pair of relations (r, r'), the SDOR is calculated by counting the overlapping dimensions in $\mathcal{K}^{\mathrm{M}}(r)$ and $\mathcal{K}^{\mathrm{M}}(r')$: $\mathrm{SDOR}(r, r') = \frac{|\mathcal{K}^{\mathrm{M}}(r) \cap \mathcal{K}^{\mathrm{M}}(r')|}{|\mathcal{K}^{\mathrm{M}}(r) \cup \mathcal{K}^{\mathrm{M}}(r')|}$. The SDOR reflects how different two relation types are in terms of exhibiting smoothness in signal dimensions and we report the pair-wise SDOR in table 1. The result suggests that for pairs of relation types with lower SDOR the signals will exhibit smoothness in different dimensions, which reveals the possibility of distinguishing them by comparing the smoothest dimensions found.

With all the above facts, we define two important factors impacting model performance:

6 experiments

There are three research questions to ask: RQ1) How do our algorithms perform compared to the vanilla GSL algorithms? RQ2) How do the factors remarked earlier impact the HGSL performance? RQ3) Does our model grants interpretability to the problem?

We consider the following extension to the experiment:

- We will compare the HD and HR on their impact on the HGSL performance. The conclusion should be that HD is a better metric for the algorithm performance, instead of HR.
- More datasets should be added to test the model performance. (Molecule dataset might not be ideal... as the signal is just the node type what else?)
- Maybe we can also consider a few pure multi-relational datasets?
- Or do we construct the dataset by ourselves?
- What real-world application can we have? - a new dataset in urban life Barlacchi et al. (2015)

6.1 Dataset Setups

Synthetic Datasets. The synthetic dataset construction consists of 3 phases:

- **Graph backbone generation.** We followed the process in Pu et al. (2021a) and used stochastic block model and Watts–Strogatz model to generate the graph backbone with 20-100 nodes that encodes the graph structure.
- **Graph labeling** We label the graph based on the network schema that encodes the node and edge type information Shi (2022). Network schema is a meta template of a heterogeneous graph that represents how nodes/edges with different types are connected. We traverse the graph by breadth-first search to follow the rule defined by the network schema and the nodes are labeled by the preset label distribution.
- **Feature generation.** We then generate signals for nodes that satisfy the probability distribution as follows,

$$p(\mathbf{X}, \mathbf{E} \mid \mathbb{W},) \propto \exp\left(-\frac{1}{\sigma} \sum_{\{v,u,r\} \in \mathcal{E}'} w_{vur} \|\mathbf{e}_r \circ (\mathbf{x}_v - \mathbf{x}_u)\|_2^2\right) \tag{24}$$

Real-world Datasets: Quantitative Experiments are conducted in datasets including IMDB Guo et al. (2023), and ACM Lv et al. (2021). The statistics are in table 1 and a brief summarization are as follows,

- **IMDB** Fu et al. (2020): a movie review dataset with node types including directors (D), actors (A), and movies (M) and with signals as 3066-D bag-of-words representation of keywords in the movie plot.

Table 1: Edge identification with different dimensions. AUC-M are the results with the smoothest dimensions and AUC-L the least ones.

Dataset	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\mathcal{R}$	HR	SDOR
IMDB	11k	550k	Movie-Actor (MA)	0.51	MA-MD: 0.65
			Movie-Director (MD)	0.900	
ACM	21k	87k	Paper-cite-Paper (PP)	0.64	PP-PA: 0.66
			Author-write-Paper (PA)	0.925	PP-PS: 0.67
			Paper-has-subject (PS)	0.804	PA-PS: 0.94

Table 2: Experiment results on Heterogeneous Graph Structure Learning

Dataset	**HR**	**K**	**Vanilla GSL**		**HGSL-GD**		**HGSL-IR**	
			AUC	NMSE	AUC	NMSE	AUC	NMSE
Synthetic	0.95+	60	0.66 ± 0.05	0.03 ± 0.00	0.70 ± 0.04	0.27 ± 0.05	**0.75 ± 0.02**	**0.02 ± 0.00**
		300	0.65 ± 0.01	0.03 ± 0.00	0.75 ± 0.05	0.30 ± 0.04	**0.83 ± 0.04**	**0.02 ± 0.01**
		600	0.78 ± 0.03	**0.04 ± 0.02**	0.77 ± 0.04	0.26 ± 0.04	**0.89 ± 0.04**	0.06 ± 0.01
IMDB	0.51	3066	0.74 ± 0.04	0.29 ± 0.08	0.75 ± 0.06	0.21 ± 0.05	**0.81 ± 0.07**	**0.07 ± 0.05**
ACM	0.64	1902	0.65 ± 0.06	**0.07± 0.10**	0.62 ± 0.06	0.09 ± 0.15	**0.73 ± 0.02**	0.12 ± 0.07

* Experiment results are evaluated over 30 trials and the mean/standard deviation is calculated. 0.00 means the value < 0.01.

- **ACM** Lv et al. (2021): an academic dataset contains papers (P), authors (A), and subjects (S). Signals correspond to the 1902-D bag-of-words representation of the keywords in diverse research areas.

In order to augment more graph instances for a comprehensive evaluation, we subsample the dataset to generate smaller graphs with number of nodes ranging from 50-200 following the strategy in Hu et al. (2020b).

We also conduct the qualitative experiment in a financial dataset to recover the relationship between S&P 100 companies.

6.2 Model Comparison

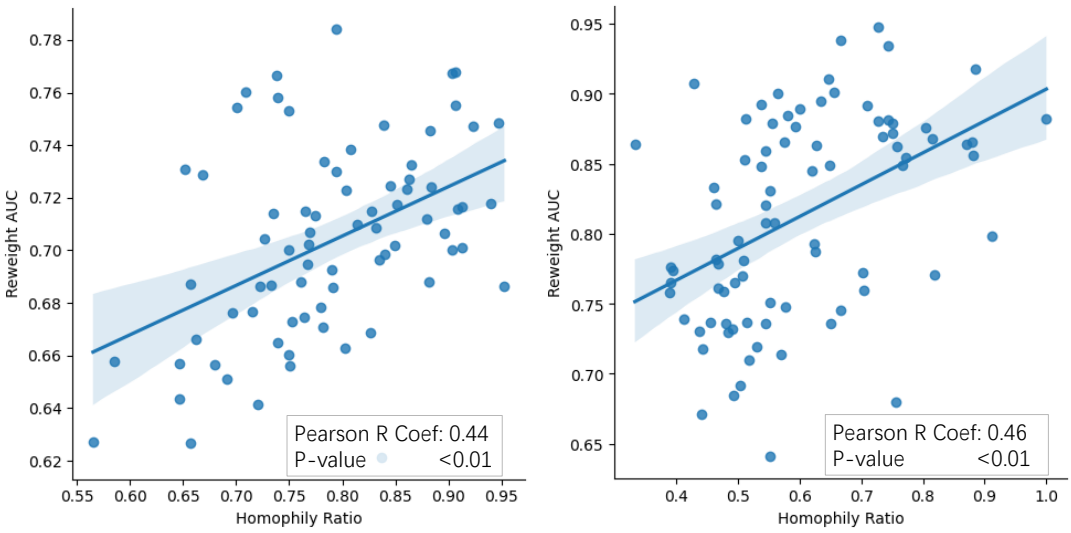
To answer RQ1, we benchmark existing algorithms with respect to structure recovery ability and stability. We follow the benchmark implementation in Pu et al. (2021a). The edge identification error (measured by AUC) and edge weight recovery error (measured by the normalized mean squared error, NMSE, defined in) are reported in table 2.

$$\text{GMSE} = \frac{1}{|\mathcal{V}| \times |\mathcal{V}| \times |\mathcal{R}|} \sum_{\{u,v,r\}} \frac{\|\hat{w}_{uvr} - w_{uvr}\|_{\text{F}}^2}{\|w_{uvr}\|_{\text{F}}^2} \tag{25}$$

With our HGSL algorithm with iterative reweighting (IR), a consistent improvement is found in all the datasets in terms of both AUC and NMSE. This suggests the efficacy of our algorithm in both tasks. When testing the HGSL with gradient descent (GD), though a better AUC is found in most datasets, the improvement is rather marginal and the NMSE is much larger. The improvement in synthetic datasets (avg +18.48%) is relatively larger than the real-world datasets (avg +10.88%), which suggests the more challenging nature of the real-world experiments.

6.3 Ablation Study

We could add an experiment comparing fixing the linear transformation matrix $\mathbf{V} = \mathbf{1}$ or make it learnable. To address the limitations of failing to focus on different signal dimensions adaptatively.



(a) ACM (Pearson Coef: 0.44) (b) IMDB (Pearson Coef: 0.46)

Figure 3: The correlation test between homophily ratio and AUC

Maybe use the brain signal dataset? or synthetic dataset?

6.4 Robustness Analysis

Homophily We first look into how **the Homophily Ratio** (HR) and the Hompphily Degree (HD defined in our paper) of a graph could impact the model robustness.

The overall HR and HD of 3 datasets are reported in table 2. We compare the experiment results among datasets with the homophily ratio defined in Guo et al. (2023) and homophily degree in our paper. It is clear that the dataset with a higher homophily ratio (ACM) does not perform better than datasets with a lower HR (IMDB), but IMDB has a higher Homophily Degree than ACM. This suggests that HD is a better indicator of the HGSL problem's solvability.

From the result, we are interested in how the algorithm performance is related to the homophily ratio. To verify our hypothesis that a lower HR impairs the performance, we record the homophily ratio of all the subgraphs when sampling and regress their corresponding AUC in both IMDB and ACM datasets, as shown in fig. 3. The Pearson Correlation Coefficients are reported. According to the results, we can conclude that the homophily ratio is positively correlated with AUC, and one future work could be understanding if this phenomenon is general among all the GSL algorithms.

Smoothest-Dimension Overlapping Ratio To understand the behavior of our algorithm against high **SDOR**, we fix the number of relation types $|\mathcal{R}| = 2$ in our synthetic experiment, and manually adjust the SDOR from 0 to 1. The higher SDOR is thought to impair the distinguishability of relation types. We illustrate the AUC of the vanilla GSL and HGSL algorithms in fig. 4, and calculate its relative increase. It is clear that the performance of HGSL is significantly affected when the SDOR increases, which suggests that a higher SDOR would impair the distinguishability of relation types. However, the HGSL algorithm is robust until the SDOR is increased to approximately 0.7, which is almost the same level in real-world datasets as shown in table 1.

6.5 Qualitative Results

We apply the HGSI algorithm to a financial dataset where we aim to recover the relationship between S&P 100 companies. We use the daily returns of stocks obtained from YahooFinance5. section 6.5 visualized the estimated relation-wise graph adjacency matrix where sectors sort the rows and columns. The heatmap

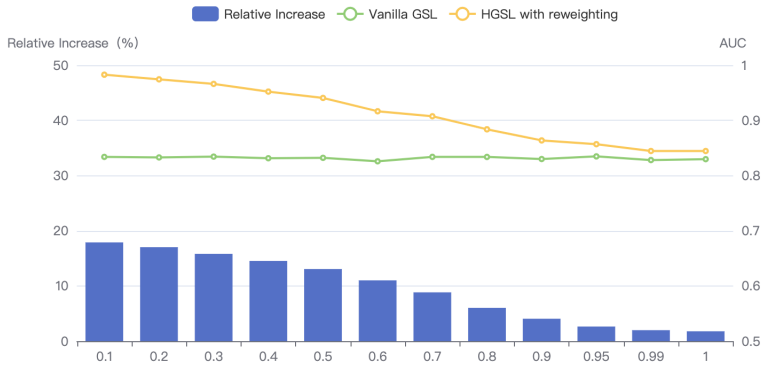
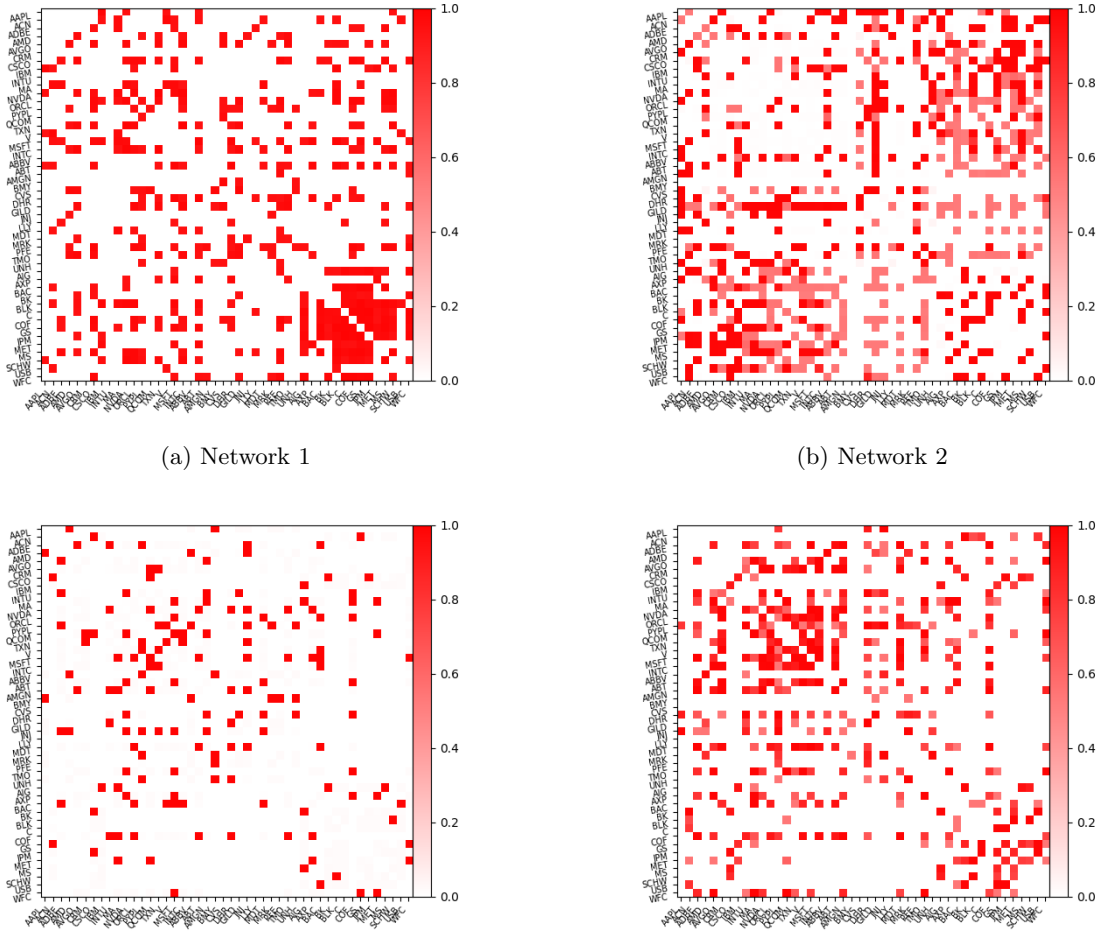


Figure 4: The SDOR and model performance



(a) Network 1

(b) Network 2

(c) Network 3

(d) Network 4

Figure 5: The average and standard deviation of critical parameters: Region R4

clearly shows that two stocks in the same sectors are likely to behave similarly. This is intuitive as both are related to health care.

7 discussion

In this study, we propose an HGSL framework based on minimizing generalized smoothness. The method is grounded in the assumption that “the smoothest dimensions are crucial", which has proven beneficial in our tests. However, certain real-world heterogeneous graphs, especially those with low homophily ratios, may not conform to this paradigm. In such instances, the smoothness assumption should be revisited and the model should also be modified to capture such signal behaviors. Additionally, the use cases of the current best method, HGSL-IR, are limited to a few assumptions made, e.g. the fixed signal dimension size and positive weights on different signal dimensions. However, HGSL with gradient descent is not constrained by these. Thus, another future work would be developing more reliable GD-based algorithms for general HGSL tasks.

8 Conclusion

In this paper, we propose a new heterogeneous graph structure learning problem, which aims at learning the heterogeneous graph structure from observed data. We provide a preliminary algorithm capable of solving HGSL, and the experiments suggest that our algorithm consistently outperforms vanilla graph structure learning algorithms. We then try to understand the limitations of our algorithm and conclude that the homophily ratio and smoothest-dimension overlapping ratio are the two crucial factors of the model performance.

Broader Impact Statement

In this optional section, TMLR encourages authors to discuss possible repercussions of their work, notably any potential negative impact that a user of this research should be aware of. Authors should consult the TMLR Ethics Guidelines available on the TMLR website for guidance on how to approach this subject.

Author Contributions

If you’d like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors. Only add this information once your submission is accepted and deanonymized.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments, including those to funding agencies, go at the end of the paper. Only add this information once your submission is accepted and deanonymized.

References

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d’Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, 2008. doi: 10.5555/1390681.1390696. URL `https://dl.acm.org/doi/10.5555/1390681.1390696`.

João Barata and Mahir Hussein. The moore-penrose pseudoinverse. a tutorial review of the theory. *Brazilian Journal of Physics*, 42, 10 2011. doi: 10.1007/s13538-011-0052-z.

Gianni Barlacchi, Marco De Nadai, Roberto Larcher, Antonio Casella, Cristiana Chitic, Giovanni Torrisi, Fabrizio Antonelli, Alessandro Vespignani, Alex Pentland, and Bruno Lepri. A multi-source dataset of urban life in the city of milan and the province of trentino. *Scientific Data*, 2:150055, 10 2015. doi: 10.1038/sdata.2015.55.

Claudio Battiloro, Indro Spinelli, Lev Telyatnikov, Michael M. Bronstein, Simone Scardapane, and Paolo Di Lorenzo. From latent graph to latent topology inference: Differentiable cell complex module. *CoRR*, abs/2305.16174, 2023.

Eugene Belilovsky, Kyle Kastner, Gaël Varoquaux, and Matthew B. Blaschko. Learning to discover sparse graphical models. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pp. 440–448. PMLR, 2017.

Peter Berger, Gabor Hannak, and Gerald Matz. Efficient graph learning from noisy and incomplete data. *IEEE Trans. Signal Inf. Process. over Networks*, 6:105–119, 2020.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 1247–1250, 2008.

Stephen P. Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, 2011.

Ines Chami, Sami Abu-El-Haija, Bryan Perozzi, Christopher Ré, and Kevin Murphy. Machine learning on graphs: A model and comprehensive taxonomy. *J. Mach. Learn. Res.*, 23:89:1–89:64, 2022.

Siheng Chen, Aliaksei Sandryhaila, José M. F. Moura, and Jelena Kovacevic. Signal recovery on graphs: Variation minimization. *IEEE Trans. Signal Process.*, 63(17):4609–4624, 2015.

Zengde Deng and Anthony Man-Cho So. A fast proximal point algorithm for generalized graph laplacian learning. In *ICASSP*, pp. 5425–5429. IEEE, 2020.

Yash Deshpande, Subhabrata Sen, Andrea Montanari, and Elchanan Mossel. Contextual stochastic block models. In *NeurIPS*, pp. 8590–8602, 2018.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, pp. 1811–1818. AAAI Press, 2018.

Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. Laplacian matrix learning for smooth graph signal representation. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3736–3740, 2015. doi: 10.1109/ICASSP.2015.7178669.

Xiaowen Dong, Dorina Thanou, Michael G. Rabbat, and Pascal Frossard. Learning graphs from data: A signal representation perspective. *IEEE Signal Process. Mag.*, 36(3):44–63, 2019.

Xiaowen Dong, Dorina Thanou, Laura Toni, Michael M. Bronstein, and Pascal Frossard. Graph signal processing for machine learning: A review and new perspectives. *IEEE Signal Process. Mag.*, 37(6): 117–127, 2020.

Hilmi E. Egilmez, Eduardo Pavez, and Antonio Ortega. Graph learning from data under laplacian and structural constraints. *IEEE J. Sel. Top. Signal Process.*, 11(6):825–841, 2017.

László Erdős, Antti Knowles, Horng-Tzer Yau, and Jun Yin. Spectral statistics of erdős–rényi graphs i: Local semicircle law. 2013.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 12 2007. ISSN 1465-4644. doi: 10.1093/biostatistics/kxm045. URL `https://doi.org/10.1093/biostatistics/kxm045`.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Xinyu Fu, Jiani Zhang, Ziqiao Meng, and Irwin King. MAGNN: metapath aggregated graph neural network for heterogeneous graph embedding. In *WWW*, pp. 2331–2341. ACM / IW3C2, 2020.

Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS*, pp. 585–593. Curran Associates, Inc., 2009.

Zoubin Ghahramani. An introduction to hidden markov models and bayesian networks. *International journal of pattern recognition and artificial intelligence*, 15(01):9–42, 2001.

Jiayan Guo, Lun Du, Wendong Bi, Qiang Fu, Xiaojun Ma, Xu Chen, Shi Han, Dongmei Zhang, and Yan Zhang. Homophily-oriented heterogeneous graph rewiring. In *WWW*, pp. 511–522. ACM, 2023.

William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *NeurIPS*, 2020a.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *WWW*, pp. 2704–2710. ACM / IW3C2, 2020b.

Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31:253–258, 1925. URL `https://api.semanticscholar.org/CorpusID:122157319`.

Vassilis Kalofolias. How to learn a graph from smooth signals. In *AISTATS*, volume 51 of *JMLR Workshop and Conference Proceedings*, pp. 920–929. JMLR.org, 2016a.

Vassilis Kalofolias. How to learn a graph from smooth signals. In Arthur Gretton and Christian C. Robert (eds.), *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pp. 920–929, Cadiz, Spain, 09–11 May 2016b. PMLR.

Sandeep Kumar, Jiaxi Ying, Jose Vinicius de Miranda Cardoso, and Daniel Palomar. Structured graph learning via laplacian spectral constraints. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL `https://proceedings.neurips.cc/paper/2019/file/90cc440b1b8caa520c562ac4e4bbcb51-Paper.pdf`.

Sandeep Kumar, Jiaxi Ying, José Vinícius de Miranda Cardoso, and Daniel P. Palomar. Structured graph learning via laplacian spectral constraints. In *NeurIPS*, pp. 11647–11658, 2019b.

Brenden M Lake and Joshua B Tenenbaum. Discovering structure by learning sparse graphs. In *Proceedings of the 32nd Annual Conference of the Cognitive*, 2010.

Jie Liu, Chunming Zhang, Elizabeth Burnside, and David Page. Learning Heterogeneous Hidden Markov Random Fields. In Samuel Kaski and Jukka Corander (eds.), *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 576–584, Reykjavik, Iceland, 22–25 Apr 2014. PMLR. URL `https://proceedings.mlr.press/v33/liu14.html`.

Ziqi Liu, Chaochao Chen, Xinxing Yang, Jun Zhou, Xiaolong Li, and Le Song. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2077–2085, 2018.

Zhaosong Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM J. Matrix Anal. Appl.*, 31(4):2000–2016, 2010.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. Are we really making much progress?: Revisiting, benchmarking and refining heterogeneous graph neural networks. In *KDD*, pp. 1150–1160. ACM, 2021.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M. Suchanek. YAGO3: A knowledge base from multilingual wikipedias. In *CIDR*. www.cidrdb.org, 2015.

Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415–444, 2001. doi: 10.1146/annurev.soc.27.1.415.

Antonio Ortega, Pascal Frossard, Jelena Kovačević, José M. F. Moura, and Pierre Vandergheynst. Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*, 106(5):808–828, 2018. doi: 10.1109/JPROC.2018.2820126.

Youngsuk Park, David Hallac, Stephen P. Boyd, and Jure Leskovec. Learning the network structure of heterogeneous data via pairwise exponential markov random fields. In *AISTATS*, volume 54 of *Proceedings of Machine Learning Research*, pp. 1302–1310. PMLR, 2017.

Xingyue Pu, Tianyue Cao, Xiaoyun Zhang, Xiaowen Dong, and Siheng Chen. Learning to learn graph topologies. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021a. URL https://openreview.net/forum?id=ZqabiikWeyt.

Xingyue Pu, Siu Lun Chau, Xiaowen Dong, and Dino Sejdinovic. Kernel-based graph learning from smooth signals: A functional viewpoint. *IEEE Trans. Signal Inf. Process. over Networks*, 7:192–207, 2021b. doi: 10.1109/TSIPN.2021.3059995. URL https://doi.org/10.1109/TSIPN.2021.3059995.

Pradeep Ravikumar, Garvesh Raskutti, Martin J. Wainwright, and Bin Yu. Model selection in gaussian graphical models: High-dimensional consistency of l_1-regularized MLE. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou (eds.), *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pp. 1329–1336. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper/2008/hash/61f2585b0ebcf1f532c4d1ec9a7d51aa-Abstract.html.

Benjamin T. Rolfs, Bala Rajaratnam, Dominique Guillot, Ian Wong, and Arian Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In *NIPS*, pp. 1583–1591, 2012.

Emanuele Rossi, Federico Monti, Yan Leng, Michael M. Bronstein, and Xiaowen Dong. Learning to infer structures of network games. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18809–18827. PMLR, 2022.

Hawraa Salami, Bicheng Ying, and Ali H. Sayed. Social learning over weakly connected graphs. *IEEE Trans. Signal Inf. Process. over Networks*, 3(2):222–238, 2017. doi: 10.1109/TSIPN.2017.2668138. URL https://doi.org/10.1109/TSIPN.2017.2668138.

Michael Sejr Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *ESWC*, volume 10843 of *Lecture Notes in Computer Science*, pp. 593–607. Springer, 2018.

Santiago Segarra, Antonio G. Marques, Gonzalo Mateos, and Alejandro Ribeiro. Network topology inference from spectral templates. *IEEE Trans. Signal Inf. Process. over Networks*, 3(3):467–483, 2017.

Rasoul Shafipour, Santiago Segarra, Antonio G. Marques, and Gonzalo Mateos. Network topology inference from non-stationary graph signals. In *ICASSP*, pp. 5870–5874. IEEE, 2017.

Chuan Shi. *Heterogeneous Graph Neural Networks*, pp. 351–369. Springer Nature Singapore, Singapore, 2022. ISBN 978-981-16-6054-2. doi: 10.1007/978-981-16-6054-2_16. URL https://doi.org/10.1007/978-981-16-6054-2_16.

Harsh Shrivastava, Xinshi Chen, Binghong Chen, Guanghui Lan, Srinivas Aluru, Han Liu, and Le Song. GLAD: learning sparse graph recovery. In *ICLR*. OpenReview.net, 2020.

Martin Slawski and Matthias Hein. Estimation of positive definite m-matrices and structure learning for attractive gaussian markov random fields. *Linear Algebra and its Applications*, 473:145–179, 2015.

Ben Taskar, Pieter Abbeel, Ming-Fai Wong, and Daphne Koller. Relational markov networks. 2007.

Dorina Thanou, Xiaowen Dong, Daniel Kressner, and Pascal Frossard. Learning heat diffusion graphs. *IEEE Trans. Signal Inf. Process. over Networks*, 3(3):484–499, 2017.

Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. Social structure of facebook networks. *CoRR*, abs/1102.2166, 2011.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (eds.), *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*, pp. 165–174. ACM, 2019. doi: 10.1145/3331184.3331267. URL https://doi.org/10.1145/3331184.3331267.

Xiao Wang, Deyu Bo, Chuan Shi, Shaohua Fan, Yanfang Ye, and Philip S. Yu. A survey on heterogeneous graph embedding: Methods, techniques, applications and sources. *CoRR*, abs/2011.14867, 2020.

Yuling Wang, Hao Xu, Yanhua Yu, Mengdi Zhang, Zhenhao Li, Yuji Yang, and Wei Wu. Ensemble multi-relational graph neural networks. In *IJCAI*, pp. 2298–2304. ijcai.org, 2022.

Rongzhe Wei, Haoteng Yin, Junteng Jia, Austin R. Benson, and Pan Li. Understanding non-linearity in graph neural networks from the bayesian-inference perspective. In *NeurIPS*, 2022.

Wang Xiao, Ji Houye, Shi Chuan, Wang Bai, Cui Peng, Yu P., and Ye Yanfang. Heterogeneous graph attention network. *WWW*, 2019.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1): 19–35, 2007.

Kamilia Zaripova, Luca Cosmo, Anees Kazi, Seyed-Ahmad Ahmadi, Michael M. Bronstein, and Nassir Navab. Graph-in-graph (gig): Learning interpretable latent graphs in non-euclidean domain for biological and healthcare applications. *Medical Image Anal.*, 88:102839, 2023.

Muhan Zhang and Yixin Chen. Inductive matrix completion based on graph neural networks. In *ICLR*. OpenReview.net, 2020.

Dengyong Zhou and Bernhard Schölkopf. A regularization framework for learning from graph data. In *ICML 2004 Workshop on Statistical Relational Learning and Its Connections to Other Fields (SRL 2004)*, pp. 132–137, 2004.

Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. *Semi-supervised learning: From Gaussian fields to Gaussian processes*. School of Computer Science, Carnegie Mellon University, 2003.

A Proof

A.1 Derivation of the joint likelihood for vanilla graph structure learning problem

Tracing back to Zhu et al. (2003) and Lake & Tenenbaum (2010), the node-wise and pair-wise potential are selected to be $\psi_1(\mathbf{y}_v;\theta_1) = \exp(-(d_v+\nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2)$ with $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm and $\psi_2(\mathbf{y}_u,\mathbf{y}_v;\theta_2) = \exp(w_{uv}\mathbf{y}_u^\top\mathbf{y}_v)$ respectively. This leads to a colored multi-variate Gaussian on each column of the hidden variable $\mathbf{Y}_{:k}$ with zero mean and precision matrix $\Lambda = \mathbf{L} + \nu\mathbf{I}$ where $\mathbf{I}$ is the identity matrix. The overall likelihood function for $\mathbf{Y}$ is then,

$$\begin{aligned} P(\mathbf{Y};\mathbf{W}) &= \frac{1}{Z(\mathbf{W})}\prod_v \psi_1(\mathbf{y}_v)\prod_{v,u}\psi_2(\mathbf{y}_v,\mathbf{y}_u) \\ &= \frac{1}{Z(\mathbf{W})}\prod_v \exp(-(d_v+\nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2)\prod_{v,u}\exp(w_{uv}\mathbf{y}_u^\top\mathbf{y}_v) \\ &= \frac{1}{Z(\mathbf{W})}\exp\left\{-\frac{1}{2}\operatorname{tr}(\mathbf{Y}^\top\Lambda\mathbf{Y})\right\} \end{aligned} \tag{26}$$

where tr is the trace operator. It is noted that the likelihood function differs from a simple multivariate Gaussian as $\mathbf{Y}\in\mathbb{R}^{N\times C}$ is a matrix instead of a vector. Without loss of interpretability, we denote the concatenation of two label vectors as $\mathbf{y}\equiv\operatorname{cat}(\mathbf{y}_v,\mathbf{y}_u)\in\mathbb{R}^{2C}$ and consider Λ_{uv} as an entry of the precision matrix. The joint distribution for the label vectors on a pair of nodes, $\mathbf{y}_v$ and $\mathbf{y}_u$, becomes,

$$P(\mathbf{y}\mid\Lambda) = \frac{1}{Z}\exp\left\{-\frac{1}{2}(\mathbf{y}^\top\begin{bmatrix}(d_v+\nu)\mathbf{I} & -\Lambda_{uv}\mathbf{I}\\ -\Lambda_{vu}\mathbf{I} & (d_v+\nu)\mathbf{I}\end{bmatrix}\mathbf{y})\right\}, \tag{27}$$

Then, the observable variable $\mathbf{X}$ is emitted from $\mathbf{Y}$, with its individual row $\mathbf{x}_v$ depending only on $\mathbf{y}_v$, namely,

$$P(\mathbf{X}\mid\mathbf{Y};\mathbf{V}) = \prod_{i=1}^{N} P(\mathbf{x}_v\mid\mathbf{y}_v;\mathbf{V}). \tag{28}$$

To transit from the hidden variable $\mathbf{Y}$ to $\mathbf{X}$, a commonly used assumption for the emission function is also considering a multi-variate Gaussian Liu et al. (2014), where,

$$P(\mathbf{x}_v\mid\mathbf{y}_v,\Theta)\sim\mathcal{N}(\mathbf{V}\mathbf{y}_v,\Sigma_{\mathbf{x}}). \tag{29}$$

We denote $\mathbf{V}$ as the linear transformation matrix and Σ_x the marginal covariance matrix for $\mathbf{x}_v$, which is shared across all nodes $v\in\mathcal{V}$. Thus, by marginalizing out $\mathbf{y}_v$, we have the likelihood function for $\mathbf{x}\equiv\operatorname{cat}(\mathbf{x}_v,\mathbf{x}_u)$ as

$$\begin{aligned} &P(\mathbf{x}\mid\Theta) = \int P(\mathbf{x}\mid\mathbf{y})P(\mathbf{y}\mid\Theta)d\mathbf{y} \\ &\text{and } \mathbf{x}\mid\Theta\sim\mathcal{N}(0,\Sigma_x+\mathbf{V}\begin{bmatrix}(d_v+\nu)\mathbf{I} & -\Lambda_{uv}\mathbf{I}\\ -\Lambda_{vu}\mathbf{I} & (d_u+\nu)\mathbf{I}\end{bmatrix}^{-1}\mathbf{V}^\top). \end{aligned} \tag{30}$$

To further simplify the notation, we make the first assumption that the determinant of covariance matrix on $\mathbf{X}$ marginal is relatively small to the colored multi-variate Gaussian distribution that emits $\mathbf{Y}$, i.e., $\det(\Sigma_{\mathbf{x}})\ll\det(\Lambda^{-1})$. This assumption assures that the marginal noise on $\mathbf{x}$ is not so strong. Thus, the joint likelihood of the signal $\mathbf{X}$ on a graph follows,

$$P(\{\mathbf{x}_v\}\mid\Theta) = \frac{1}{Z}\prod_v\exp(-(\nu+d_v)\|\mathbf{V}^\dagger\mathbf{x}_v\|_{\mathrm{F}}^2)\cdot\prod_{u,v}\exp\left\{w_{uv}\operatorname{tr}(\mathbf{x}_u^\top\mathbf{V}^{\dagger\top}\mathbf{V}^\dagger\mathbf{x}_v)\right\}, \tag{31}$$

where $\mathbf{V}^\dagger$ is the Moore–Penrose Pseudo inverse of matrix $\mathbf{V}$. In the next chapter, we will show how such formulation can be used for graph structure learning.

A.2 Derivation of Graph Structure Learning training objective

According to eq. (3), the parameters contain the weighting matrix $\mathbf{W}$ (or Laplacian matrix equivalently), transformation matrix $\mathbf{V}^\dagger$ and a scalar ν (but we consider it as a hyperparameter). Current graph structure learning (GSL) algorithms Dong et al. (2015); Pu et al. (2021a); Kumar et al. (2019b) can be considered as a special case of the above MAP problem when $\mathbf{V}^\dagger$ is rectangular orthogonal (semi-orthogonal) Matrix. In such a scenario, the likelihood function of eq. (3) reduces to a same form as the Gaussian Graphical Model with,

$$P(\{\mathbf{x}_v\} \mid \mathbf{W}) = \frac{1}{Z} \prod_v \exp(-(\nu + d_v)\|\mathbf{x}_v\|_{\mathrm{F}}^2) \cdot \prod_{u,v} \exp\left\{w_{uv}(\mathbf{x}_u^\top \mathbf{x}_v)\right\}. \tag{32}$$

The only parameters left are the weight matrix $\mathbf{W} = \{w_{ij}\}_{i,j\in\{1:N\}}$ representing the graph structure. Then the problem becomes, $\mathbf{W}^* = \arg\max_{\mathbf{W}} \log P(\mathbf{X} \mid \mathbf{W}) + \log P(\mathbf{W})$. One would care about the log-likelihood of the signal $\mathbf{X}$ for convenient optimization,

$$\begin{aligned}
\log\left[P\left(\mathbf{X} \mid \mathbf{W}\right)\right] &= \sum_{v=1}^{N} -(d_v + \nu)\|\mathbf{x}_v\|_{\mathrm{F}}^2 + \sum_{u,v} w_{uv}\mathbf{x}_u^\top \mathbf{x}_v - \log Z(\mathbf{W}) \\
&= -\nu \sum_{u=1}^{N} \|\mathbf{x}_u\|_{\mathrm{F}}^2 - \sum_{u,v} \mathbf{L}_{uj}\mathbf{x}_u^\top \mathbf{x}_v - \log Z(\mathbf{W}) \\
&= -\nu\,\mathrm{tr}((\mathbf{X}^\top \mathbf{X}) - \mathrm{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) + \log\det(\mathbf{L} + \nu\mathbf{I}),
\end{aligned} \tag{33}$$

where the last step is derived given the only term related to the optimization goal in $Z(\mathbf{W})$ is the log-determinant of the covariance matrix. ν works as the factor balancing the node-wise and pair-wise effects. The pair-wise potentials are scaled by parameter $\mathbf{L}_{uv}$ which encodes the connection strength. In graph signal processing, the second term, in the quadratic form of the graph Laplacian, is called the "smoothness" of the signal on the graph. As shown in Zhou & Schölkopf (2004), it can be measured in terms of pair-wise difference of node signals,

$$\mathrm{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) = \frac{1}{2} \sum_{\{u,v\}} w_{uv}\|\mathbf{x}_u - \mathbf{x}_v\|_{\mathrm{F}}^2, \tag{34}$$

given that $\sum_v w_{uv} = \mathbf{D}_{uu}$. In other words, if two signal vectors x_u and x_v from a smooth set reside on two well-connected nodes with large w_{uv}, they are expected to have a small dissimilarity.

However, the computationally demanding log-determinant term makes it difficult to solve Kalofolias (2016a). Thus, researcher in Egilmez et al. (2017) consider a relaxed optimization on the lower bound instead. Using $-\sum_v \log\left(\mathrm{diag}(\Lambda)_v\right) \leq -\log\det(\Lambda)$ and substituting the log-likelihood with eq. (6), we can further derive the optimization objective as,

$$\begin{aligned}
&\arg\max_{\mathbf{W}} \log P(\mathbf{X} \mid \mathbf{W}) + \log P(\mathbf{W}) \\
=&\arg\max_{\mathbf{L}} -\nu\,\mathrm{tr}((\mathbf{X}^\top \mathbf{X}) - \mathrm{tr}(\mathbf{X}^\top \mathbf{L}\mathbf{X}) + \mathbf{1}^\top \log\left(\mathrm{diag}(\mathbf{L} + \nu\mathbf{I})_v\right) - \log P(\mathbf{W}) \\
=&\arg\min_{\mathbf{W}} \sum_{ij} w_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathrm{F}}^2 + \mathbf{1}^\top \log\left(\mathbf{W} \cdot \mathbf{1}\right) - \log P(\mathbf{W}) \\
\equiv&\arg\min_{\mathbf{W}} S(\mathbf{X}, \mathbf{W}) + \Omega(\mathbf{W})
\end{aligned} \tag{35}$$

The training objective consists of graph-signal fidelity term $S(\mathbf{X}, \mathbf{W}) \equiv \sum_{ij} w_{ij}\|\mathbf{x}_i - \mathbf{x}_j\|_{\mathrm{F}}^2$ and a structural regularizer $\Omega(\mathbf{W}) \equiv \mathbf{1}^\top \log\left(\mathbf{W} \cdot \mathbf{1}\right) - \log P(\mathbf{W})$.

A.3 Derivation of the data-generating models for heterogeneous graphs

Specifically, $\psi_v\left(\mathbf{y}_v\right) = \exp\left(-\sum_v (d_v + \nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2\right)$ and the pair-wise potential $\psi_{uvr}\left(\mathbf{y}_u, \mathbf{y}_v, \mathbf{B}_r\right) = \exp\left\{w_{uvr}\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v\right\}$. d_v is the degree of node v defined as $d_v = \sum_{ru} w_{uvr}$. We consider the There are

multiple types of distribution that satisfy the form in eq. (10). For simplicity, we assume $P(\mathbf{y}_u, \mathbf{y}_v)$ is a joint Gaussian, which has,

$$\begin{aligned} &\text{cat}([\mathbf{y}_u, \mathbf{y}_v]) \mid w_{uvr}, \mathbf{B}_r \\ &\sim \mathcal{N}\left(\begin{bmatrix} \mu_{\mathbf{y}_v} \\ \mu_{\mathbf{y}_u} \end{bmatrix}, \begin{bmatrix} (d_u+\nu)\mathbf{I} & -w_{vur}\mathbf{B}_r \\ -w_{vur}\mathbf{B}_r^\top & (d_v+\nu)\mathbf{I} \end{bmatrix}^{-1} \right), \end{aligned} \tag{36}$$

where cat is the concatenation operator on two vectors that yield another vector. Given all the above necessary notations, we can now define the generation density of $\{\mathbf{y}_v\}$ in heterogeneous graphs as follows,

$$\begin{aligned} &P(\{\mathbf{y}_v\} \mid \mathbb{W}, \{\mathbf{B_r}\}) \\ &\propto \exp\left(-\sum_v (d_v+\nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2 + \sum_{u\neq v, r} w_{vur}\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v\right), \end{aligned} \tag{37}$$

To further link label variable $\mathbf{y}_v$ with the signal variable $\mathbf{x}_v$, we assume that the conditional distribution of $P(\mathbf{x}_v \mid \mathbf{y}_v)$ has a mean that is a linear function of $\mathbf{y}_v$, and a covariance that is independent of $\mathbf{y}_v$, which gives,

$$\mathbf{x}_v \mid \mathbf{y}_v \sim \mathcal{N}\left(\mathbf{x}_v \mid \mathbf{V}_{\phi(v)}(\mathbf{y}_v - \mu_{\mathbf{y}_v}), \Sigma_{\mathbf{x}_v}\right) \tag{38}$$

where we assume that the liner transformation matrix $\mathbf{V}_{\phi(v)} \in \mathbb{R}^{K\times|C_{\phi(u)}|}$ from $\mathbf{y}_v$ to $\mathbf{x}_v$ is determined by the node type $\phi(v)$ and simply consider $\Sigma_{\mathbf{x}_v} = \sigma^2\mathbf{I}$. Thus, the joint distribution $\mathbf{y}_v$ and $\mathbf{x}_v$ also follow a Gaussian distribution, which gives,

$$\begin{aligned} &\begin{bmatrix} \mathbf{x}_v \\ \mathbf{y}_v \end{bmatrix} \sim \\ &\mathcal{N}\left(\begin{bmatrix} 0 \\ \mu_{\mathbf{y}_v} \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathbf{x}_v} + \mathbf{V}_{\phi(v)}\Sigma_{\mathbf{y}_v}\mathbf{V}_{\phi(v)}^\top & -\Sigma_{\mathbf{y}_v}\mathbf{V}_{\phi(v)}^\top \\ -\mathbf{V}_{\phi(v)}\Sigma_{\mathbf{y}_v} & \Sigma_{\mathbf{y}_v} \end{bmatrix} \right). \end{aligned} \tag{39}$$

Until now we have derive the $P(\mathbf{x}_v \mid \mathbf{y}_v)$ and $P(\mathbf{y}_u, \mathbf{y}_v \mid \lambda_{uvr})$ where for simplicity the parameters are denoted as $\lambda_{uvr} = \{w_{uvr}, \mathbf{B}_r\}$. Without loss of interpretability, we denote $\mathbf{x} \equiv \text{cat}(\mathbf{x}_v, \mathbf{x}_u)$ and $\mathbf{y} \equiv \text{cat}(\mathbf{y}_v, \mathbf{y}_u)$. We can apply Bayesian rule and obtain the conditional distribution of $P(\mathbf{x} \mid \lambda_{uvr})$ also being a Gaussian with the following joint likelihood function,

$$P(\mathbf{x} \mid \lambda_{uvr}) \propto \exp\left\{-\frac{1}{2}\,\text{tr}(\mathbf{x}^\top \Lambda_{\mathbf{x}|\Theta}\mathbf{x})\right\}, \tag{40}$$

where the mean $\mu_{x|\Theta} = 0$ and the precision matrix,

$$\Lambda_{\mathbf{x}|\Theta} = (\Sigma_{\mathbf{x}} + \mathbf{V}\Lambda_{\mathbf{y}|\Theta}\mathbf{V}^\top)^{-1} \tag{41}$$

with $\mathbf{V} = \left[\mathbf{V}_{\phi(v)}, \mathbf{V}_{\phi(u)}\right]$. If we consider the variance of $\mathbf{x}$ is relatively small compared to the graph emission matrix $\sigma_{\mathbf{y}}$, we obtain,

$$\begin{aligned} &\Sigma_{\mathbf{x}|\lambda_{uvr}}^{-1} = (\Sigma_{\mathbf{x}} + \mathbf{V}\Sigma_{\mathbf{y}|\lambda_{uvr}}\mathbf{V}^\top)^{-1} \approx \\ &\left(\begin{bmatrix} \mathbf{U}_{\phi(u)} \\ \mathbf{U}_{\phi(v)} \end{bmatrix}^\top \begin{bmatrix} (d_u+\nu)\mathbf{I} & -w_{vur}\mathbf{B}_r \\ -w_{vur}\mathbf{B}_r^\top & (d_v+\nu)\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{\phi(u)} \\ \mathbf{U}_{\phi(v)} \end{bmatrix} \right) \end{aligned} \tag{42}$$

where $\mathbf{U}_{\phi(u)} \in \mathbb{R}^{|C_{\phi(u)}|\times K}$ is the Moore–Penrose Pseudo inverse of matrix $\mathbf{V}_{\phi(u)}$ which satisfies $\mathbf{V}_{\phi(u)}\mathbf{U}_{\phi(u)}\mathbf{V}_{\phi(u)} = \mathbf{V}_{\phi(u)}$ and $\mathbf{U}_{\phi(u)}\mathbf{V}_{\phi(u)}\mathbf{U}_{\phi(u)} = \mathbf{U}_{\phi(u)}$. For the details of such a pseduo inverse method, we refer to Barata & Hussein (2011) for a tutorial. With such a derivation, we can get

$$\begin{aligned} \Sigma_{\mathbf{x}|\lambda_{uvr}}^{-1} &= -(d_u+\nu)\mathbf{U}_{\phi(u)}^\top\mathbf{U}_{\phi(u)} - (d_v+\nu)\mathbf{U}_{\phi(v)}^\top\mathbf{U}_{\phi(v)} \\ &\quad + 2w_{uvr}\mathbf{U}_{\phi(u)}^\top\mathbf{B}_r\mathbf{U}_{\phi(v)}. \end{aligned} \tag{43}$$

We then combine **??** and **??** to obtain the overall likelihood for the whole graph signals,

$$\begin{aligned} P(\{\mathbf{x}_v\} \mid \mathbb{W}, \{\mathbf{B}_r\}) &\propto \prod \exp\left\{-\frac{1}{2}\operatorname{tr}(\mathbf{x}^\top \Sigma^{-1}_{\mathbf{x}|\lambda_{uvr}}\mathbf{x})\right\} \\ &= \prod_v \exp(-(\nu+d_v)\|\mathbf{U}_{\phi_v}\mathbf{x}_v\|_{\mathrm{F}}^2) \\ &\cdot \prod_{u,v,r} \exp\left\{w_{uvr}\operatorname{tr}(\mathbf{x}_u^\top \mathbf{U}_{\phi(u)}^\top \mathbf{B}_r \mathbf{U}_{\phi(v)}\mathbf{x}_v)\right\} \end{aligned} \tag{44}$$

We further discuss the likelihood function in

A.4 The derivation of relation embedding update step in HGSL optimization

Here we gave a detailed analysis of algorithm 1 when updating relation embeddings. We first repeat the training objective here,

$$\arg\min_{\{\mathbf{e}_r\}} \quad \sum_{vur} w_{vur}\|\mathbf{e}_r \odot (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 + \beta' w_{vur}\|\mathbf{e}_r\|_{\mathrm{F}}^2 + \Omega(\mathbf{E}). \tag{45}$$

The training objective is derived as

$$\begin{aligned} &\arg\min_{\mathbf{e}_r} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r \circ (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 \\ =&\arg\min_{\mathbf{e}_r} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\| \circ \|\mathbf{x}_v - \mathbf{x}_u\|_{\mathrm{F}}^2 \\ =&\arg\min_{\mathbf{e}_r} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\| \circ (\|\mathbf{x}_v\|_{\mathrm{F}}^2 - 2\mathbf{x}_v^T \cdot \mathbf{x}_u + \|\mathbf{x}_u\|_{\mathrm{F}}^2) \end{aligned} \tag{46}$$

As $\mathbf{x}_v$ is a constant for all $v \in \mathcal{V}$ in the optimization of e_r, normalization of the $\mathbf{x}_v$ does not impact the optimization problem, which gives

$$\begin{aligned} &\arg\min_{\mathbf{e}_r} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\| \circ (\|\mathbf{x}_v\|_{\mathrm{F}}^2 - 2\mathbf{x}_v^T \cdot \mathbf{x}_u + \|\mathbf{x}_u\|_{\mathrm{F}}^2) \\ =&\arg\min_{\mathbf{e}_r} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\| \circ (2 - 2\frac{\mathbf{x}_v^T \cdot \mathbf{x}_u}{\|\mathbf{x}_u\|_{\mathrm{F}}^2 \cdot \|\mathbf{x}_v\|_{\mathrm{F}}^2}) \\ =&\arg\max_{\mathbf{e}_r} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\| \circ (\frac{\mathbf{x}_v^T \cdot \mathbf{x}_u}{\|\mathbf{x}_u\|_{\mathrm{F}}^2 \cdot \|\mathbf{x}_v\|_{\mathrm{F}}^2} - 1) \end{aligned} \tag{47}$$

We further add two regularizers in the optimization problem, l_2 norm on $\mathbf{e_r}$ and a node degree regularizer promoting $\sum \mathbb{W}_{uvr}\|\mathbf{e_r}\|$ to be non-zero. The regularizers can avoid arbitrary solutions and give

$$\begin{aligned} \arg\max_{\mathbf{e}_r} &\sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\|\mathbf{e}_r\| \circ (\frac{\mathbf{x}_v^T \cdot \mathbf{x}_u}{\|\mathbf{x}_u\|_{\mathrm{F}}^2 \cdot \|\mathbf{x}_v\|_{\mathrm{F}}^2} - 1) \\ &- \alpha \sum_r \|\mathbf{e}_r\|_{\mathrm{F}}^2 + \beta \sum_{\{v,u,r\}\in\mathcal{E}} \|w_{vur}\mathbf{e}_r\| \end{aligned} \tag{48}$$

With the properties that w_{uvr} and e_r are non-negative, we take the derivative of training objective and set it to 0, which yields a solution for us:

$$\mathbf{e}_r = \frac{1}{2\alpha} \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur}\left(\frac{\mathbf{x}_v^T \cdot \mathbf{x}_u}{\|\mathbf{x}_u\|_{\mathrm{F}}^2 \cdot \|\mathbf{x}_v\|_{\mathrm{F}}^2} + \beta - 1\right) \tag{49}$$

Note that we can only consider the sum over a subset of edges, $\mathcal{E}'$, when updating the $\mathbf{e}_r$. If we select $\alpha = 1/2$ and $\beta = 1$, we get exactly eq. (21).

B Extension of the algorithm

B.1 Learning the heterogeneous graphs when the node labels are observable.

When labels are observable, such as in IMDB and ACM dataset Fu et al. (2020), we wish to learn $\mathbb{W}$ and $\{\mathbf{B}_r\}$ directly from the labels $\{\mathbf{y}_v\}_{v\in\mathcal{V}}$. To this end, we consider the HGSL problem as a MAP estimation problem parameterized by eq. (24) and define $\Omega(\cdot)$ as the negative log prior, which gives,

$$\begin{aligned} &\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \log P(\mathbb{W}, \{\mathbf{B}_r\} \mid \{\mathbf{y}_v\}) \\ =&\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \log P(\{\mathbf{y}_v\} \mid \mathbb{W}, \{\mathbf{B}_r\}) - \Omega(\mathbb{W}) - \Omega(\{\mathbf{B}_r\}). \end{aligned} \tag{50}$$

The log-likelihood follows in eq. (24), thus we derive the training objective as,

$$\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \sum_{u\neq v,r} (\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v) \cdot w_{uvr} - \sum_v (d_v + \nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2 - \Omega(\mathbb{W}) - \Omega(\{\mathbf{B}_r\}). \tag{51}$$

Given that $\|\mathbf{B}_r\|_1 = \bar{d}/|\mathcal{V}|$, we can further divide first two terms as,

$$\begin{aligned} &-\sum_v (d_v + \nu)\|\mathbf{y}_v\|_{\mathrm{F}}^2 + \sum_{\{u\neq v,r\}\in\mathcal{E}} w_{vur}\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v \\ =&-\nu\sum_v \|\mathbf{y}_v\|_{\mathrm{F}}^2 + \sum_{u\neq v,r} w_{vur}\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v - \sum_v d_v\|\mathbf{y}_v\|_{\mathrm{F}}^2 \\ =&-\sum_{u,v,r} w_{uvr}\|\operatorname{vec}(\mathbf{B}_r) \odot (\mathbf{y}_u \otimes \mathbf{1}_v - \mathbf{y}_v \otimes \mathbf{1}_u)\|_{\mathrm{F}}^2. \\ &-\nu\sum_v \|\mathbf{y}_v\|_{\mathrm{F}}^2 \end{aligned} \tag{52}$$

Here vec is the vectorization function that flattens the matrix into a long vector, $\otimes$ is the Kronecker product, $\odot$ is the element-wise product, and $\mathbf{1}_v$ is the all-1 vector that has the same size as y_v. The node-wise effects solely related to $\mathbf{y}_v$ do not affect the optimization over $\mathbb{W}$ and $\{\mathbf{B}_r\}$ so can be omitted. Thus, we get,

$$\begin{aligned} \arg\min_{\mathbb{W},\{\mathbf{B}_r\}} &\sum_{u,v,r} \operatorname{vec}(\mathbf{B}_r)\|\mathbf{y}_u \otimes \mathbf{1}_v - \mathbf{y}_v \otimes \mathbf{1}_u\|_2^2 \cdot w_{uvr} \\ &+ \Omega(\mathbb{W}) + \Omega(\{\mathbf{B}_r\}). \end{aligned} \tag{53}$$

C When can we learn the graph? Connecting Smoothness and Homophily

In the vanilla GSI problem, it is widely acknowledged that "On a homophily graph, minimizing the smoothness could lead to a meaningful graph structure", just like what we did in eq. (9). However, mathematically why we have such a conclusion is not yet clear. In more complex graphs, e.g. heterogeneous graphs, the definition of smoothness and homophily is never properly stated. In this section, we gave the first attempt at understanding how the homophily concepts and smoothness assumption would impact the GSI algorithms from the perspective of stochastic blocking models. We then move to heterogeneous cases to see how we can interpret the homophily in heterogeneous graphs through the H2MN, and derive the condition that our algorithm, proposed in algorithm 1, will converge to a meaningful solution.

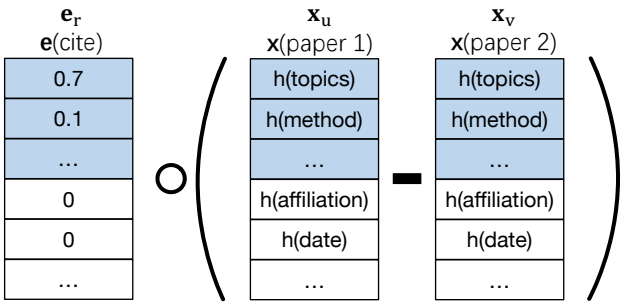


Figure 6: Visualization of the generalized smoothness.

C.1 Generalized Smoothness for Heterogeneous Graph Signals

According to appendix D.7, we realize that solving HGSL requires: 1) focusing on specific dimensions of signals to measure smoothness and 2) weighing these dimensions depending on the relation type. To this end, we propose a *generalized smoothness* on the heterogeneous graph by specifying a *learnable* relation embedding $\mathbf{e}_r \in \mathbb{R}^K, \forall r \in \mathcal{R}$ that has the same dimension size as node signals. The idea is to introduce a *reweighted smoothness* scheme by first measuring dimension-wise smoothness, and then integrating it by emphasizing specific signal dimensions according to r. This process is illustrated in fig. 6. Using academic networks again as an example, while determining whether a 'Cite'-typed edge should be formed between two paper nodes, the model should focus on the signal dimensions that represent the 'topics', but weigh less the irrelevant ones like 'affiliation' or 'date'. In other words, larger weights should be assigned to the 'topics' dimension and smaller ones to others. According to this, our new generalized smoothness is,

$$\begin{aligned} S &= \sum_{\{v,u,r\}\in\mathcal{E}} w_{vur} \|\mathbf{e}_r \circ (\mathbf{x}_v - \mathbf{x}_u)\|_{\mathrm{F}}^2 \\ &= \langle \mathbb{W}, \|(\mathbf{X}\otimes\mathbf{1} - \mathbf{1}\otimes\mathbf{X})\otimes\mathbf{E}\|_F^2 \rangle \end{aligned} \tag{54}$$

where $\circ$ is the element-wise product and $\|\cdot\|_F$ is the Frobenius norm along the last dimension which collapses the tensor from dimension 4 to 3. $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_{|\mathcal{V}|}]^{\mathbf{T}} \in \mathbb{R}^{|\mathcal{V}|\times K}$, $\mathbf{E} = [\mathbf{e}_1, ..., \mathbf{e}_{|\mathcal{R}|}]^{\mathbf{T}} \in \mathbb{R}^{|\mathcal{R}|\times K}$, and $\mathbf{1} \in \mathbb{R}^{|\mathcal{V}|\times K}$ is an all-one matrix. $\langle\cdot,\cdot\rangle$ and $\otimes$ are the tensor inner product and outer product, respectively. Different from the original smoothness, if node v and u are connected w.r.t. r, the contribution of each signal dimension to the overall smoothness is reweighted by $\mathbf{e}_r$. Intuitively, a heterogeneous graph is thought to be smooth if strongly connected nodes (with larger w_{vur}) have similar signal values in the dimensions emphasized by $\mathbf{e}_r$ for the relation r.

C.2 Connecting homophily and Graph Structure Inference

We will first start the connection between the generative process of homogeneous graphs and the graph structure inference algorithms. Then we slowly move toward the heterogeneous graph cases. Here we use the Bernoulli Stochastic Blocking Models as an example.

The concept of homophily in networks was first proposed in McPherson et al. (2001), suggesting that a connection between similar entities occurs at a higher probability than among dissimilar entities. The most accurate formulation in the statistical community would be having a higher probability of observing an edge between two nodes belonging to the same communities. Such a concept gives,

$$P(\mathbf{W}_{ij} = 1 \mid Y_i = Y_j) > P(\mathbf{W}_{ij} = 1 \mid Y_i \neq Y_j), \tag{55}$$

where Y_i and Y_j are the labels for nodes i and j. For simplicity, we denote $P(\mathbf{W}_{ij} = 1 \mid Y_i = Y_j) = p$ and $P(\mathbf{W}_{ij} = 1 \mid Y_i \neq Y_j) = q$. Such a concept can be easily modeled by stochastic blocking models, with $p > q$.

Before diving deeper, we first consider the case when $\mathbf{B}$ is a 2×2 matrix encoding only the "homophily" and "Heterophily", which gives the following optimization objective:

$$\begin{aligned}
&\arg\max_{\mathbf{W}} \log P(\mathbf{Y} \mid \mathbf{W}) + \log P(\mathbf{W}) \\
&= \arg\max_{\mathbf{W}} \sum_{i \le j} \mathbf{y}_i^\top \mathbf{B} \mathbf{y}_j \cdot \mathbf{W}_{ij} - \beta \|\mathbf{W}\|_1 \\
&= \arg\max_{\mathbf{W}} \sum_{\{i,j\}: c(i)=c(j)} p\mathbf{W}_{ij} + \sum_{\{i,j\}: c(i)\neq c(j)} q\mathbf{W}_{ij} - \beta \|\mathbf{W}\|_1 \\
&= \arg\max_{\mathbf{W}} \quad (p-q) \sum_{\{i,j\}: c(i)=c(j)} \mathbf{W}_{ij} + (q-\beta)\|\mathbf{W}\|_1.
\end{aligned} \tag{56}$$

Here we only consider the ℓ_1 regularizer. If we consider the re-scaled $\mathbf{B} = \bar{d}\bar{\mathbf{B}}/N$ with $d \ll N$, the p and q should be both smaller. In such a situation, the training objective can be decomposed into two parts. The first part suggests that the algorithm should put edges between homophily nodes, and the second encourages the sparsity of the solution. Then to avoid trivial solution (i.e. simply maximizing $\|\mathbf{W}\|_1$ or setting $\|\mathbf{W}\|_1 = 0$) for the problem, we need to have $p - q > 0$ and $q - \beta < 0$, i.e the graph is homophily with $p > q$. Then the solution becomes a trade-off game between the two components.

C.3 Homophily in Heterogeneous Graphs

Similar to the homogeneous graphs, we wish to understand what setting of $\mathbf{B}_r, r \in \mathcal{R}$ could give a guarantee on the optimization problem of heterogeneous graph structure inference. Let us reconsider the optimization problem related to $\mathbf{Y}$ introduced in eq. (16) but ignore the regularizer for now. Considering the Cartesian product of node type combination space $t \in \mathcal{T} = \mathcal{A} \times \mathcal{A}$, a possible relation type as $r \in \mathbb{R}_{\phi(u)\phi(v)}$ and then we denote an element of possible node label combinations within two node types as $\tau \in \mathcal{Y}_{\phi(u)} \times \mathcal{Y}_{\phi(v)}$, We can derive the sum as follows

$$\begin{aligned}
\hat{\mathbb{W}} &= \arg\max_{\mathbb{W}} \sum_{u,v,r} \mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v \mathbb{W}_{uvr} - \beta \|\mathbb{W}\|_1 \\
&= \arg\max_{\mathbb{W}} \sum_{\{\phi(u),\phi(v)\} \in \mathcal{T}} \sum_{\tau, r} \sum_{\{u,v\}: \{c_u, c_v\} = \tau} p_{r,\tau} \mathbb{W}_{uvr} - \beta \|\mathbb{W}\|_1
\end{aligned} \tag{57}$$

where $p_{r,\tau}$ is an element in the $\mathbf{B}$. The sum is over three parts: 1) summation over all possible node types combination 2) over all relation types within one node type combination (in many scenarios this is only one type) 3) over all possible node labels combination.

Then we define a "representative" connection between two nodes within a combination of node types, that gives the highest probability of connection $\hat{p}_{r^*,\tau^*}$, we can derive the summation as,

$$\begin{aligned}
&\sum_{\{\phi(u),\phi(v)\} \in \mathcal{T}} (p_{r^*,\tau^*} - \sum_{\substack{\tau \neq \tau^*, \\ r \neq r^*}} p_{r,\tau}) \sum_{\substack{r = r^*, \\ \{c_u, c_v\} = \tau^*}} \mathbb{W}_{uvr} + \sum_{\substack{\tau \neq \tau^*, \\ r \neq r^*}} p_{r,\tau} (\sum_{\substack{r = r^*, \\ \{c_u, c_v\} = \tau^*}} \mathbb{W}_{uvr} + \sum_{\substack{r \neq r^*, \\ \{c_u, c_v\} \neq \tau^*}} \mathbb{W}_{uvr}) - \beta \|\mathbb{W}\|_1 \\
&= \sum_{\{\phi(u),\phi(v)\} \in \mathcal{T}} (p_{r^*,\tau^*} - \sum_{\tau \neq \tau^*, r \neq r^*} p_{r,\tau}) \sum_{\substack{r = r^*, \\ \{c_u, c_v\} = \tau^*}} \mathbb{W}_{uvr} + (\sum_{\tau \neq \tau^*, r \neq r^*} p_{r,\tau} - \beta) \|\mathbb{W}\|_1
\end{aligned} \tag{58}$$

Thus, the sufficient condition for the optimization problem to have meaningful solutions is that $\forall r, \tau$,

$$p_{r,\tau^*} - \sum_{\tau \neq \tau^*} p_{r,\tau} > 0, \tag{59}$$

We can further prove that when the true probability $p_{r^*,\tau^*} > \sum_{r \neq r^*, \tau \neq \tau^*} p_{r,\tau}$, the above inequality is satisfied. The proof is shown in appendix D.5. Intuitively, this says that the probability of having a representative connection is significantly greater than the total probability of having other connections. The property inspires us to define a "Homophily" on heterogeneous graphs:

Defnition 1 ***Homophily on Heterogeneous graphs.*** *A heterogeneous graph is said to be homophily, if, for each pair of node types, there exists a representative connection between two specific classes of nodes, denoted as r^*, τ^*, which has a significantly larger probability of connection compared to the total probability of having other connections. Following the statement in eq.* (22)*, the homophily degree (HD) of a heterogeneous graph is given as,*

$$\begin{aligned} \mathrm{HD}\left(\phi(u), \phi(v)\right) &= \frac{p_{r^*,\tau^*}}{\sum_{r\neq r^*, \tau\neq\tau^*} p_{r,\tau}} \\ &\approx \frac{\text{Maximum \#edges within a type of connection}}{\text{Total \#edges within this type of connection}} \end{aligned} \tag{60}$$

However, due to the high complexity of heterogeneous graph structure data, while processing the datasets, people usually retain one type of "Target node" and record the label of that node type, and ignore the others. This brought issues when directly applying the definition 1. Thus, we further derive a relaxed condition for homophily.

Defnition 2 ***Relaxed Homophily on Heterogeneous graphs.*** *We consider taking a random walk starting from node v with type $\phi(v)$ and label $c(v)$, the heterogeneous graph is said to be homophily if there exists a representative node label c_v^* for each type of node $\phi(v)$, that the random walk starting from c_v^* has a large probability of reaching another same-typed node with same label c_v^*. With the formal mathematical statement in eq.* (61)*.*

Derive the random-walk form of such a homophily.

$$\hat{p}_{r^*,\tau^*} - \sum_{\tau,r} p_{r,\tau} = \log \frac{\mathbf{B}_{r^*,\tau^*}}{1-\mathbf{B}_{r^*,\tau^*}} - \sum_{\tau,r} \frac{\mathbf{B}_{r,\tau}}{1-\mathbf{B}_{r,\tau}} > 0, \tag{61}$$

D Stochatic Blocking Models

In this paper, we consider undirected graphs with no self-edges, which yields symmetric $\mathbf{W}$. In random graph models, It is assumed that $\mathbf{W}_{ij}$ are independent Bernoulli variables for $i < j$, i.e.,

$$\begin{aligned} &\mathbf{W}_{ij} \sim \text{Bernoulli}\left(p_{ij}\right), \text{for } i<j \\ &p_{ij} \in (0,1), \mathbf{W}_{ji} = \mathbf{W}_{ij}, \mathbf{W}_{ii} = 0. \end{aligned} \tag{62}$$

If we assume $\forall\{i,j\} \in \mathcal{E}, p_{ij} = p$, this degenerates to the well-known Erdös-Rényi model Erdős et al. (2013), which could be too simple to be practical. A practical generalization is the Contextual Stochastic Block Model (C-SBM) Deshpande et al. (2018), which is a coupling of the standard stochastic block model (SBM) and a Gaussian Mixture model. It introduces a (symmetric) blocking matrix $\mathbf{B} \in \mathbb{R}^{C\times C}$, with $C \ll N$ being the number of clusters (label classes), and a clustering map $c : \{1, \cdots, N\} \to \{1, \cdots, C\}$, such that $p_{ij} = \mathbf{B}_{c(i),c(j)}$. In SBM, N nodes are grouped in C groups, with group labels given by map $c(\cdot)$, and we call these groups communities. In the literature analyzing graph neural networks, the communities would be the number of label classes for a node type. Thus, the probability of having an edge between node i and j is,

$$\mathbf{W}_{ij} \mid \mathbf{Y} \sim \text{Bernoulli}(\mathbf{y}_i^\top \mathbf{B}\mathbf{y}_j), \text{for } i \neq j \tag{63}$$

and $\mathbf{y}_i \in \mathbb{R}^C$ being the one-hot vector with only non-zero entry indicating its cluster[4]. We can further derive $\mathbf{y}_i^\top \mathbf{B}\mathbf{y}_j = \mathbf{B}_{c(i)c(j)}$ in the one-hot setting, but in mixed membership SBM, when $\mathbf{y}_i$ is a continuous non-zero vector, it is safe to use eq. (63). Thus, with $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_N]^\top$, the expected adjacency matrix can be expressed as,

$$\mathbb{E}(\mathbf{W}) = \mathbf{Y}^\top \mathbf{B}\mathbf{Y} - \mathrm{diag}\left(\mathbf{Y}^\top \mathbf{B}\mathbf{Y}\right). \tag{64}$$

[4] In semi-supervised graph machine learning scenarios, the clustering vector $\mathbf{Y}_i$ is usually the label matrix $\mathbf{Y}_i$ that we wish to predict, such as Wei et al. (2022).

As a side note, we dive a bit deeper into the blocking matrix $\mathbf{B}$. In traditional SBM, each column is summarized to 1 for normalization, assuming that the graph is dense. In the recent development of CSBM Deshpande et al. (2018), the probability is modulated by the sparsity degree of the graph, which has $\mathbf{B}_{modulated} = \bar{d} \cdot \mathbf{B}/N$, with $\bar{d}$ the average degree of the graph and N the number of nodes. With a bit of overuse of notation, we will rely on such a form and by default consider $\mathbf{B}$ as the modulated form.

Furthermore, the CSBM controls the generation of node features $\mathbf{X} \in \mathbb{R}^{N \times K}$ with a probability distribution $P(\mathbf{X} \mid \mathbf{Y}) = \Pi_{i=1}^{N} P(\mathbf{X}_i \mid \mathbf{Y}_i)$. It follows,

$$\mathbf{X}_i = \mathbf{Y}_i \mathbf{V}^\top + \epsilon, \tag{65}$$

where ϵ is the noise with variance σ_ϵ. The linear transformation matrix $\mathbf{U}^\top$ can be derived through a joint Gaussian distribution between $\mathbf{X}$ and $\mathbf{Y}$, and we define the inverse transformation as $\mathbf{Y}_i = \mathbf{X}_i \mathbf{U}^\top + \epsilon$. This definition is slightly different from the traditional SBM, and we refer to appendix D.2 for a detailed explanation.

It is noted that the formulation of CSBM makes node feature $\mathbf{X}$ and graph structure $\mathbf{W}$ conditional independent given the labels $\mathbf{Y}$. In appendix D.6, we will first introduce how such CSBM can motivate the task of graph structure inference. Then, we will introduce how we can extend such SBM models into the heterogeneous graph cases, which we name Heterogeneous Stochastic Blocking Models (HSBM). And we will show how such a model can further motivate a novel task called Heterogeneous Graph Structure Learning (HGSL).

D.1 Gaussian Contextual Stochastic Blocking Models

A practical generalization of such a random model is the Gaussian Stochastic Block Model (GSBM). We consider undirected graphs with no self-edges, which yields symmetric $\mathbf{W}$. For simplicity, we consider a Gaussian random graph, which assumes that $\mathbf{W}_{ij}$ are independent Gaussian variables for $i < j$, i.e.,

$$\begin{aligned} &\mathbf{W}_{ij} \sim \mathcal{N}\left(\mu_{ij}, \sigma^2\right), \\ &\text{s.t. } \mu_{ji} = \mu_{ij}, \mu_{ij} \in [0,1], i < j, \mathbf{W}_{ii} = 0. \end{aligned}$$

It introduces a (symmetric) blocking matrix $\mathbf{B} \in \mathbb{R}^{c \times c}$, with $k \ll n$, and a clustering map $C : \{1, \cdots, n\} \to \{1, \cdots, c\}$. In GSBM, n nodes are grouped in k groups, with group labels given by map C, and we call these groups communities. By definition, there are $c(c+1)/2$ parameters to estimate in SBM. Thus, for each element,

$$\mathbf{W}_{ij} \mid \mathbf{Y} \sim \mathcal{N}(\mathbf{y}_i^\top \mathbf{B} \mathbf{y}_j, \sigma^2), \text{for } i \neq j, \tag{66}$$

and $\mathbf{y}_i \in \mathbb{R}^c$ being the one-hot vector with only non-zero entry indicating its cluster[5]. Thus, with $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, ...\mathbf{y}_n]^\top$, the expected adjacency matrix can be expressed as,

$$E(\mathbf{W} \mid \mathbf{Y}) = \mathbf{Y}^\top \mathbf{B} \mathbf{Y} - \operatorname{diag}\left(\mathbf{Y}^\top \mathbf{B} \mathbf{Y}\right). \tag{67}$$

Furthermore, the Contextual GSBM Deshpande et al. (2018), a coupling of the GSBM with a Gaussian Mixture model, controls the generation of node features with a probability distribution $P(\mathbf{X} \mid \mathbf{Y}) = \Pi_{i=1}^{n} P(\mathbf{X}_i \mid \mathbf{Y}_i)$. With a prior $P(\mathbf{X}_i \mid \mathbf{Y}_i)$ follows a Gaussian distribution, which has mean $\theta^\top \mathbf{Y}_i$ and variance σ_X^2

D.2 Gaussian Process and Stochastic Blocking Models

If we consider the joint distribution of $P(\mathbf{X}, \mathbf{Y})$ as a Gaussian, the linear transformation will become invertible. It is possible to derive directly from a Gaussian process, which we assume that,

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathbf{X}} & \Sigma_{\mathbf{XY}} \\ \Sigma_{\mathbf{YX}} & \Sigma_{\mathbf{Y}} \end{bmatrix}\right). \tag{68}$$

[5] In semi-supervised graph machine learning scenarios, the clustering vector $\mathbf{Y}_i$ is usually the label matrix that we wish to predict, such as Wei et al. (2022).

This gives the posterior of $P(Y \mid X)$,

$$\begin{aligned}\mu_{\mathbf{Y}|\mathbf{X}} &= \mu_{\mathbf{Y}} + \Sigma_{\mathbf{YX}}\Sigma_{\mathbf{XX}}^{-1}(\mathbf{X} - \mu_{\mathbf{X}}) \\ \Sigma_{\mathbf{Y}|\mathbf{X}} &= \Sigma_{\mathbf{YY}} - \Sigma_{\mathbf{YX}}\Sigma_{\mathbf{XX}}^{-1}\Sigma_{\mathbf{XY}}\end{aligned} \tag{69}$$

We then assume that the prior for $\mathbf{X}$ and $\mathbf{Y}$ both have variance 1, then,

$$\begin{aligned}\mu_{\mathbf{Y}|\mathbf{X}} &= \mu_{\mathbf{Y}} + \Sigma_{\mathbf{YX}}(\mathbf{X} - \mu_{\mathbf{X}}) \\ \Sigma_{\mathbf{Y}|\mathbf{X}} &= \mathbf{I} - \Sigma_{\mathbf{YX}}\Sigma_{\mathbf{XY}}\end{aligned} \tag{70}$$

It is possible to adjust the variance to rescale such a linear transformation. For simplicity, in this thesis, we assume that the node signal $\mathbf{X}$ is simply a deterministic linear transformation of the node label $\mathbf{Y}$, i.e. $\mathbf{X} = \mathbf{U}\mathbf{Y} + \text{bias}$, and $\mathbf{Y} = \mathbf{V}\mathbf{X} + \text{bias}$.

D.3 More about Heterogeneous Graph Stochastic Blocking Models

However, in the analysis, this might not be always easy. For simplicity, we still turn to Gaussian SBM with,

$$\mathbb{W}_{uvr} \mid \mathbf{y}_u, \mathbf{y}_v, \mathbf{B}_r \sim \mathcal{N}(\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v, \sigma_\epsilon^2), \tag{71}$$

Model Complexity. The parameter complexity for the above model is $O(C^2 \times |\mathcal{R}| + N \times C + K \times C \times |\mathcal{A}|) = O((N + C \times |\mathcal{R}| + K \times |\mathcal{A}|) \times C)$, where C is the largest number of communities among all the communities, N is the number of nodes and $|\mathcal{R}|$ is the number of relations. The first part comes from the blocking matrices $\mathbf{B}_r, \forall r$, the second part from node labels, $\mathbf{Y}$, and the third part from the emission matrices $\mathbf{U}_{\phi(v)}, \forall \phi(v) \in \mathcal{A}$. In common heterogeneous graphs, the leading term will still be $N \times C$ as usually in the datasets $C \times |\mathcal{R}| + K \times |\mathcal{A}| \ll N$. So the complexity is not significantly greater than the common stochastic blocking models. But in multi-relational graphs when $|\mathcal{R}|$ grows quickly, or in the over-parameterization scenarios such that $N \ll K$, the scenario changes a lot with more parameters to be learned.

D.4 More about the generating process

To simplify the learning problem, we further define

Similarly, we can decompose the overall We can further assume the marginal (or prior?) distribution of x_v and y_v follows a Gaussian with zero mean and variance $\Sigma_{\mathbf{x}} = \Sigma_{\mathbf{y}} = \sigma^2\mathbf{I}$. In such a scenario, $\mathbf{y}$ can be viewed as a linear transformation from observed $\mathbf{x}$ plus noise, and vice versa.

It is natural to consider the linear transformation only depends on the node type, $\phi(v)$. Accordingly, we denote $\mathbf{V}_{\phi(v)} \equiv \Sigma_{\mathbf{xy}}\Sigma_{\mathbf{yy}}^{-1}$ and obtain,

$$\mathbf{x}_v = \mathbf{V}_{\phi(v)}\mathbf{y}_v + \epsilon_v, \tag{72}$$

with ϵ_v the Gaussian noise and $\mathbf{V}_{\phi(v)}$ the linear transformation matrix specified to $\phi(v)$. Similarly,

$$\mathbf{y}_v = \mathbf{U}_{\phi(v)}\mathbf{x}_v + \epsilon_v. \tag{73}$$

Equivalently,

$$\begin{aligned}P(\mathbf{x}_v \mid \mathbf{y}_v) \sim &\mathcal{N}\left(\mathbf{V}_{\phi(v)}\mathbf{y}_v, \sigma^2\mathbf{I} + \sigma^{-2}\Sigma\mathbf{xy}\Sigma\mathbf{yx}\right) \\ = &\mathcal{N}\left(\mathbf{V}_{\phi(v)}\mathbf{y}_v, \sigma^2(\mathbf{I} + \mathbf{V}_{\phi(v)} \cdot \mathbf{V}_{\phi(v)}^\top)\right)\end{aligned} \tag{74}$$

and,

$$\begin{aligned}P(\mathbf{y}_v \mid \mathbf{x}_v) \sim &\mathcal{N}\left(\mathbf{U}_{\phi(v)}\mathbf{x}_v, \sigma^2\mathbf{I} + \sigma^{-2}\Sigma\mathbf{yx}\Sigma\mathbf{xy}\right) \\ = &\mathcal{N}\left(\mathbf{U}_{\phi(v)}\mathbf{x}_v, \sigma^2(\mathbf{I} + \mathbf{U}_{\phi(v)}\mathbf{U}_{\phi(v)}^\top)\right).\end{aligned} \tag{75}$$

The details of derivation can be found in appendix D.2.

D.5 Proofs for properties related to Homphily

Theorem 1 *When $p^* > \sum_i p_i$, the inequality $\log \frac{p^*}{1-p^*} > \sum_i \log \frac{p_i}{1-p_i}$ holds.*

Consider the function $f(x) = \log \frac{x}{1-x}$, the above inequality is the same as showing $f(x) > \sum f(y_i)$ when $x > \sum y_i$. As $f(\cdot)$ is monotone increasing , $f(x) > f(\sum y_i)$ holds. Since $f(x)$ is concave within the support $[0, 1/2)$, $f(\sum y_i) \geq \sum f(y_i)$ holds with Jensen's inequality. Thus,

$$f(x) > f(\sum y_i) \geq \sum f(y_i) \text{ when } 0 < x, \{y_i\} < 1/2.$$

Replace x with p^* and y_i with p_i, we obtain Theorem 1 when $p^* < 1/2$. When $p^* > 1/2$, $\sum p_i < 1/2$, the left-hand side of the inequality is bigger than 0 and the right-hand side is smaller than 0, it is trivial to see the inequality hold.

Theorem 2 *Random walk and homophily.*

The concept of homophily in networks was first proposed in McPherson et al. (2001), suggesting that a connection between similar entities occurs at a higher probability than among dissimilar entities. The most accurate formulation in the statistical community would be having a higher probability of observing an edge between two nodes belonging to the same communities. Such a concept gives,

$$P(\mathbf{W}_{ij} = 1 \mid Y_i = Y_j) \geq P(\mathbf{W}_{ij} = 0 \mid Y_i \neq Y_j), \tag{76}$$

where Y_i and Y_j are the labels for nodes i and j.

Now we consider the maximum likelihood estimation of $P(\mathbf{W} \mid \mathbf{Y})$ which gives

$$\arg\max P(\mathbf{W} \mid \mathbf{Y}) = \arg\max \log P(\mathbf{W} \mid \mathbf{Y}) \tag{77}$$

the log-likelihood is then,

$$\begin{aligned}
&\log P(\mathbf{W} \mid \mathbf{Y}) = \log \Pi_{i \leq j} P(\mathbf{W}_{ij} \mid \mathbf{Y}_i, \mathbf{Y}_j, \mathbf{B}) \\
&= \log \Pi_{i \leq j} (\mathbf{Y}_i^\top \mathbf{B} \mathbf{Y}_j)^{\mathbf{W}_{ij}} (1 - \mathbf{Y}_i^\top \mathbf{B} \mathbf{Y}_j)^{1 - \mathbf{W}_{ij}} \\
&= \log \Pi_{i \leq j} (\mathbf{B}_{\phi(i)\phi(j)})^{\frac{1 + (2\mathbf{W}_{ij} - 1) \cdot (2\mathbf{Y}_i^\top \mathbf{Y}_j - 1)}{2}} \\
&\quad \cdot (1 - \mathbf{B}_{\phi(i)\phi(j)})^{\frac{1 - (2\mathbf{W}_{ij} - 1) \cdot (2\mathbf{Y}_i^\top \mathbf{Y}_j - 1)}{2}} \\
&= \frac{1}{2} \sum_{i \leq j} \log \frac{\mathbf{B}_{c_1 c_2}}{1 - \mathbf{B}_{c_1 c_2}} \cdot (2\mathbf{W}_{ij} - 1) \cdot (2\mathbf{Y}_i^\top \mathbf{Y}_j - 1) + \mathrm{C}
\end{aligned} \tag{78}$$

Thus, we further derive the optimization objective:

$$\begin{aligned}
&\arg\max_{\mathbf{W}} \log P(\mathbf{W} \mid \mathbf{Y}) \\
&= \arg\max_{\mathbf{W}} \sum_{i \leq j} \frac{\mathbf{B}_{ij}}{1 - \mathbf{B}_{ij}} (4\mathbf{W}_{ij} \cdot \mathbf{Y}_i^\top \mathbf{Y}_j - 2\mathbf{W}_{ij})
\end{aligned} \tag{79}$$

while $\mathbf{X}$ is a linear transformation of $\mathbf{Y}$. If we constrain $\mathbf{X}$ to be a unit vector, and we assume the graph is homophily with two classes, i.e. $\mathbf{B}_{ij} = p \geq 0.5$, the above optimization problem yields,

$$\begin{aligned}
&\arg\max_{\mathbf{W}} \log P(\mathbf{W} \mid \mathbf{X}) \\
&= \mathrm{argmin}_{\mathbf{W}} \sum_{i \leq j} (\alpha \mathbf{W}_{ij} \cdot (\mathbf{X}_i - \mathbf{X}_j)^2 + \beta \mathbf{W}_{ij}) \\
&= \mathrm{argmin}_{\mathbf{W}} \, \alpha \mathbf{X}^\top \mathbf{L} \mathbf{X} + \beta \|\mathbf{W}\|
\end{aligned} \tag{80}$$

The above derivation shows that the solution of maximum likelihood estimation of $\mathbf{W}$ on a homophily graph is exactly the optimization solution of minimizing smoothness with a ℓ_1 regularizer.

Note that the above derivation until eq. (89) does not assume the property of the graph. Thus, it is applicable to graphs with multiple communities other than homophily graphs. If we consider a heterogeneous graph with the following parameterization,

$$\begin{aligned} &\mathbb{W}_{uvr} = 1 \mid \mathbf{y}_u, \mathbf{y}_v, \mathbf{B}_r \sim \text{Bernoulli}(\mathbf{y}_u^\top \mathbf{B}_r \mathbf{y}_v), \\ &\mathbf{x}_i = \sqrt{\frac{\mu}{n}} \mathbf{U}_{\phi(i)}^\top \mathbf{y}_i + \frac{\epsilon}{\sqrt{p}}, \end{aligned} \tag{81}$$

Such a parametrization leads to a likelihood function as follows,

$$\begin{aligned} &P(\mathbb{W} \mid \mathbf{Y}) = \Pi_{u \le v, r} P(\mathbb{W}_{uvr} \mid \mathbf{Y}_u, \mathbf{Y}_v, \mathbf{B}_r) \\ &= \Pi_{u \le v} (\mathbf{Y}_u^\top \mathbf{B}_r \mathbf{Y}_v)^{\mathbb{W}_{uvr}} (1 - \mathbf{Y}_u^\top \mathbf{B}_r \mathbf{Y}_v)^{1 - \mathbb{W}_{uvr}} \\ &= \Pi_{u \le v} (\mathbf{B}_{r,\phi(u)\phi(v)})^{\frac{1 + (2\mathbb{W}_{uvr} - 1) \cdot (2\mathbf{Y}_u^\top \mathbf{Y}_v - 1)}{2}} \\ &\quad \cdot (1 - \mathbf{B}_{r,\phi(u)\phi(v)})^{\frac{1 - (2\mathbb{W}_{uvr} - 1) \cdot (2\mathbf{Y}_u^\top \mathbf{Y}_v - 1)}{2}} \\ &= \frac{1}{2} \sum_{i \le j} \log \frac{\mathbf{B}_{c_1 c_2}}{1 - \mathbf{B}_{c_1 c_2}} \cdot (2\mathbf{W}_{ij} - 1) \cdot (2\mathbf{Y}_i^\top \mathbf{Y}_j - 1) + \mathrm{C} \end{aligned} \tag{82}$$

We decompose the sum into two parts, with

$$\sum_{r \in \mathcal{T}_{c_1 c_2}} \sum_{i \in \mathcal{V}_{c_1}, j \in \mathcal{V}_{c_2}} p_r \cdot (2\mathbf{W}_{ijr} - 1) \cdot (2\langle \mathbf{E}_r, \mathbf{Y}_i, \mathbf{Y}_j \rangle - 1), \tag{83}$$

where $p_r = \log \frac{\mathbf{B}_{c_1 c_2, r}}{1 - \mathbf{B}_{c_1 c_2, r}}$. If we introduce a reweighting parameter $\mathbf{E}$, we get the reformulation,in **??**. Now, in order to get this equivalence, we have the following constraint: 1) p_r needs to be positive; 2) $\mathbf{E}$, $\mathbf{X}_i$ and $\mathbf{X}_j$ has to be a unit vector.

However, due to the high complexity of heterogeneous graph structure data, while processing the datasets, people usually retain one type of "Target node" and record the label of that node type, and ignore the others. This brought issues when directly applying the definition 1. Thus, we further derive a relaxed condition for homophily through a random walk.

We consider the joint distribution of observing relation r and node label $c(v)$, sampling from a node u with $c(u)$. We use $v : c(v)$ to denote that sampling a node v with node label $c(v)$

$$P(v : c(v), r \mid u : c(u)) = P(v : c(v) \mid u : c(u)) \cdot P(r \mid u : c(u), v : c(v)). \tag{84}$$

As randomly walking on a graph is equivalent to bootstrap sampling of neighborhoods, when the sample size is large enough, we derive

$$\begin{aligned} P(r \mid u : c(u), v : c(v)) &= \mathbf{B}_{r,\tau} \\ P(v : c(v) \mid u : c(u)) &= \mathbf{D}_u^{-1/2} \mathbf{W}_{uv} \mathbf{D}_u^{-1/2} \\ &= \mathbf{D}_u^{-1} \mathbf{W}_{uv} \approx \sum_r \mathbf{B}_{r,\tau} \end{aligned} \tag{85}$$

Thus, the probability of reaching a node with the same label is

$$\begin{aligned} &\sum_{r_1, r_2} \mathbb{I}_{c(u) = c(w)} P(v : c(v), r_1 \mid u : c(u)) \times P(w : c(w), r_2 \mid v : c(v)) \\ &= \end{aligned} \tag{86}$$

Defnition 3 ***Relaxed Homophily on Heterogeneous graphs.*** *We consider taking a random walk starting from node v with type $\phi(v)$ and label $c(v)$, the heterogeneous graph is said to be homophily if there exists a representative node label c_v^* for each type of node $\phi(v)$, that the random walk starting from c_v^* has a large probability of reaching another same-typed node with same label c_v^*. With the formal mathematical statement in eq.* (61)

Derive the random-walk form of such a homophily.

$$\hat{p}_{r^*,\tau^*} - \sum_{\tau,r} p_{r,\tau} = \log \frac{\mathbf{B}_{r^*,\tau^*}}{1-\mathbf{B}_{r^*,\tau^*}} - \sum_{\tau,r} \frac{\mathbf{B}_{r,\tau}}{1-\mathbf{B}_{r,\tau}} > 0, \tag{87}$$

This homophily was empirically studied in Guo et al. (2023) but was not theoretically justified. Here we work as the first to connect the empirical study and the theory behind it. The homophily is also closely related to the meta-path-based method which considers the network schema, given that the random walk between same-typed nodes is always considered in a meta-path.

Another important aspect of estimating heterogeneous graphs from the node feature is how much the node feature $\mathbf{X}$ can represent the true label $\mathbf{Y}$ behind. This requires us to have some further assumptions on the transformation matrix $\phi(v)$. Intuitively, the node features for various node types should be unified into the same measurable space. In many datasets, such as ACM and IMDB, this is not difficult since the features are just the keywords of academic work and movie captions respectively. However, in other datasets, e.g. DBLP this is more difficult given that the features of venue and terms are not unified into the same space. For such datasets, a more reliable task would be link prediction, in which one gets the opportunity to learn the feature transformation a prior.

D.6 Motivating Graph Structure Learning: A Statistical Perspective

D.6.1 Graph Structure Inference and Stochastic Blocking Models

First, we consider the maximum a posterior (MAP) estimation of the adjacency matrix given the node labels,

$$\arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{Y}) = \arg\max_{\mathbf{W}} \log P(\mathbf{W} \mid \mathbf{Y}). \tag{88}$$

The log-posterior is then,

$$\begin{aligned}
&\log P(\mathbf{W} \mid \mathbf{Y}) = \log \Pi_{i\le j} P(\mathbf{W}_{ij} \mid \mathbf{Y}_i, \mathbf{Y}_j, \mathbf{B}) \\
&= \log \Pi_{i\le j} (\mathbf{Y}_i^\top \mathbf{B} \mathbf{Y}_j)^{\mathbf{W}_{ij}} (1-\mathbf{Y}_i^\top \mathbf{B} \mathbf{Y}_j)^{1-\mathbf{W}_{ij}} \\
&= \log \Pi_{i\le j} (\mathbf{Y}_i^\top \mathbf{B} \mathbf{Y}_j)^{\mathbf{W}_{ij}} [\mathbf{Y}_i^\top (\mathbb{I}_{c(i)c(j)} - \mathbf{B}) \mathbf{Y}_j]^{1-\mathbf{W}_{ij}} \\
&= \sum_{i,j} \mathbf{W}_{ij} \log(\mathbf{B}_{c(i)c(j)}) + (1-\mathbf{W}_{ij}) \log(1-\mathbf{B}_{c(i)c(j)}) \\
&= \sum_{i\le j} \mathbf{y}_i^\top \mathbf{B}' \mathbf{y}_j \cdot \mathbf{W}_{ij} + \text{Terms unrelated to } \mathbf{W}
\end{aligned} \tag{89}$$

where $\mathbf{B}'_{c(i)c(j)} = \log \frac{\mathbf{B}_{c(i)c(j)}}{(1-\mathbf{B}_{c(i)c(j)})} \coloneqq \beta_{ij}$ and $\mathbb{I}_{c(i)c(j)}$ is the identity matrix with only non-zero entry at $c(i), c(j)$.

With the above observation, we will show that the graph structure learning algorithm stated in eq. (9) is actually the extreme case when $\mathbf{B}' = \log[p/(1-p)]\mathbf{I}$, where one assumes homophily nodes are more likely to connect and heterophily nodes are connected arbitrarily, i.e. $P(\mathbf{W}_{ij} = 1 \mid \mathbf{y}_i = \mathbf{y}_j) > 1/2$ and $P(\mathbf{W}_{ij} = 1 \mid \mathbf{y}_i \neq \mathbf{y}_j) = 1/2$. To this end, we maximize eq. (89) and formalize an optimization problem as

follows,

$$\begin{aligned}
&\arg\max_{\mathbf{W}} \log P(\mathbf{W} \mid \mathbf{Y}) \\
&= \arg\max_{\mathbf{W}} \quad \log(\frac{p}{1-p}) \sum_{i \leq j} \mathbf{W}_{ij} \cdot \mathbf{y}_i^\top \mathbf{y}_j \\
&= \arg\min_{\mathbf{W}} \sum_{i \leq j} \mathbf{W}_{ij} \cdot \|\mathbf{y}_i - \mathbf{y}_j\|^2.
\end{aligned} \tag{90}$$

The last step comes with the fact that $\{\mathbf{Y}_i\}_{i=1}^N$ are one-hot vectors. The challenge in practice is that usually we do not have access to $\mathbf{Y}$. Instead we observe $\mathbf{X}$, which as stated in eq. (65), is a Gaussian variable transformed from $\mathbf{Y}$ as $\mathbf{Y}_i = \mathbf{X}_i \mathbf{U}^\top + \epsilon$. We can obtain $P(\mathbf{W} \mid \mathbf{X})$ by marginalizing out the noise,

$$\begin{aligned}
&\arg\max_{\mathbf{W}} P(\mathbf{W} \mid \mathbf{X}) \\
&= \arg\min_{\mathbf{W}} \mathbb{E}_{\epsilon_i, \epsilon_j} \sum_{i \leq j} \mathbf{W}_{ij} \cdot \|(\mathbf{X}_i - \mathbf{X}_j)\mathbf{U}^\top + \epsilon_i - \epsilon_j\|^2 \\
&= \arg\min_{\mathbf{W}} \sum_{i \leq j} \mathbf{W}_{ij} \cdot \|\mathbf{X}_i \mathbf{U}^\top - \mathbf{X}_j \mathbf{U}^\top\|^2 + 2\sigma_\epsilon^2 \|\mathbf{W}\| \\
&= \arg\min_{\mathbf{W}} \sum_{i \leq j} \mathbf{W}_{ij} \cdot \|\mathbf{X}_i - \mathbf{X}_j\|^2 + \Omega(\mathbf{W}).
\end{aligned} \tag{91}$$

$\Omega(\mathbf{W})$ works as the prior regularizer applied to the parameters to guarantee a meaningful learned graph structure (here it is a ℓ_1 norm, but there are various options in the literature Pu et al. (2021a); Kalofolias (2016b); Dong et al. (2015)). This formulation yields exactly the same optimization objective in eq. (9).

Remark: The derivation shows that the solution of MAP estimation of $\mathbf{W}$ on a homophily graph is exactly the optimization solution of minimizing smoothness. The convergence of such an optimization problem depends on the estimation from $\mathbf{X}$ to $\mathbf{Y}$, i.e. the matrix $\mathbf{U}$. We also show that the current graph structure learning algorithm serves as a special case when strong homophily is assumed on the graph. In what follows, we will introduce how such an algorithm can be extended to more general cases, i.e. heterogeneous graphs.

D.7 Motivating example

In this section, we give a motivating example to better understand how we could tackle the two limitations mentioned earlier. We start with an intuition behind academic networks, where nodes have types 'paper' and 'author' and signals are the attributes like 'topics', 'affiliation', and 'publication-date', etc. In such a context, papers on specific topics are inclined to cite papers in the same fields, indicating 'topics' as a pivotal attribute to identify the edges with the type 'Paper-Cite-Paper' ('PP'). Meanwhile, attributes like 'Affiliations' cannot help us determine the existence of 'PP'-typed edges, though they exist in the signal space and are beneficial for identifying other types of edges, e.g., 'Author-Write-Paper' ('PA'). Hence, when deciding whether to form an edge of type 'PP', the model should focus on the dimension of the node signals that represent highly relevant attributes like 'topics', but ignore the irrelevant ones like 'affiliation' or 'date'. To do so, a key concept, *dimension-wise smoothness* for the relation type r and the dimension k, is defined as

$$\mathcal{S}(\mathbf{X}_{:k}, \mathbb{W}_{::r}) = \sum_{\{v,u,r\} \in \mathcal{E}} w_{vur} \|\mathbf{x}_{v,k} - \mathbf{x}_{u,k}\|_{\mathrm{F}}^2, \tag{92}$$

where $\mathbb{W}_{::r}$ is the r-th slice of the adjacency tensor and $\mathbf{x}_{v,k}$ is the k-th element of the signal at node v.

We then consider two questions when designing HGSL within the framework of minimizing smoothness: Q1) How does dimension-wise smoothness help form edges? Q2) Can we distinguish various relation types based on dimension-wise smoothness? We will answer these questions by studying two heterogeneous graph datasets.

- **IMDB** Fu et al. (2020): a movie review dataset with node types including directors (D), actors (A), and movies (M) and with signals as 3066-D bag-of-words representation of keywords in the movie plot.
- **ACM** Lv et al. (2021): an academic dataset contains papers (P), authors (A), and subjects (S). Signals correspond to the 1902-D bag-of-words representation of the keywords in diverse research areas.

D.8 When labels are not observable

Now we can further factorize the likelihood of $P(\mathbf{X} \mid \mathbb{W}, \{\mathbf{B}_r\})$ by considering the graphical model in fig. 2,

$$\begin{aligned} &\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \log \int P(\mathbf{X} \mid \mathbf{Y}) P(\mathbf{Y} \mid \mathbb{W}, \{\mathbf{B}_r\}) d\mathbf{Y} \\ &- \Omega(\mathbb{W}) - \Omega(\{\mathbf{B}_r\}) \\ =&\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \log \mathbb{E}_{\mathbf{Y}\sim P(\cdot\mid\mathbb{W},\{\mathbf{B}_r\})} P(\mathbf{X} \mid \mathbf{Y}) \\ &- \Omega(\mathbb{W}) - \Omega(\{\mathbf{B}_r\}) \end{aligned} \tag{93}$$

In order to solve the above optimization, we instead optimize over the Evidence Lower Bound (ELBO) similarly in the expectation-maximization (EM) algorithm, where the training objective in the maximization step is reorganized as (the prior is omitted for now), The second step is problematic:

$$\begin{aligned} &\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \mathbb{E}_{\mathbf{Y}\sim P(\cdot\mid\mathbb{W},\{\mathbf{B}_r\})} \log P(\mathbf{X} \mid \mathbf{Y}) \\ =&\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \int d\mathbf{Y} P(\mathbf{Y} \mid \mathbb{W}, \{\mathbf{B}_r\}) \log P(\mathbf{X} \mid \mathbf{Y}) \end{aligned} \tag{94}$$

If we substitute $\mathbf{Y}$ with $\mathbf{U}_{\phi(v)}^{\top}(\mathbf{X} + \epsilon)$, the first term becomes

$$\begin{aligned} &\int_{\mathbf{Y}} \log P(\mathbf{X} \mid \mathbf{Y}) \\ &= \frac{1}{Z} \int_{\epsilon} \|\mathbf{X} - \mathbf{V}\mathbf{U}^T\mathbf{X} + \epsilon\|_{\mathrm{F}}^2, \end{aligned} \tag{95}$$

which is unrelated to the optimization problem. The training objective can then be reformalized as,

$$\arg\max_{\mathbb{W},\{\mathbf{B}_r\}} \int_{\mathbf{Y}} \log P(\mathbf{Y} \mid \mathbb{W}, \{\mathbf{B}_r\}) \tag{96}$$

To understand Q1, by exhaustive search, we identified the 200 most and least smooth dimensions $\mathcal{K}^{\mathrm{M}}(r)$ and $\mathcal{K}^{\mathrm{L}}(r)$, respectively, by ranking $\mathcal{S}(\mathbf{X}_{:k}, \mathbb{W}_{::r})$. For each of these two sets, we evaluate the pair-wise similarity (which is consistent with smoothness but normalized to [0,1]) of the signals, i.e., $\sum_{k\in\mathcal{K}^{\mathrm{L}}(r)} \mathbf{x}_{v,k} \cdot \mathbf{x}_{u,k}$, and $\sum_{k\in\mathcal{K}^{\mathrm{M}}(r)} \mathbf{x}_{v,k} \cdot \mathbf{x}_{u,k}$. For similarity higher than a given threshold, we establish an edge w_{vur} and compared the constructed graph with the ground truth one in an edge identification task. We compared the performance between the two generated graphs by the area under the curve (AUC) in table 1. We found that dimensions with higher smoothness are more helpful for edge identification task in HGSL.

To answer Q2, we assume that edges with distinct relation types are associated with smoothness evaluated in different signal dimensions, and the smoothest dimensions are the most representative ones. This motivates us to define *Smoothest-Dimension Overlapping Ratio (SDOR)*: for each pair of relations (r, r'), the SDOR is calculated by counting the overlapping dimensions in $\mathcal{K}^{\mathrm{M}}(r)$ and $\mathcal{K}^{\mathrm{M}}(r')$: $\mathrm{SDOR}(r, r') = \frac{|\mathcal{K}^{\mathrm{M}}(r) \cap \mathcal{K}^{\mathrm{M}}(r')|}{|\mathcal{K}^{\mathrm{M}}(r) \cup \mathcal{K}^{\mathrm{M}}(r')|}$. The SDOR reflects how different two relation types are in terms of exhibiting smoothness in signal dimensions and we report the pair-wise SDOR in table 1. The result suggests that for pairs of relation types with lower SDOR the signals will exhibit smoothness in different dimensions, which reveals the possibility of distinguishing them by comparing the smoothest dimensions found.

E Future Work

E.0.1 Link Prediction

Besides heterogeneous graph structure learning, other tasks related to heterogeneous graph, e.g. node classification and link prediction can also be formalized in our setting.

This section will discuss another setting of Graph Structure Learning, when a few edges are observed through the training stage. This can provide information to the model on metric learning for "the smoothness". And they are usually considered as link prediction (or knowledge graph completion) problems.

In this case, we assume that the model has access to a subset of edges. So it is possible to learn an approximated transformation matrix prior to learning the graph structure based on the smoothness assumption.

A common method for smoothness-based link prediction is proposed in Kalofolias (2016a). We generalize this into heterogeneous graph case, and obtain the edge-wise estimation as :

$$\hat{w}_{urv} = \exp\left(-\frac{\|(\mathbf{e}_r) \circ (\mathbf{x}_u - \mathbf{x}_v)\|_{\mathrm{F}}^2}{2\sigma^2}\right) \tag{97}$$

which uses a negative exponential to obtain the estimated weight matrix. And now the training objective is

$$\min_{\hat{\mathbb{W}}} \|\mathcal{M} \cdot (\mathbb{W} - \hat{\mathbb{W}})\|_F^2 + \alpha \langle \hat{\mathcal{L}}, (\mathbf{X}' \otimes \mathbf{E}' \otimes \mathbf{X}')_k$$

We decompose the learning objective into 2 parts. The first part is to use the observed entries to estimate the transformation matrix $\mathbf{M}_{\phi(v)}$ and $\mathbf{M}_{\psi(r)}$; and the second part is to optimize the graph with the smoothness assumption. For the first method, we can look at it edge-wise and it enables us to use a sub-sampling strategy to accelerate the training:

$$\min_{\mathbf{M}} \|\mathcal{M} \cdot (\mathbb{W} - \hat{\mathbb{W}})\|_F^2 = \min_{\hat{\mathbb{W}}} \sum \|w_{urv} - \hat{w}_{urv}\|_F^2$$

This gives us an approximated transformation matrix. And we can further use it for graph structure learning, or do link prediction for the unobserved part by

F Commands

$$\begin{aligned} \text{Learning: } \hat{\Theta} &= \operatorname{argmax}_{\Theta} \prod_{(\mathbf{x}_k, \boldsymbol{y}_k) \in D_{\mathrm{tr}}} P\left(\boldsymbol{y}_k \mid \boldsymbol{x}_k; \Theta\right) \\ &\text{where, } \boldsymbol{y}_k \sim \mathcal{N}\left(h_\Theta(\mathbf{x}_k), \sigma^2\right) \end{aligned} \tag{98}$$

$$\hat{\Theta} = \operatorname{argmax}_{\Theta} \prod_{(\mathbf{x}_k) \sim p_{\mathrm{data}}} q_\Theta\left(\boldsymbol{x}_k\right) \tag{99}$$

$$\hat{\Theta} = \operatorname{argmax}_{\Theta} \prod_{(\mathbf{x}_k) \sim p_{\mathrm{data}}} P\left(\boldsymbol{x}_k \mid \Theta\right) \tag{100}$$

$$\begin{aligned} \hat{\Theta} &= \arg\max_{\Theta} P\left(\Theta \mid \boldsymbol{y}, \mathbf{x}\right) \\ &= \arg\max_{\Theta} \log \underbrace{P\left(\boldsymbol{y} \mid \mathbf{x}; \Theta\right)}_{\text{likelihood}} + \log \underbrace{P(\Theta \mid \mathbf{x})}_{\text{prior}} \end{aligned} \tag{101}$$

$$
\begin{aligned}
P(\mathbf{X}, \mathbf{Y} \mid \Theta) &= P(\mathbf{Y}; \Sigma) P(\mathbf{X} \mid \mathbf{Y}; \mathbf{V}) \\
&\propto \prod_{v \in \mathcal{V}} \psi_v \left(\mathbf{y}_v\right) \prod_{u,v \in \mathcal{E}} \psi_{uv} \left(\mathbf{y}_v, \mathbf{y}_u; \Sigma\right) \prod_{i=1}^{N} P\left(\mathbf{x}_v \mid \mathbf{y}_v; \mathbf{V}\right)
\end{aligned} \tag{102}
$$

$$
P\left(\mathbf{x}_v \mid \mathbf{y}_v; \mathbf{V}\right) \sim \mathcal{N}\left(\mathbf{V}\mathbf{y}_v, \sigma_{\mathbf{x}}^2\right) \tag{103}
$$