Learning Shared Subgraphs in Ising Model Pairs

Burak Varıcı[1]
Rensselaer Polytechnic Institute

Saurabh Sihag[1]
Rensselaer Polytechnic Institute

Ali Tajer
Rensselaer Polytechnic Institute

Abstract

Probabilistic graphical models (PGMs) are effective for capturing the statistical dependencies in stochastic databases. In many domains (e.g., working with multimodal data), one faces multiple information layers that can be modeled by structurally similar PGMs. While learning the structures of PGMs in isolation is well-investigated, the algorithmic design and performance limits of learning from multiple coupled PGMs are not well-investigated. This paper considers learning the structural similarities shared by a pair of Ising PGMs. The objective is to learn only the shared structure with no regard for the structures exclusive to either of the graphs. This is significantly different from the existing approaches that focus on learning the entire structures of the graphs. This paper proposes an algorithm for learning the shared structure, evaluates its performance empirically and analytically, and compares the performance with that of the existing approaches.

1 INTRODUCTION

Probabilistic graphical models (PGMs) are commonly used for capturing the conditional interdependence in probabilistic databases or random fields (Lauritzen, 1996; Pearl, 2009). Each vertex in a PGM represents a random variable (RV), and the edges encode the interdependence among the RVs. The complete structure of a PGM is captured by the joint probability measure of all the random variables involved. PGMs offer a rich context for designing effective and tractable algorithms for various inferential and decision-making objectives in a broad range of technological, social, or biological networks. Some examples include computer vision (Won and Derin, 1992), genetics (Chen et al., 2013; Fang et al., 2016; Dobra et al., 2004), and social networks (Jacob et al., 2014).

There are various domains in which we face multiple layers of information networks. Specifically, due to the proliferation of sensing technologies, there is a growing trend of facing multimodal databases. Graphically modeling such databases, as a result, involves multiple PGMs, each corresponding to one data mode. For instance, consider localizing anomalous events in networks. When a network's behavior undergoes abnormal changes, localizing the changes requires determining the discrepancies between the pre- and post-change network models. In another example, consider diffusion tensor imaging (DTI) and functional magnetic resonance imaging (fMRI) for brain imaging. DTI and fMRI images of a brain represent different structures of the underlying brain network (Honey et al., 2007), and their conformity of the two images can be leveraged to assess a brain's cognitive health (Buckner and DiNicola, 2019). Finding the maximum common subgraph in graph modeling of molecular structures in biology is another application important for drug discovery (Okamoto, 2020).

In this paper, we focus on learning the structural similarities between a pair of PGMs. We emphasize that our focus is on learning *only the shared* subgraphs without any interest in the structures exclusive to the individual graphs. A naive approach to this problem is learning each of the graphs independently and then searching for similarities. This approach is highly inefficient since it learns the graphs *completely*, resulting in unnecessarily learning the exclusive structures of the graphs too. This compromises the efficiency (e.g., sample complexity), especially in large-scale graphical models that have relatively small shared structures.

Learning the structure of a single graph, while being NP-hard (Chickering, 1996) in general, becomes tractable under various restrictions on graph struc-

[1]Equal contribution.

Proceedings of the 24[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

tures (Yuan and Lin, 2007; Rothman et al., 2008; Ravikumar et al., 2010; Banerjee et al., 2008; Anandkumar et al., 2010; Klivans and Meka, 2017). Relevant studies to the scope of this paper are the structure learning algorithms for tree-structured Ising models in (Bresler et al., 2020); for loosely connected Ising models with correlation decay in (Wu et al., 2013); and for degree-bounded Ising models in (Anandkumar et al., 2010; Bresler, 2015; Vuffray et al., 2016; Klivans and Meka, 2017). A greedy learning approach for structure learning is also proposed in (Bresler, 2015). A convex optimization framework for structure learning is formalized in (Vuffray et al., 2016). An online learning-based algorithm driven by the principles of prediction with expert advice is proposed in (Klivans and Meka, 2017).

While there exists rich literature on structure learning of a single graph, achievable and fundamental limits of structure learning from a group of related graphs are not as well-investigated. These existing studies focus on joint learning of all graphs in their *entirety*. The existing studies include strategies for joint learning of multiple graphical models that leverage the side information about the similarities among different graphs. In (Chen et al., 2013), an empirical Bayes method is studied to identify interactions that are unique to each class of cancers and that are common across all classes. In (Fang et al., 2016; Guo et al., 2011; Danaher et al., 2014; Mohan et al., 2014; Yang et al., 2015), joint inference of Gaussian graphical models is studied using graphical Lasso-based algorithms. Joint estimation of the graph structures based on discrete data is studied in (Guo et al., 2015). Information-theoretic bounds on the sample complexity of joint learning of graphical models with their similarity as side information are studied in (Sihag and Tajer, 2019a,b).

In a relevant problem, there exist studies on directly estimating the difference between graphs. (Zhao et al., 2014; Liu et al., 2014) study this problem for undirected graphs, where (Wang et al., 2019; Ghoshal and Honorio, 2019) consider causal linear structural equation models. This literature focuses on sparse discrepancies between the graph pairs, i.e., the pair are overwhelmingly similar.

In contrast to the studies mentioned above on learning the entire structures of structurally similar graphs, and complementary to learning their differences, in this paper, our objective is to learn only the *common* edge structure between two graphs using their data samples. Our focus is on Ising models with similarly labeled vertices and freely distinct structures. The subgraphs' size and its location are unknown, and we propose an algorithm for determining the size, identifying the location, and learning the structure of a subgraph shared by two distinct Ising models. In the face of lack of any information about the shared structure's size or location, a direct approach to learning the shared structure involves learning both graphs entirely and then finding their alignment. This approach will inevitably learn the unshared parts, which translates into severe inefficiency when our objective is learning only the shared structure. To circumvent this, we devise an adaptive algorithm that progressively prunes the vertices deemed highly unlikely to belong to the shared subgraph. We evaluate the effectiveness of the proposed algorithm by experiments on synthesized datasets and compare the performance of joint shared structure learning with that achieved by the direct approach of learning the two graphs independently to determine their shared structure in different regimes of interest such as correlation decay (Anandkumar et al., 2010) and bounded model width (Klivans and Meka, 2017; Wu et al., 2019).

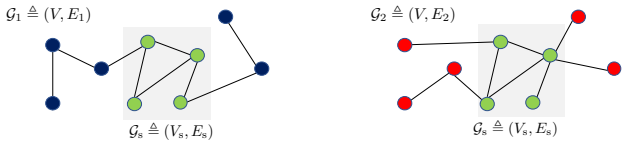


Figure 1: Green vertices and the edges between them represent the shared subgraph $\mathcal{G}_s$.

2 GRAPHICAL MODELS

Consider two distinct undirected graphs $\mathcal{G}_1 \triangleq (V, E_1)$ and $\mathcal{G}_2 \triangleq (V, E_2)$ over a set of vertices $V \triangleq \{1, \dots, p\}$. The graphs are formed by two distinct collections of undirected edges denoted by $E_1 \subseteq V \times V$ and $E_2 \subseteq V \times V$. When there exists an edge between vertices $u, v \in V$ in graph $\mathcal{G}_i$, we denote it by $(u, v) \in E_i$. The set of neighbors of vertex u in graph $\mathcal{G}_i$ is denoted by

$$\mathcal{N}_i(u) \triangleq \{w \in V : (u, w) \in E_i\} \ . \tag{1}$$

The graphs are degree-bounded, with their maximum degrees being upper bounded by $d \in \mathbb{N}$. We say that the edge (u, v) is shared by both graphs if $(u, v) \in E_i$ for $i \in \{1, 2\}$. The *unknown* set of edges that are shared by both graphs is denoted by $E_s = E_1 \cap E_2$. Accordingly, we define V_s as the set of vertices that the edges in E_s cover, and denote its size by $q \triangleq |V_s|$. Finally, the graph formed by the vertices in V_s and the edges in E_s is denoted by $\mathcal{G}_s \triangleq (V_s, E_s)$.

Ising model: We assume Ising models for both graphs, and define $X_i^u \in \mathcal{X} \triangleq \{-1, 1\}$ as the random variable associated with the vertex $u \in V$ in graph $\mathcal{G}_i$, and define the random vector $\mathbf{X}_i \triangleq [X_i^1, \dots, X_i^p]$ as the collection of the random variables in $\mathcal{G}_i$. We define

$\lambda \in \mathbb{R}^+$ to capture the dependence between the random variables associated with the vertices in a graph. The joint probability mass function (pmf) of random vectors $\mathbf{X}_1$ and $\mathbf{X}_2$, denoted by $f(\mathbf{X}_1, \mathbf{X}_2)$, is given by

$$f(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{Z_{12}} \exp\left(\sum_{(u,v)\in E_{\mathrm{s}}} \lambda(X_1^u X_1^v + X_2^u X_2^v) + \sum_{(u,v)\in \tilde{E}_1} \lambda X_1^u X_1^v + \sum_{(u,v)\in \tilde{E}_2} \lambda X_2^u X_2^v \right), \tag{2}$$

where Z_{12} is the partition function that ensures $f(\mathbf{X}_1, \mathbf{X}_2)$ is a valid pmf, and we have defined $\tilde{E}_1 \triangleq E_1 \backslash E_{\mathrm{s}}$ and $\tilde{E}_2 \triangleq E_2 \backslash E_{\mathrm{s}}$. The class of graphs associated with $\mathcal{G}_{\mathrm{s}}$ is given by $\mathcal{I}_p^{\mathrm{s}}$ and it is formally defined as follows.

Definition 1 ($\mathcal{I}_p^{\mathrm{s}}$ class)**.** *Define $\mathcal{I}_p$ as the class of Ising models with p vertices, and define $\mathcal{I}_p^{\mathrm{s}} \subseteq \mathcal{I}_p$ as the class of all valid subgraphs $\mathcal{G}_{\mathrm{s}}$ that can be shared by two distinct Ising models with p vertices. A valid edge structure refers to any instance of a collection of edges that form E_{s}.*

Furthermore, we define $\mathcal{I}_p^{\mathrm{s}}(\mathcal{G}_{\mathrm{s}}) \subseteq \mathcal{I}_p \times \mathcal{I}_p$ as the class of all possible pairs of Ising models whose shared structure is given by $\mathcal{G}_{\mathrm{s}}$. We also denote the set of random variables associated with V_{s} in $\mathcal{G}_i$ by $\mathbf{X}_i^{\mathrm{s}}$ and those with $V \backslash V_{\mathrm{s}}$ by $\mathbf{X}_i^{\mathrm{c}}$. The marginal joint pmf of the random variables $\mathbf{X}_i^{\mathrm{s}}$ is given by

$$\begin{aligned}\tilde{f}(\mathbf{X}_1^{\mathrm{s}}, \mathbf{X}_2^{\mathrm{s}}) &\triangleq \frac{1}{|\mathcal{I}_p^{\mathrm{s}}(\mathcal{G}_{\mathrm{s}})|} \exp\left(\sum_{(u,v)\in E_{\mathrm{s}}} \lambda(X_1^u X_1^v + X_2^u X_2^v) \right) \\ &\times \left(\sum_{\mathbf{X}_1^{\mathrm{c}}, \mathbf{X}_2^{\mathrm{c}}} \sum_{(\tilde{E}_1, \tilde{E}_2)\in \mathcal{I}_p^{\mathrm{s}}(\mathcal{G}_{\mathrm{s}})} \frac{1}{Z_{\tilde{E}_1, \tilde{E}_2}} \right. \\ &\left. \times \exp\left(\sum_{(u,v)\in \tilde{E}_1} \lambda X_1^u X_1^v + \sum_{(u,v)\in \tilde{E}_2} \lambda X_2^u X_2^v \right) \right),\end{aligned} \tag{3}$$

where $Z_{\tilde{E}_1, \tilde{E}_2}$ is a partition function associated with pmf of the pair of Ising models with edge structures $\tilde{E}_1$ and $\tilde{E}_2$ unique to $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Clearly, finding a closed-form for $\tilde{f}(\mathbf{X}_1^{\mathrm{s}}, \mathbf{X}_2^{\mathrm{s}})$ is intractable in general (except for specific cases, such as connected $\mathcal{G}_{\mathrm{s}}$ in tree-structured graphs or isolated graph $\mathcal{G}_{\mathrm{s}}$) and performing marginal inference on Ising models is an open problem. We next provide the objectives and performance metrics for learning the structure of $\mathcal{G}_{\mathrm{s}}$.

3 OBJECTIVES AND METRICS

Graph decoding. In this section, we formalize the criterion for learning the structure of the shared subgraph and the attendant performance metrics. Our objective is to learn only the structure of $\mathcal{G}_{\mathrm{s}}$ based on a collection of n samples generated by each graph. The collection of n independent samples from $\mathcal{G}_i$ is denoted by $\mathbf{x}_i^n \in \mathcal{X}^{n\times p}$. We formally define the graph decoder to represent the algorithmic framework that uses samples from both graphs and estimates the shared structure.

Definition 2 (Graph Decoder)**.** *We define a graph decoder ψ_{s} as a function that maps the data samples to a subgraph in $\mathcal{I}_p^{\mathrm{s}}$, i.e.,*

$$\psi_{\mathrm{s}} : \mathcal{X}^{n\times p} \times \mathcal{X}^{n\times p} \to \mathcal{I}_p^{\mathrm{s}} \,. \tag{4}$$

Our objective is to *perfectly* learn the structure of the shared subgraph $\mathcal{G}_{\mathrm{s}}$. Hence, an error event occurs when $\psi(\mathbf{x}_1^n, \mathbf{x}_2^n) \neq E_{\mathrm{s}}$. In this paper, we characterize the graph decoder ψ by providing an adaptive algorithmic framework and empirically evaluating its performance. In order to specify relevant performance metrics for ψ in terms of number of samples, we first provide a brief overview of the algorithm with its details specified in Section 4.

Algorithm Overview. We design an adaptive algorithm based on the following premise. Given that the size, the location, and structure of the subgraph $\mathcal{G}_{\mathrm{s}} = (V_{\mathrm{s}}, E_{\mathrm{s}})$ are all unknown, an attempt for *directly* identifying it will inevitably require learning significant fractions of the structures of both graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, well beyond only their shared graphs, thus, potentially penalizing the efficiency (e.g., sample complexity). Motivated by this, we devise an algorithm that is focused on learning only $\mathcal{G}_{\mathrm{s}}$. We initialize our algorithm by considering all vertices in V as candidates for being in the shared subgraph. The set of vertices of interest at the k-th iteration of the algorithm is denoted by $\hat{V}_{\mathrm{s}}(k)$, where $\hat{V}_{\mathrm{s}}(0) = V$. At every iteration, we evaluate the empirical pairwise correlation between the data samples of the pair of vertices that could potentially be in $\mathcal{G}_{\mathrm{s}}$. If all pairwise empirical correlations for a vertex fall below a certain threshold, that vertex ceases to be of interest, and we stop collecting new data samples from it. In parallel to pairwise correlation-based pruning of the vertices of interest in the graphs, we also run an online structure learning algorithm with similar principles as adopted in (Klivans and Meka, 2017), but with appropriate modifications to facilitate joint learning using samples from two graphs.

Performance metrics. First, we collect n_{L} graph samples $\{(\mathbf{x}_1^k, \mathbf{x}_2^k) : k \in \{1, \ldots, n_{\mathrm{L}}\}\}$ one-at-a-time and

evaluate $\hat{V}_{\rm s}(k) \subseteq V$ and $(E_1^k \cap E_2^k)$ to form estimates of $\mathcal{G}_s = (V_{\rm s}, E_{\rm s})$. We denote the probability of error in recovering shared structures from a class of graphs $\mathcal{I}_p^{\rm s}$ by

$$\mathsf{P}_{\rm L}(\mathcal{I}_p^{\rm s}) \triangleq \max_{\mathcal{G}_1,\mathcal{G}_2 \in \mathcal{I}_p} \mathbb{P}(|E_{\rm s} \Delta \hat{E}_{\rm s}| \neq 0) \ , \tag{5}$$

where Δ denotes the difference between two edge sets, i.e., $E_1 \Delta E_2 = (E_1 \setminus E_2) \cup (E_2 \setminus E_1)$. For a structure learning process, we evaluate the performance of our algorithm for the exact recovery of shared subgraph in two ways:

- The total number of algorithm iterations, denoted by $n_{\rm T}$.
- The total number of measurements collected over $n_{\rm T}$ iterations, i.e.,

$$\mathsf{N}(n_{\rm T}) \triangleq \sum_{k=1}^{n_{\rm T}} |\hat{V}_{\rm s}(k)| \ . \tag{6}$$

If we collect samples from all vertices in each iteration, we have $\mathsf{N}(n_{\rm T}) = pn_{\rm T}$. However, since the number of vertices in $\hat{V}_{\rm s}(k)$ potentially decreases over the iterations, we expect $\mathsf{N}(n_{\rm T})$ to be considerably smaller than $pn_{\rm T}$ for graph pairs with $V_{\rm s} \subset V$.

4 STRUCTURE LEARNING

In this section, we provide Algorithm 1 for shared structure learning, provide an insight into the logic of the included steps, and characterize its sample complexity under certain assumptions. Our algorithm mainly relies on two ideas: pruning the set of vertices of interest and jointly learning the edge structure spanned by vertices of interest in the two graphs.

4.1 Similarity-based Pruning

We focus on localizing the vertices in $V_{\rm s}$ during the structure learning, and hence, pruning the vertices that are highly likely not to be a part of the shared subgraph. These decisions are made in an online manner, and their objective is to form *coarse* decisions about the vertices that are promising candidates for being members of $V_{\rm s}$, and therefore, are retained for further scrutiny. In this part of the algorithm, as more data is collected, the sampling resources progressively shift to the vertices that are deemed to have a higher chance of belonging to the shared subgraph. This facilitates a significant reduction in the sampling complexity, as observed in the experiments.

For k samples from vertices u and v, we define the empirical pairwise mean of X_i^u and X_i^v as follows:

$$\bar{\mathbb{E}}_k(X_i^u X_i^v) \triangleq \frac{1}{k} \sum_{\ell=1}^{k} x_i^u(\ell) x_i^v(\ell) \ , \tag{7}$$

where $x_i^u(\ell)$ represents the ℓ-th data sample from vertex u in $\mathcal{G}_i$. We note that the pairwise correlation in (7) is a sufficient statistic for capturing the structure of Ising models in certain correlation-decay regimes (Bento and Montanari, 2011). Using the empirical pairwise correlations, we design the thresholding rule for controlling false negative rates. Specifically, at iteration k, we include vertices u, v in $\hat{V}_{\rm s}(k)$ if they meet the criterion:

$$\min_{i \in \{1,2\}} \bar{\mathbb{E}}_k[X_i^u X_i^v] \ > \ h(\lambda, \alpha, p, k) \ , \tag{8}$$

where we have defined

$$h(\lambda, \alpha, p, k) \triangleq \tanh \lambda - \sqrt{\frac{\alpha \log p}{2k}} \ , \tag{9}$$

and $\alpha > 2$ controls the rate of false negatives. The rationale for the above choice of $h(\lambda, \alpha, p, k)$ will become clear with the following lemma.

Lemma 1. *The pruning rule specified in* (8) *with k samples and setting $h(\lambda, \alpha, p, k)$ according to* (9) *ensures that $\mathbb{P}(V_{\rm s} \subseteq \hat{V}_{\rm s}(k)) \geq 1 - 2p^{2-\alpha}$.*

Proof. Note that for a vertex pair (u, v) that is connected via an edge in an Ising model, we have $\mathbb{E}[X^u X^v] \geq \tanh \lambda$ (Daskalakis et al., 2019). Using the Hoeffding's inequality (Hoeffding, 1994) to establish the the concentration bound around the true pairwise means, for any $\epsilon > 0$, we have

$$\mathbb{P}[|\bar{\mathbb{E}}_k[X_i^u X_i^v] - \mathbb{E}[X_i^u X_i^v]| \leq \epsilon] \geq 1 - \frac{2}{p^\alpha} \ , \tag{10}$$

for $k = \frac{\alpha \log p}{2\epsilon^2}$ number of samples. This ensures that by plugging $\epsilon = \sqrt{\alpha \log p / 2k}$ in (10) and using (9) for an edge (u, v), the relationship

$$\bar{\mathbb{E}}_k[X_i^u X_i^v] \geq \mathbb{E}[X_i^u X_i^v] - \sqrt{\frac{\alpha \log p}{2k}} \geq h(\lambda, \alpha, p, k) \ , \tag{11}$$

holds with a probability at least $1 - 2p^{-\alpha}$. Taking a union bound over all possible pairs in both graphs, (11) holds for all such pairs with probability at least $(1 - 2p^{2-\alpha})$. This implies that this decision rule (8) guarantees that $V_{\rm s} \subseteq \hat{V}_{\rm s}(k)$ with a probability not smaller than $1 - 2p^{2-\alpha}$. □

Algorithm 1 Learning Shared subgraph $\mathcal{G}_{\rm s}$

Input α, β, $\hat{V}_{\rm s}(0) = V$, $\hat{E}_{\rm s}(0) = \emptyset$, $n_{\rm L} = n_{\rm T} + n_{\rm M}$ pairs of data samples
Initialize $\kappa_i^{uv}(1) = 1/(|V|-1)$, $\tilde{\kappa}_i^{uv}(1) = 1/(|V|-1)$ and $w_i^{uv}(1) = 0$ for all $u \neq v$ and $i \in \{1, 2\}$
for a new pair of data sample $k \in \{1, \ldots, n_{\rm T}\}$ **do**
  for each pair $u, v \in \hat{V}_{\rm s}(k)$, $u \neq v$ **do**
    Calculate $\bar{\mathbb{E}}_k[X_1^u X_1^v]$ and $\bar{\mathbb{E}}_k[X_2^u X_2^v]$
    if (8) is satisfied **then**
      Add (u, v) to $\hat{E}_{\rm s}(k)$
    end if
    Update the weights $\tilde{\kappa}_i^{uv}(k+1) = \tilde{\kappa}_i^{uv}(k) \exp(\beta/2)$
  end for
  Pruning step: Construct $\hat{V}_{\rm s}(k) = \{u | \exists v, (u, v) \in \hat{E}_{\rm s}(k)\}$
  Get samples from $\hat{V}_{\rm s}(k)$ vertices on both graphs and compute the losses $\ell_i^{uv}(k)$
  Weight update step:
  for each pair $(u, v) \in \hat{V}_{\rm s}(k)$, $u \neq v$ **do**
    Update the weights jointly $\kappa_i^{uv}(k+1)$ according to (14)
  end for
  for each pair $(u, v) \notin \hat{V}_{\rm s}(k)$, $u \neq v$ **do**
    Update the weights $\kappa_i^{uv}(k+1) = \kappa_i^{uv}(k) \exp(\beta/2)$
  end for
  for each pair $u \neq v$ **do**
    Compute $w_i^{uv}(k+1)$ via normalizing $\kappa_i^{uv}(k+1)$
  end for
  Compute estimates E_1^k and E_2^k such that for every pair $u \neq v$, $(u, v) \in E_i^k$ if $w_i^{uv} \geq \lambda/2$
  Compute empirical risks ϵ_i^k using $n_{\rm M}$ samples
end for
Compute $\mathrm{m}_1 = \operatorname{argmin}_k \epsilon_1^k$ and $\mathrm{m}_2 = \operatorname{argmin}_k \epsilon_2^k$
return $\hat{E}_{\rm s} = E_1^{\mathrm{m}_1} \cap E_2^{\mathrm{m}_2}$

4.2 Prediction-guided Structure Learning

Given the coarse estimate $\hat{V}_{\rm s}(k)$ at the k-th iteration, we then focus on learning the internal graph structures of vertices in $\hat{V}_{\rm s}(k)$. For this purpose, we provide a prediction-guided learning algorithm where we leverage the data samples only from the vertices in $\hat{V}_{\rm s}(k)$. It is noteworthy that since the estimate $\hat{V}_{\rm s}(k)$ may be imperfect and the internal structure of $\hat{V}_{\rm s}(k)$ may have additional edges, the subgraphs spanned by these vertices in the two graphs are not necessarily the same.

To formalize the learning process, corresponding to each random variable $X_i^u \in \{-1, 1\}$, we define the Bernoulli random variable $B_i^u \triangleq \frac{1}{2}(1 - X_i^u)$ and instead of analyzing X_i^u we equivalently analyze B_i^u. Accordingly, we denote the the k-th realization of B_i^u by $b_i^u(k) = \frac{1}{2}(1 - x_i^u(k))$. The prediction-guided algorithm consists of two steps. Step 1 collects $n_{\rm T} < n_{\rm L}$ samples to form prediction-guided estimates for $E_{\rm s}$ at each iteration k. We note that since the estimate $\hat{V}_{\rm s}(k)$ becomes more accurate with an increasing number of samples, we need measurements from a gradually decreasing number of vertices. Once the predictions are formed, we use the remaining $n_{\rm M} \triangleq n_{\rm L} - n_{\rm T}$ samples to form a final estimate for $\mathcal{G}_{\rm s}$ in Step 2.

Step 1: Predicting $E_{\rm s}$. We start with the premise that any pair of vertices in $\hat{V}_{\rm s}(k)$ can be potential neighbors. Each vertex can act as an *expert* and predicts the value of its neighbors. In the k-th iteration, at vertex u in $\mathcal{G}_i$, we form a prediction for $b_i^u(k)$ by aggregating the predictions provided by other vertices in the graph for this vertex according to

$$\hat{b}_i^u(k) = \sigma\Big(\sum_{v \neq u} \kappa_i^{uv}(k) x_i^v(k) \Big), \quad \forall k \in \{1, \ldots, n_{\rm T}\}, \tag{12}$$

where $\{\kappa_i^{uv}\}$ are the weights to be selected properly and σ is the standard sigmoid function.

Loss function. To quantify the quality of the predictions, for every pair $u, v \in \hat{V}_{\rm s}$ we evaluate the pairwise loss function

$$\ell_i^{uv}(k) \triangleq \frac{1}{2}\Big(1 + [\hat{b}_i^u(k) - b_i^u(k)] x_i^v(k)\Big) . \tag{13}$$

Predictor Update. Note that $\hat{V}_{\rm s}(k)$ is formed with the decision rule (8). Hence, for all pairs $u, v \in \hat{V}_{\rm s}(k)$, we allow the transfer of loss functions between the graphs for updating the multiplicative weights

$$\kappa_i^{uv}(k+1) = \kappa_i^{uv}(k) \cdot \exp\left(\frac{\beta}{2} [\ell_1^{uv}(k) + \ell_2^{uv}(k)] \right) , \tag{14}$$

where β is an appropriately set hyperparameter, for instance $\beta = \log\Big(1/(1 + \sqrt{(\log p / n_{\rm T})})\Big)$ [1]. Otherwise, if $u, v \notin \hat{V}_{\rm s}(k)$, $\kappa_i^{uv}(k)$ is updated according to:

$$\kappa_i^{uv}(k+1) = \kappa_i^{uv}(k) \cdot \exp\left(\beta/2\right) . \tag{15}$$

Before continuing to the next step, we comment on the multiplicative updates in (14) which facilitate joint learning in our algorithm. We note that, independently learning each graph corresponds to the algorithm in (Klivans and Meka, 2017) that uses the following update for any pair $u, v \in V$:

$$\kappa_i^{uv}(k+1) = \kappa_i^{uv}(k) \cdot \exp\left(\beta \ell_i^{uv}(k)\right) . \tag{16}$$

[1] This choice allows us to leverage the standard Hedge algorithm regret in proof of Theorem 1.

If (u, v) lies in the shared subgraph $\mathcal{G}_{\mathrm{s}}$, then (14) corresponds to processing two graph samples at one step rather than processing 1 graph sample independently as in (16).

We illustrate the improvement of performance by using the mean of losses in the graphs for $u, v \in V_{\mathrm{s}}$ in joint structure learning over the algorithm for learning a single Ising model in (Klivans and Meka, 2017) in Section 5, and provide additional empirical results in Appendix D.

Normalizing weights κ_i^{uv}. Note that weight updates in (14) and (15) do not ensure that the maximum neighborhood weight of any vertex is bounded by λd (since d is the maximum degree). Therefore, we introduce normalized weights:

$$w_i^{uv}(k+1) = \frac{\lambda d \kappa_i^{uv}(k+1)}{\sum\limits_{x \neq u} \left(\kappa_i^{ux}(k+1) + \tilde{\kappa}_i^{ux}(k+1)\right)}, \quad (17)$$

where $\tilde{\kappa}_i^{uv}$ are pseudo-weights to handle vertices with a degree less than d. The theoretical justification for these pseudo-weights is discussed in Appendix A.

Predictions for E_{s}. Using the coefficients $\{w_i^{uv}(k) : k \in \{1, \ldots, n_{\mathrm{T}}\}\}$ we perform thresholding tests to estimate the internal structure of $\hat{V}_{\mathrm{s}}(k)$. Specifically, we form two estimates, one corresponding to each graph:

$$E_i^k \triangleq \left\{(u, v) \ : \ w_i^{uv}(k) \geq \frac{\lambda}{2}\right\}, \quad \forall i \in \{1, 2\}, \quad (18)$$

for $k \in \{1, \ldots, n_{\mathrm{T}}\}$. The processes described above continue sequentially until all the n_{T} samples are exhausted.

Step 2: Estimating E_{s}**.** Finally, to determine the correct subgraph $\mathcal{G}_{\mathrm{s}}$, we collect additional n_{M} samples for vertices in $\hat{V}_{\mathrm{s}}(n_{\mathrm{T}})$ and based on these, we assign a risk metric to each predicted set E_i^k according to:

$$\epsilon_i^k = \frac{1}{n_{\mathrm{M}}} \sum_{k=n_{\mathrm{T}}+1}^{n_{\mathrm{L}}} \sum_{u \in \hat{V}_{\mathrm{s}}(k)} \left[\sigma\left(\sum_{v \in E_i^k} -2 w_i^{uv}(k) x_i^v(k)\right) - b_i^u(k)\right]^2. \quad (19)$$

We select the predictions with the lowest empirical risks. By setting $m_i \triangleq \arg\min_k \epsilon_i^k$ for $i \in \{1, 2\}$, we form the final estimate as $\hat{E}_{\mathrm{s}} = E_1^m \cap E_2^m$.

4.3 Sample Complexity

We remark that in specific scenarios such as when $\mathcal{G}_{\mathrm{s}}$ is an isolated subgraph or in the asymptote of $\mathcal{G}_{\mathrm{s}} = \mathcal{G}_1 = \mathcal{G}_2$, the pmf $\tilde{f}$ becomes tractable and has closed-forms that conform to distributions for an Ising model, which facilitate theoretically characterizing the performance of the proposed algorithm. The sample complexity of the algorithm in a restricted setting is provided in the following theorem.

Theorem 1 (Learning the shared subgraph)**.** *When the graph $\mathcal{G}_{\mathrm{s}}$ is isolated and marginal distribution of $\mathcal{G}_{\mathrm{s}}$ admits Ising model distribution, the sample complexity of Algorithm 1 for ensuring $\mathsf{P}(\mathcal{I}_p^{\mathrm{s}}) \leq (1 - \frac{2}{\rho})(1 - 2p^{2-\alpha})$ is*

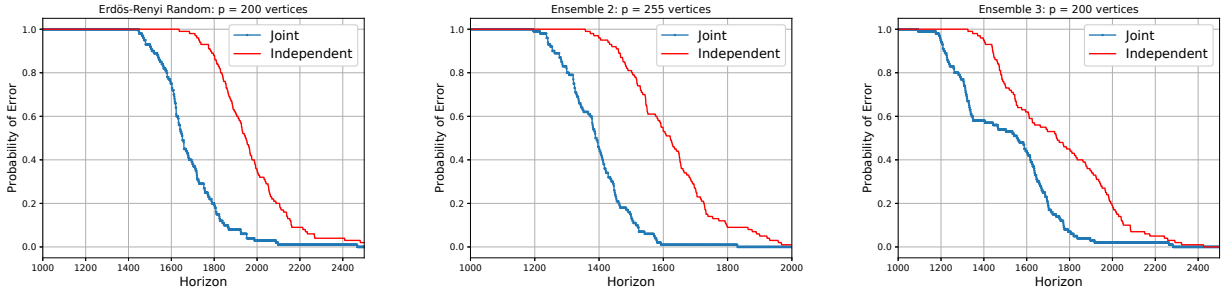
$$O\left(\frac{1}{\lambda^2} \exp(\lambda d) \log \frac{\rho q}{\lambda}\right) + O\left(\frac{1}{\lambda^2} \log p\right). \quad (20)$$

We remark that the condition on the marginal distribution of $\mathcal{G}_{\mathrm{s}}$ is too stringent in practical settings. Therefore, to emphasize upon the algorithm's applicability in wider settings, the rest of the paper focuses on empirical comparisons of Algorithm 1 with the existing structure learning algorithms in different regimes, such as correlation decay and bounded model width.

Remark 1. *The assumption that λ is uniform and known is made for clarity in presentation. It can be readily generalized to assume that (i) the edge weights are non-uniform, (ii) they are not fully known, and are only assumed to belong to a known range, i.e., $\lambda_{ij} \in \lambda \in [\lambda_{\min}, \lambda_{\max}]$. All analytical guarantees will still hold valid with proper adjustments. In Lemma 1, we replace λ with $\lambda_{\min}$ and in Theorem 1, we replace $\exp(\lambda d)$ with $\exp(\lambda_{\max} d)$ and other λ terms with $\lambda_{\min}$.*

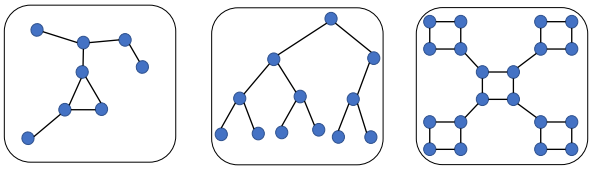
5 EXPERIMENTS

In this section, we evaluate the performance of our algorithm on synthetic data. Our experiments are evaluated on a broad set of graph structures to demonstrate their applicability beyond the stringent assumptions made in Theorem 1. We compare Algorithm 1 against several baseline approaches that are characterized by independently learning two graphs and identifying their shared structure afterwards. We consider three ensembles of graphs: (i) S_1: sparse Erdős-Rényi random graphs $\mathcal{G}(p, c/p)$; (ii) S_2: graphs with tree structures; and (iii) S_3: graphs with cyclic building blocks with degree $d \leq 3$. The figurative representations of these ensembles are shown in Fig. 3. For our experiments, we generate graph pairs with 200 vertices per graph for ensembles S_1 and S_3, and 255 vertices per graph for S_2. The edge structures are generated such that the subgraph E_{s} spans $q = 20$ vertices for the graph pairs in S_1 and S_3, and $q = 25$ for the graph pairs in S_2. The results are reported for 100 random realizations of the graph pairs.



(a) Ensemble 1 (b) Ensemble 2 (c) Ensemble 3

Figure 2: Probability of error P_{L} for recovering E_{s} with n_{L} iterations.



a. Ensemble 1 b. Ensemble 2 c. Ensemble 3

Figure 3: Representative examples of a graph in three ensembles used for analyzing the performance on simulated data.

Joint versus independent structure learning. First, we study the final probability of error P_{L} defined in (5) in graph pairs with bounded width. Note that an error may occur when either the pruning step of estimating $\hat{V}_{\mathrm{s}}(k)$ fails to include a vertex $u \in V_{\mathrm{s}}$ in $\hat{V}_{\mathrm{s}}$, or the shared structure learning part fails to recover the exact E_{s}. We noted in Section 4.2 that removing the pruning step and the update rule in (14) reduces our algorithm to the algorithm in (Klivans and Meka, 2017) with update rule in (16) for a single graph. To compare *joint versus independent learning*, we run the algorithm in (Klivans and Meka, 2017) to learn structures of two graphs separately and then evaluate the error in the shared subgraph of the estimates. Figure 2 illustrates this comparison in terms of the probability of error versus increasing horizon, where horizon refers to number of iterations. We observe that our algorithm significantly outperforms the baseline approach of (Klivans and Meka, 2017) in all three ensembles.

Comparison with correlation thresholding algorithms. Computing and using empirical pairwise means in our algorithm bear similarities to structure learning algorithms that work based on *correlation thresholding* in the *correlation decay* regimes. Thus, we compare our method's performance to that of the correlation thresholding (CT) Algorithm presented in (Anandkumar et al., 2010). In all ensembles, average degree and λ are chosen appropriately for the correlation decay regime.

In Fig. 4, we show this comparison by plotting the error with respect to the number of vertex measurements. Our joint learning of the shared subgraph outperforms the dual approach that uses CT algorithm independently on two graphs and identifies the shared structure at the end. We note that the slow start of our method is due to the initialization of the multiplicative weight updates in the algorithm. Since each vertex is initialized as a candidate neighbor for all the other vertices, it usually takes some training for weights to converge to the proper values. Eventually, we observe significant gains in terms of sample complexity measure $\mathsf{N}(n_{\mathrm{T}})$ defined in (6) as due to the adaptive aspect of our algorithm, the algorithm gradually focuses on sampling from the vertices that are more likely to be in $\mathcal{G}_{\mathrm{s}}$.

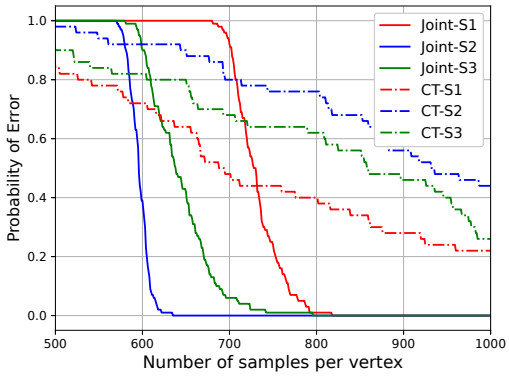


Figure 4: Joint shared subgraph learning and CT algorithm in (Anandkumar et al., 2010) with $\mathsf{N}(n_{\mathrm{T}})$ vertex samples.

Comparison with sparse logistic regression algorithms. (Wu et al., 2019) proposed an algorithm to learn the structure of a single graph by solving an ℓ_1-constrained logistic regression problem for the models with bounded width. We apply this algorithm to learn

two graphs independently and identify their shared structure afterward. To speed up the convergence of their mirror descent updates, we have modified this algorithm with stochastic mirror descent. We run the simulations on random graph pairs from $\mathcal{G}(p, 2/p)$ that has $|V_s| = p/10$ and $\lambda = 0.2$.

Figure 5 offers a number of observations. First, we note that time complexity of the algorithm in (Wu et al., 2019) scales with Tp^2, where T is the number of iterations for mirror descent updates and T scales with $1/\epsilon^4$, where ϵ is the bound for the norm of the error on parameter estimates (see (Wu et al., 2019) for details). This indicates that for a successful structure learning, time complexity grows quickly, especially in large graphs. To illustrate this effect, we run the algorithm of (Wu et al., 2019) for $\{1000, 1250, 1500, 2000, 3000\}$ samples with $T = 100$ (red dots) and $T = 250$ (green dots) iterations. We plot the number of samples for our algorithm both in terms of graph samples and vertex samples per graph, where the latter is significantly smaller. Notice that $\mathsf{N}(n_T)/p$ is plotted in Fig. 5 to illustrate the comparison with (Wu et al., 2019) that uses p vertex measurements for a graph sample.

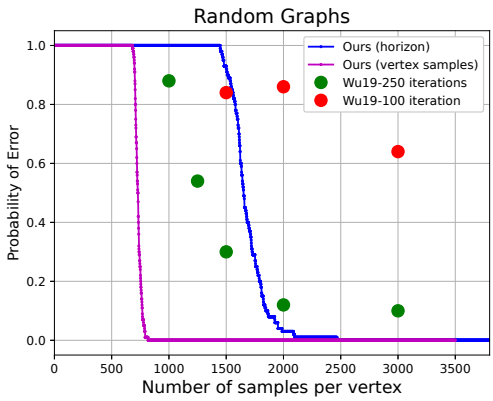


Figure 5: Comparison with (Wu et al., 2019) with respect to n_T and $\mathsf{N}(n_T)$.

Varying the edge parameter λ on sparse random graphs. Next, we study the performance for different edge parameters λ on random graphs $\mathcal{G}(p, 2/p)$. Note that the average degree is chosen to be small so that we will stay in the bounded width regime. Figure 6 shows that for moderately large enough values of λ, our algorithm can achieve a small probability of error. On the other hand, we note that sample complexity of our algorithm scales with $1/\lambda^2$ for a fixed model width. Hence, we expect a performance drop when λ becomes too small.

Varying the average degree of sparse random graphs. We note that our pruning decision rule in (8) is degree independent, and the multiplicative weight updates for structure learning use the maximum degree value only for normalizing the weights. More details can be bound in Appendix A. Next, we create random graphs $\mathcal{G}(p, c/p)$ with $p = 200$ vertices, $\lambda = 0.2$ and $c \in \{1, 2, 3, 4, 5\}$, where c denotes the average degree. Figure 7 shows that our algorithm recovers the shared structure with similar error rates for sparse random graphs with reasonably small average degree values.

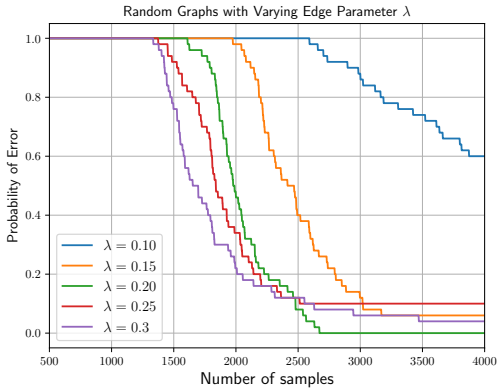


Figure 6: Performance of our method for various λ values.

Finally, we remark that our sample complexity is exponential in model width λd. For random graphs with large values of p, the degree of a vertex follows a Poisson distribution. Moreover, the maximum of these p random variables grows quickly (see (Briggs et al., 2009) for details). Hence, either model width becomes large and sample complexity grows exponentially, or λ happens to be too small to stay in the bounded width regime, which in turns scales the sample complexity with $1/\lambda^2$ and still requires a large number of samples. Therefore, we observe that our algorithm does not perform well for the exact recovery of the shared structure when the random graphs become denser (higher average degree).

Additional experiments. Evaluation of additional aspects of the algorithm, such as the effect of subgraph size, the effect of errors in pruning stage, comparison with a joint learning algorithm that aims to learn the graphs in their entirety, and an application for analysis of the U.S. Senate voting records data are provided in Appendix D.

6 CONCLUSION

The novel problem of learning the shared structure between two Ising models has been considered. An

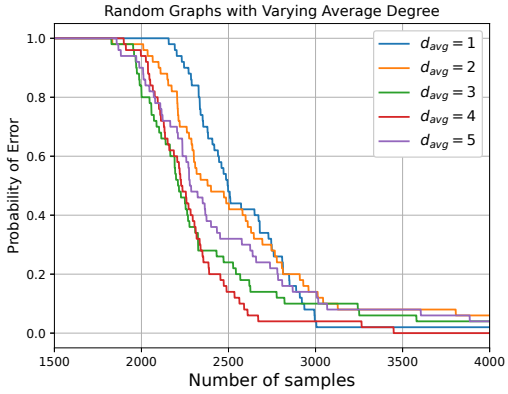


Figure 7: Performance of our method for various average degree values

algorithmic framework has been proposed, which, in contrast to the existing works, is focused on learning the structure of only the shared subgraph between the two graphs. The sample complexity of the framework has been characterized for specific scenarios. The performances of the framework have been numerically evaluated in various ensembles of graphs and shown to outperform naive approach of separately learning two graphs for three baseline approaches.

Acknowledgement

This research was supported in part by RPI-IBM Artificial Intelligence Research Center, IBM Research AI through the AI Horizons Network, and the U. S. National Science Foundation under the CAREER award ECCS-155448 and the grant ECCS-1933107.

References

A. Anandkumar, V. Y. F. Tan, and A. S. Willsky. High dimensional structure learning of Ising models on sparse random graphs. *arXiv:1011.0129*, 2010.

O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine learning research*, 9:485–516, Jun. 2008.

J. Bento and A. Montanari. On the trade-off between complexity and correlation decay in structural learning algorithms. *arXiv:1110.1769*, 2011.

G. Bresler. Efficiently learning Ising models on arbitrary graphs. In *Proc. of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '15, page 771–782, New York, NY, USA, Jun. 2015.

G. Bresler, M. Karzand, et al. Learning a tree-structured Ising model in order to make predictions. *Annals of Statistics*, 48(2):713–737, Apr. 2020.

K. Briggs, L. Song, and T. Prellberg. A note on the distribution of the maximum of a set of poisson random variables. *arXiv:0903.4373*, 2009.

R. L. Buckner and L. M. DiNicola. The brain's default network: updated anatomy, physiology and evolving insights. *Nature reviews. Neuroscience*, 20(10):593–608, Oct. 2019.

X. Chen, F. J. Slack, and H. Zhao. Joint analysis of expression profiles from multiple cancers improves the identification of microRNA–gene interactions. *Bioinformatics*, 29(17):2137–2145, Sep. 2013.

D. M. Chickering. Learning Bayesian networks is NP-complete. In *Learning from Data*, pages 121–130. Springer, New York, NY, 1996.

P. Danaher, P. Wang, and D. M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, Mar. 2014.

C. Daskalakis, N. Dikkala, and G. Kamath. Testing Ising Models. *IEEE Transactions on Information Theory*, 65(11):6829–6852, Nov. 2019.

A. Dobra, C. Hans, B. Jones, J. R. Nevins, G. Yao, and M. West. Sparse graphical models for exploring gene expression data. *Journal of Multivariate Analysis*, 90(1):196–212, Jul. 2004.

J. Fang, L. S. Dongdong, S. Charles, Z. Xu, V. D. Calhoun, and Y.-P. Wang. Joint sparse canonical correlation analysis for detecting differential imaging genetics modules. *Bioinformatics*, 32(15):3480–3488, Nov. 2016.

Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

A. Ghoshal and J. Honorio. Direct estimation of difference between structural equation models in high dimensions. *arXiv:1906.12024*, 2019.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, Feb. 2011.

J. Guo, J. Cheng, E. Levina, G. Michailidis, and J. Zhu. Estimating heterogeneous graphical models for discrete data with an application to roll call voting. *The Annals of Applied Statistics*, 9(2):821 – 848, Jun. 2015.

W. Hoeffding. Probability Inequalities for sums of Bounded Random Variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426, New York, NY, 1994. Springer.

C. J. Honey, R. Kötter, M. Breakspear, and O. Sporns. Network structure of cerebral cortex shapes functional connectivity on multiple time scales. *Proceedings of the National Academy of Sciences*, 104(24): 10240–10245, Jun. 2007.

Y. Jacob, L. Denoyer, and P. Gallinari. Learning latent representations of nodes for classifying in heterogeneous social networks. In *Proc. of the 7th ACM international conference on Web search and data mining*, pages 373–382, New York, NY, Feb. 2014.

A. Klivans and R. Meka. Learning graphical models using multiplicative weights. In *Proc. IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 343–354, Oct. 2017.

S. L. Lauritzen. *Graphical Models*, volume 17. Clarendon Press, May 1996.

J. B. Lewis, K. Poole, H. Rosenthal, A. Boche, A. Rudkin, and L. Sonnet. Voteview: Congressional rollcall votes database. https://voteview.com/, 2020.

S. Liu, J. A. Quinn, M. U. Gutmann, T. Suzuki, and M. Sugiyama. Direct learning of sparse changes in markov networks by density ratio estimation. *Neural Computation*, 26(6):1169–1197, Jun. 2014.

K. Mohan, P. London, M. Fazel, D. Witten, and S.-I. Lee. Node-based learning of multiple Gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488, Feb. 2014.

Y. Okamoto. Finding a Maximum Common Subgraph from Molecular Structural Formulas through the Maximum Clique Approach Combined with the Ising Model. *ACS omega*, 5(22):13064–13068, Jun. 2020.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, Cambridge, UK, 2nd edition, 2009.

P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, Jun. 2010.

A. J. Rothman, P. J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2(none):494 – 515, Jun. 2008.

S. Sihag and A. Tajer. Structure learning of similar ising models: Information-theoretic bounds. In *Proc. IEEE International Symposium on Information Theory*, pages 1307–1311, Paris, France, Jul. 2019a.

S. Sihag and A. Tajer. Structure learning with side information: Sample complexity. In *Proc. Advances in Neural Information Processing Systems*, pages 14357–14367, Toronto, Canada, 2019b.

M. Vuffray, S. Misra, A. Lokhov, and M. Chertkov. Interaction screening: Efficient and sample-optimal learning of Ising models. In *Proc. Advances in Neural Information Processing Systems*, pages 2595–2603, Barcelona, Spain, 2016.

Y. Wang, C. Squires, A. Belyaeva, and C. Uhler. Direct estimation of differences in causal graphs. In *Proc. Advances in Neural Information Processing Systems*, page 3774–3785, Montréal, Canada, 2019.

C. S. Won and H. Derin. Unsupervised segmentation of noisy and textured images using markov random fields. *CVGIP: Graphical models and image processing*, 54(4):308–328, Jul. 1992.

R. Wu, R. Srikant, and J. Ni. Learning loosely connected Markov random fields. *Stochastic Systems*, 3 (2):362–404, Feb. 2013.

S. Wu, S. Sanghavi, and A. G. Dimakis. Sparse logistic regression learns all discrete pairwise graphical models. In *Proc. Advances in Neural Information Processing Systems*, page 8069–8079, Vancouver, Canada, 2019.

S. Yang, Z. Lu, X. Shen, P. Wonka, and J. Ye. Fused multiple graphical lasso. *SIAM Journal on Optimization*, 25(2):916–943, May 2015.

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

S. D. Zhao, T. T. Cai, and H. Li. Direct estimation of differential networks. *Biometrika*, 101(2):253–268, Jun. 2014.

A Algorithm Details

We add a few notes on some steps of Algorithm 1 that are necessary for the implementation.

- **Normalization of weights**: Note that the weight updates in (14) and (15) do not guarantee that $\sum_{v\in\hat{V}_{\rm s}} \kappa_i^{uv} \leq \lambda d$ for all $u \in V_{\rm s}$ in $\mathcal{G}_i$. Therefore, we introduce the normalized weights w_i^{uv} which are evaluated in the k-th iteration as

$$w_i^{uv}(k+1) = \frac{\lambda d \kappa_i^{uv}(k+1)}{\sum_{x\neq u}\left(\kappa_i^{ux}(k+1) + \tilde{\kappa}_i^{ux}(k+1)\right)} \ . \tag{21}$$

- **Pseudo-weights**: We introduce the pseudo-weights $\tilde{\kappa}_i^{uv}$ to accommodate for the setting when the degree of u or v vertex is strictly less than d. The theoretical justification behind the introduction of these weights is included in Appendix B.

- **Dummy weight updates:** We note that update of pseudo-weights as $\tilde{\kappa}_i^{uv}(k+1) = \tilde{\kappa}_i^{uv}(k)\exp(\beta/2)$ implies that these are not affected by the loss and mere technical components. Similarly, for the vertices $u, v \notin \hat{V}_{\rm s}(k)$, we do not have samples from them to compute $l_i^{uv}(k)$ at the k-th iteration and update their corresponding weights using (14). Therefore, updates in (15) refer to keeping the corresponding weights for $u, v \notin \hat{V}_{\rm s}(k)$ pairs effectively unchanged.

B Sample Complexity Analysis

Note that our algorithm consists of two subroutines that are executed in tandem. The first subroutine relates to the pruning of the set of V vertices to adaptively focus on the pairwise relationships among the vertices in $V_{\rm s}$. The second routine is the joint learning of the shared structure in the graph pair which leverages multiplicative weight updates to the vertices of interest in every iteration. For our sample complexity analysis, we assume that $V_{\rm s}$ forms an isolated subgraph in both $\mathcal{G}_1$ and $\mathcal{G}_2$. This assumption allows us to leverage different properties of the Ising model that are necessary for establishing a closed form of the sample complexity. Furthermore, for analysis under this assumption, we can decouple the two subroutines in the following manner: We first evaluate the number of samples that is needed to localize $V_{\rm s}$ with a high likelihood. Next, we evaluate the sample complexity of joint learning of the shared subgraph $\mathcal{G}_{\rm s}$ after $V_{\rm s}$ has been localized with high likelihood.

B.1 Isolating $V_{\rm s}$ vertices through pruning

Before we give the sample complexity analysis for the joint multiplicative weight updates, we will show that the pruning step of the Algorithm 1 localizes $V_{\rm s}$ correctly in the correlation decay regime.

We start by providing the following lemma, which is instrumental in establishing the edge-level decisions.

Lemma 2. *In a ferromagnetic Ising model $\mathcal{G} = (V, E)$ if the minimum distance between two vertices $u, v \in V$ is $\ell > 1$ and λ satisfies $\tanh(\lambda) \leq 1/(L+1)$, where L is the maximum number of paths between any two vertices, then we have*

$$\tanh^{\ell}(\lambda) \leq \mathbb{E}[X^u X^v] \leq (L+1)\tanh^2(\lambda) \ . \tag{22}$$

Proof. For an Ising model, the lower bound on the correlation between any two vertices relates to the shortest path between them (Anandkumar et al., 2010, Lemma 3). This provides the lower bound in (22), where the shortest path between u and v vertices have length $\ell > 1$. Note that for any graph $\mathcal{G}$, the upper bound on the correlation, stated by (Anandkumar et al., 2010, Lemma 3), is bounded as follows:

$$\mathbb{E}[X^u X^v] \leq \min_{b\geq \ell} \sum_{t=\ell}^{b} N_t(u,v)\tanh^t\lambda + |B_b(u)|\tanh^b(\lambda) \ , \tag{23}$$

where $N_t(u,v)$ is the number of paths between u and v of length t and $B_b(u)$ is the set of vertices in the self-avoiding walk tree of $\mathcal{G}$ at a distance b from vertex u. Therefore, in (23), by using $N_t(u,v) \leq L$, $\tanh^t\lambda \leq \tanh^2\lambda$, $|B_b(u)| \leq L(L-1)$ and $|B_b(u)|\tanh^b(\lambda) \leq \tanh^2\lambda$ for any $b \geq \frac{\log(L(L-1))}{\log(L+1)}$, we get the upper bound in (22). □

Now following from the proof of Lemma 1, for any $\epsilon > 0$ and $k = \frac{\alpha \log p}{2\epsilon^2}$ samples, we have

$$\mathbb{P}[\bar{\mathbb{E}}_k[X_i^u X_i^v] \geq \tanh(\lambda) - \epsilon] \geq 1 - \frac{1}{p^\alpha} \ , \quad \forall (u,v) \in E_i, \tag{24}$$

$$\mathbb{P}[\bar{\mathbb{E}}_k[X_i^u X_i^v] \leq (L+1)\tanh^2(\lambda) + \epsilon] \geq 1 - \frac{1}{p^\alpha} \ , \quad \forall (u,v) \notin E_i. \tag{25}$$

Taking a union bound over all possible pairs in both graphs, (24) and (25) hold with probability not smaller than $1 - 2p^{2-\alpha}$. Now notice that if lower bound of an edge $(u,v) \in E_{\mathrm{s}}$ is higher than the upper bound of a non-edge $(u,v) \notin E_{\mathrm{s}}$, then the thresholding decision in (8) also removes the spurious edges with high probability. Formally, if the following equations hold true,

$$(L+1)\tanh^2(\lambda) + \epsilon < \tanh(\lambda) - \epsilon \ , \tag{26}$$

$$\epsilon = \sqrt{\frac{\alpha \log p}{2k}} < \frac{\tanh(\lambda)(1 - (L+1)\tanh^2(\lambda))}{2} \ , \tag{27}$$

$$k > \frac{2\alpha \log p}{\tanh^2(\lambda)(1 - (L+1)\tanh^2(\lambda))^2} \ , \tag{28}$$

then the pruning step ensures that $\hat{V}_{\mathrm{s}} = V_{\mathrm{s}}$. In this context, we add the following lemma.

Lemma 3. *In the correlation decay regime of $\lambda = \Theta(1/L)$, with $k = O(\frac{\alpha \log p}{\lambda^2})$ samples, our pruning step localizes V_{s} exactly with probability at least $(1 - 2p^{2-\alpha})$.*

B.2 Joint Learning of Sparse GLMs

Now that we have localized V_{s} vertices, we will show that the joint learning method will lead to the result in Theorem 1. We start by noting that the Sparsitron algorithm proposed in (Klivans and Meka, 2017) for learning a sparse generalized linear model (GLM) was shown to enable structure learning of a single Ising model due to certain properties of the random variables associated with a degree bounded Ising model. Here, we will build upon the principles adopted in (Klivans and Meka, 2017) to first propose Algorithm 2 to joint learning of two sparse GLMs and characterize its performance. Then we will leverage the performance of Algorithm 2 and the properties of Ising models to complete the proof of Theorem 1.

Algorithm 2 Learning two GLMs jointly

Input β, γ, T pairs of data samples
initialize $w_i^0 = \mathbb{1}_a / a$ for $i \in \{1,2\}$
for a new pair of data sample $k \in \{1, \ldots, T\}$ **do**
  Compute $h_i^k = \frac{w_i^{k-1}}{\|w_i^{k-1}\|_1}$
  Compute losses $\ell_i^k(t) = \frac{1}{2}(\mathbb{1}_p + (\sigma(\omega h_i^k \cdot X(k)) - Y(k)))X(k)$ for $i \in \{1,2\}$
  for $t \in \{1, \ldots, a\}$ **do**
    if $\zeta_k^\gamma(\omega h_1^k(t), \omega h_2^k(t)) = 1$ for k samples **then**
      Update the weights $w_i^k(t) = w_i^{k-1}(t)\exp(\beta(\ell_1^k(t) + \ell_2^k(t))/2)$ for $i \in \{1,2\}$
    else
      Update the weights $w_i^k(t) = w_i^{k-1}(t)\exp(\beta \ell_i^k(t))$ for $i \in \{1,2\}$
    end if
  end for
end for

Define g_1 and g_2 as two pdfs defined in $[-1,1]^a \times \{0,1\}$. Denote (C_i, D_i) as a random sample from g_i, i.e., $(C_i, D_i) \sim g_i$, where $C_i \in [-1,1]^a$ and $D_i \in \{0,1\}$. Also, we denote a collection of k independent and identically distributed (i.i.d.) samples from g_i by $(\mathbf{C}_i^k, \mathbf{D}_i^k)$. We assume that C_i and D_i satisfy the property

$$\mathbb{E}[D_i | C_i] = \sigma(r_i \cdot C_i) \ , \text{ for } i \in \{1,2\} \ , \tag{29}$$

where $\sigma : \mathbb{R} \to [0,1]$ is a non-decreasing 1-Lipschitz function, and $r_i \triangleq [r_i^1, \ldots, r_i^a]$ is a vector of weights, such that, $\|r_i\|_1 \leq \omega$ for $i \in \{1,2\}$ for some $\omega > 0$. Let $\zeta_k^\gamma(j)$ be a decision rule that using k data samples from g_1

and g_2 generates the output

$$\zeta_k^\gamma(j) = \begin{cases} 1, & \text{if } r_1^j = r_2^j \\ 0, & \text{otherwise} \end{cases} , \tag{30}$$

for $j \in \{1, \ldots, a\}$ and the output is correct with a probability larger than $1-\gamma$, for some $\gamma > 0$. We also define $\boldsymbol{\zeta}_k^\gamma(j)$ as a vector consisting of decisions made on upto k data samples and is given by

$$\boldsymbol{\zeta}_k^\gamma(j) \triangleq [\zeta_1^\gamma(j), \ldots, \zeta_k^\gamma(j)] . \tag{31}$$

In this scenario, we propose Algorithm 2 to jointly learn r_1 and r_2 which builds upon the principles of Hedge algorithm in (Freund and Schapire, 1997). Theorem 2 provides the sample complexity of Algorithm 2.

Theorem 2. *Given $T = O(\omega^2(\log(a/\delta\epsilon)/\epsilon^2)$ number of i.i.d. samples from g_1 and g_2, Algorithm 2 forms estimates $\hat{r}_1$ and $\hat{r}_2$, such that, with probability at least $1-\delta$, we have*

$$\mathbb{E}_{g_1,g_2}[(\sigma(\hat{r}_i \cdot C_i) - \sigma(r_i \cdot C_i))^2] \leq \epsilon \ , \ \textit{for } i \in \{1,2\}. \tag{32}$$

Proof. Note that in Algorithm 2, given a set of k samples and a decision vector $\boldsymbol{\zeta}_k^\gamma(j)$, the weight $w_i^k(j)$ for the j-th index in w_i^k is given by

$$w_i^k(j) = w_i^0(j) \prod_{t=1}^{k} \exp(\beta L_i^t(j)) , \tag{33}$$

where

$$L_i^t(j) \triangleq \mathbb{1}_{\{\zeta_t^\gamma(j)\}} \frac{(\ell_1^t(j) + \ell_2^t(j))}{2} + (1 - \mathbb{1}_{\{\zeta_t^\gamma(j)\}}) \ell_i^t(j) , \tag{34}$$

and $\mathbb{1}_{\{\cdot\}}$ is an indicator function. First, we present a result similar to (Freund and Schapire, 1997, Theorem 5), which establishes that the overall regret of an online learning framework given by Algorithm 2 is upper bounded by the regret of the best expert with addition of terms that scale as $O(\sqrt{T \log a}) + \log a$. This result is formalized in the next lemma.

Lemma 4. *Given T data samples and a sequence of decision vectors $\boldsymbol{\zeta}_k^\gamma$, the overall regret corresponding to learning the GLM for g_i in Algorithm 2 is bounded as*

$$\sum_{k=1}^{T} \mathbf{h}_i^k \cdot \mathbf{L}_i^k \leq \min_{t \in \{1,\ldots,a\}} \sum_{k=1}^{T} L_i^k(t) + O(\sqrt{T \log a}) + \log a , \tag{35}$$

where

$$\mathbf{L}_i^k \triangleq [L_i^k(1), \ldots, L_i^k(a)]^\mathsf{T} \quad \textit{and} \quad \mathbf{h}_i^k \triangleq [h_i^k(1), \ldots, h_i^k(a)] , \tag{36}$$

such that $\|\mathbf{h}_i^k\|_1 = 1$ and $h_i^k(t) \geq 0, \forall t \in \{1, \ldots, a\}$.

Proof. Given an instance of decision sequences $\boldsymbol{\zeta}_k^\gamma$ and the corresponding weights w_i^k, we note that

$$\sum_{t=1}^{a} w_i^k(t) = \sum_{t=1}^{a} w_i^{k-1}(t) \exp(\beta L_i^k(t)) . \tag{37}$$

Since we have $L_i^k(t) \in [0,1]$, and from the convexity argument in (Freund and Schapire, 1997), we get

$$\exp(\beta L_i^k(t)) \leq 1 - (1 - \exp(\beta)) L_i^k(t) . \tag{38}$$

Therefore, it readily follows that

$$\sum_{t=1}^{a} w_i^k(t) \leq \sum_{t=1}^{a} w_i^{k-1}(t)(1 - (1 - \exp(\beta)) \mathbf{h}_i^k \cdot \mathbf{L}_i^k) . \tag{39}$$

For $k = T$ and by repeating the steps (37) and (39), we have

$$\sum_{t=1}^{a} w_i^T(t) \leq \sum_{t=1}^{a} w_i^0(t) \prod_{k=1}^{T} (1 - (1 - \exp(\beta))\mathbf{h}_i^k \cdot \mathbf{L}_i^k) \,. \tag{40}$$

By using $\sum_{t=1}^{a} w_i^0(t) = 1$ and the property $1 + x \leq \exp(x), \forall x$, we get

$$\sum_{t=1}^{a} w_i^T(t) \leq \exp(-(1 - \exp(\beta)) \sum_{k=1}^{T} \mathbf{h}_i^k \cdot \mathbf{L}_i^k) \,. \tag{41}$$

The overall regret of the Algorithm 2 is given by $\sum_{k=1}^{T} \mathbf{h}_i^k \cdot \mathbf{L}_i^k$ and from (41), we have

$$\sum_{k=1}^{T} \mathbf{h}_i^k \cdot \mathbf{L}_i^k \leq \frac{-\log(\sum_{t=1}^{a} w_i^T(t))}{1 - \exp(\beta)} \,. \tag{42}$$

Therefore, we have established that any sequence of the loss functions for joint learning of the two GLMs satisfy the same property as the loss function for learning a single GLM in (Klivans and Meka, 2017). Subsequent arguments in Lemma 4 and Lemma 5 in (Freund and Schapire, 1997) complete the proof. □

We will leverage Lemma 4 to characterize T for prediction of r_i next. Corresponding to g_i, we define the random variable

$$V_i^k \triangleq (\mathbf{h}_i^k - r_i/\omega) \cdot \mathbf{L}_i^k \,, \tag{43}$$

such that, $V_i^k \in [-1, 1]$. Based on V_i^k, we define another sequence of random variables

$$Z_i^k = V_i^k - \mathbb{E}[V_i^k | (\mathbf{C}_1^{k-1}, \mathbf{D}_1^{k-1}), (\mathbf{C}_2^{k-1}, \mathbf{D}_2^{k-1})] \,. \tag{44}$$

Then, we have $Z_i^k \in [-2, 2]$. Note that using Azuma's inequality on martingales with bounded differences, we find that the following event holds with probability at least $1 - \delta$,

$$\sum_{k=1}^{T} \mathbb{E}[V_i^k | ((\mathbf{C}_1^{k-1}, \mathbf{D}_1^{k-1}), (\mathbf{C}_2^{k-1}, \mathbf{D}_2^{k-1})] \leq \sum_{k=1}^{T} V_i^k + O(T \log(1/\delta) \,. \tag{45}$$

Furthermore, note that

$$\mathbb{E}[V_i^k | ((\mathbf{C}_1^{k-1}, \mathbf{D}_1^{k-1}), (\mathbf{C}_2^{k-1}, \mathbf{D}_2^{k-1})] = \frac{1}{\omega} \mathbb{E}[\omega \mathbf{h}_i^k - r_i) \cdot \mathbf{L}_i^k] \,, \tag{46}$$

and

$$\mathbb{E}[V_i^k| \geq \frac{1}{4\omega} \mathbb{E}[\sigma(\omega \mathbf{h}_i^k \cdot C_i) - \sigma(r_i \cdot C_i))^2] \,, \tag{47}$$

where (47) follows from the inequality that $\forall a, b \in \mathbb{R}, (a - b)(\sigma(a) - \sigma(b)) \geq (\sigma(a) - \sigma(b))^2$ and that the lower bound corresponds to correct decisions $\zeta_k^{\gamma}(t) = 1$ for all $t \in \{1, \ldots, a\}$, irrespective of the confidence γ. Then, it follows from (36), (45), and (47) that with probability at least $1 - \delta$, we have

$$\begin{aligned} &\frac{1}{4\omega} \sum_{k=1}^{T} \mathbb{E}[\sigma(\omega \mathbf{h}_i^k \cdot C_i) - \sigma(r_i \cdot C_i))^2] \\ &\quad \leq \min_{t \in \{1, \ldots, a\}} \sum_{k=1}^{T} L_i^k(t) - \sum_{k=1}^{T} (r_i/\omega) \cdot \mathbf{L}_i^k + O(\sqrt{T \log a}) + \log a + O(T \log(1/\delta)) \,. \end{aligned} \tag{48}$$

Clearly, when $\|r_i\|_1 = \omega$, we have that

$$\min_{t\in\{1,\dots,a\}} \sum_{k=1}^{T} L_i^k(t) - \sum_{k=1}^{T} (r_i/\omega)\cdot \mathbf{L}_i^k \leq 0\ . \tag{49}$$

When we have $\|r_i\|_1 < \omega$, we can augment r_i with a pseudo vector $\tilde{r}_i$, s.t., $\|[r_i, \tilde{r}_i]\|_1 = \omega$ and the random vector C_i with an additional element that corresponds to 0 such that $\tilde{r}_i$ corresponds to the weight associated with 0 and proceed further. This also motivates the inclusion of auxiliary weights $\tilde{\kappa}_i^{uv}$ in Algorithm 1. Next, we note that with probability at least $1-\delta$, we have

$$\frac{1}{4\omega}\sum_{k=1}^{T} \mathbb{E}[\sigma(\omega \mathbf{h}_i^k \cdot C_i) - \sigma(r_i \cdot C_i))^2] = O(\sqrt{T\log a}) + O(\log a) + O(T\log(1/\delta)\ . \tag{50}$$

Therefore, for $T = O(\omega^2 \log(a/\delta)/\epsilon^2$, we must have that with probability at least $1-\delta$,

$$\min_{k\in\{1,\dots,T\}} \mathbb{E}[\sigma(\omega \mathbf{h}_i^k \cdot C_i) - \sigma(r_i \cdot C_i))^2] \leq \epsilon\ . \tag{51}$$

□

B.3 Learning Ising Models Jointly

To complete the proof of Theorem 1, we note that if $\hat{V}_{\rm s}$ is an MRF, we have

$$\mathbb{E}[B_i^u] = \frac{1}{1+\exp(2\lambda \sum\limits_{\{v:(u,v)\in E_i^s\}} X_i^u X_i^v)}\ . \tag{52}$$

Therefore, every vertex $u \in \hat{V}_{\rm s}$ can determine its neighborhood in $\mathcal{G}_1$ and $\mathcal{G}_2$ using Algorithm 2 by setting σ to be a sigmoid function, $\omega = \lambda d$, $D_i = B_i^u$ in $\mathcal{G}_i$, and $a = |\hat{V}_{\rm s}| - 1$. In this scenario, we have the following lemma in the context of Ising models that is equivalent to Theorem 2.

Lemma 5. *For a u vertex in an Ising model spanned by $\hat{V}_{\rm s}$, given $n_{\rm T} = O\left(\frac{\lambda^2 d^2}{\epsilon^2}\log\frac{|\hat{V}_{\rm s}|}{\rho\epsilon}\right)$ number of pairs of samples from $\hat{V}_{\rm s}$ vertices in $\mathcal{G}_1$ and $\mathcal{G}_2$, Stage 2 of Algorithm 1 produces at least one edge structure E_i^k for $k \in \{1,\dots,n_{\rm T}\}$, such that, with probability at least $1-\frac{\rho}{|\hat{V}_{\rm s}|^2}$,*

$$\mathbb{E}\left[\sigma\left(-2\sum_{\{v:(u,v)\in E_i^k\}} \lambda X_i^v\right) - \sigma\left(-2\sum_{\{v:(u,v)\in E_i^{\rm s}\}} \lambda X_i^v\right)\right] \leq \epsilon\ , \quad \forall \epsilon > 0. \tag{53}$$

Recalling that we can localize all the q vertices of $V_{\rm s}$ exactly with $O(\frac{\log p}{\lambda^2})$ samples, the statement of the Theorem 1 follows from Lemma 5, (Bresler, 2015, Lemma 2.1) for degree bounded Ising models and (Klivans and Meka, 2017, Lemma 4.3).

By combining Lemma 3 and Lemma 5, we complete the proof of Theorem 1.

Remark 2. *We conjecture that the above analysis provided us with an upper bound on the number of samples sufficient for learning $\mathcal{G}_{\rm s}$ as we ignore the impact of multiplicative weight updates made for joint structure learning till the iteration when pruning has localized $V_{\rm s}$ successfully. However, in practice, for graphs without the strong assumption on structure for $V_{\rm s}$, we observe that for the same number of samples, our algorithm performs substantially better than learning the two graphs individually even under the correlation decay regime (refer to Fig. 4, where the structure learning algorithms leverage same pairwise statistics as our pruning subroutine), indicating that the joint structure learning subroutine converges towards learning the true structure of $\mathcal{G}_{\rm s}$ simultaneously as the pruning subroutine enables the estimate $\hat{V}_{\rm s}$ to converge to $V_{\rm s}$.*

C Necessary Conditions for Recovering $(V_{\mathrm{s}}, E_{\mathrm{s}})$

In this section, we briefly comment on the necessary conditions for recovering the subgraph $(V_{\mathrm{s}}, E_{\mathrm{s}})$ under perfect pruning. We note that in general, the joint pdf of $\mathbf{X}_1$ and $\mathbf{X}_2$ denoted by $f(\mathbf{X}_1, \mathbf{X}_2)$ is given by

$$f(\mathbf{X}_1, \mathbf{X}_2) = \frac{1}{Z_{12}} \exp\left(\sum_{(u,v)\in E_{\mathrm{s}}} \lambda(X_1^u X_1^v + X_2^u X_2^v) + \sum_{(u,v)\in \tilde{E}_1} \lambda X_1^u X_1^v + \sum_{(u,v)\in \tilde{E}_2} \lambda X_2^u X_2^v \right) , \tag{54}$$

where Z_{12} is the partition function that ensures $f(\mathbf{X}_1, \mathbf{X}_2)$ is a valid pmf, and we have defined $\tilde{E}_1 \triangleq E_1 \backslash E_{\mathrm{s}}$ and $\tilde{E}_2 \triangleq E_2 \backslash E_{\mathrm{s}}$. The class of graphs associated with $\mathcal{G}_{\mathrm{s}}$ is given by by $\mathcal{I}_p^{\mathrm{s}}$ and formally defined in Definition 1. Following in the main paper, we have defined $\mathcal{I}_p^{\mathrm{s}}(\mathcal{G}_{\mathrm{s}}) \subseteq \mathcal{I}_p \times \mathcal{I}_p$ as the class of all possible pairs of Ising models whose shared structure is given by $\mathcal{G}_{\mathrm{s}}$, and denoted the set of random variables associated with V_{s} in $\mathcal{G}_i$ by $\mathbf{X}_i^{\mathrm{s}}$ and those with $V \backslash V_{\mathrm{s}}$ by $\mathbf{X}_i^{\mathrm{c}}$. Accordingly, the marginal joint pmf of the random variables $\mathbf{X}_i^{\mathrm{s}}$ is given by

$$\begin{aligned} \tilde{f}(\mathbf{X}_1^{\mathrm{s}}, \mathbf{X}_2^{\mathrm{s}}) \triangleq \frac{1}{|\mathcal{I}_p^{\mathrm{s}}(\mathcal{G}_{\mathrm{s}})|} \exp\left(\sum_{(u,v)\in E_{\mathrm{s}}} \lambda(X_1^u X_1^v + X_2^u X_2^v) \right) \\ \times \left(\sum_{\mathbf{X}_1^{\mathrm{c}}, \mathbf{X}_2^{\mathrm{c}}} \sum_{(\tilde{E}_1, \tilde{E}_2)\in \mathcal{I}_p^{\mathrm{s}}(\mathcal{G}_{\mathrm{s}})} \frac{1}{Z_{\tilde{E}_1, \tilde{E}_2}} \times \exp\left(\sum_{(u,v)\in \tilde{E}_1} \lambda X_1^u X_1^v + \sum_{(u,v)\in \tilde{E}_2} \lambda X_2^u X_2^v) \right) \right) , \end{aligned} \tag{55}$$

where $Z_{\tilde{E}_1, \tilde{E}_2}$ is a partition function associated with pdf of the pair of Ising models with edge structures $\tilde{E}_1$ and $\tilde{E}_2$ unique to $\mathcal{G}_1$ and $\mathcal{G}_2$, respectively. Clearly, finding a closed-form for $\tilde{f}(\mathbf{X}_1^{\mathrm{s}}, \mathbf{X}_2^{\mathrm{s}})$ is intractable in general and performing marginal inference on Ising models is an open research problem with many approximation methods.

However, in certain scenarios, the pdf $\tilde{f}$ conforms to MRF properties. For instance, if $(V_{\mathrm{s}}, E_{\mathrm{s}})$ forms a tree structure in graphs with $L = 1$ or an isolated subgraph in graphs with arbitrary L, $\tilde{f}(\mathbf{X}_1^{\mathrm{s}}, \mathbf{X}_2^{\mathrm{s}})$ is given by

$$\tilde{f}(\mathbf{X}_1^{\mathrm{s}}, \mathbf{X}_2^{\mathrm{s}}) = \frac{1}{\hat{Z}_{12}} \exp\left(\sum_{(u,v)\in E_{\mathrm{s}}} \lambda(X_1^u X_1^v + X_2^u X_2^v) \right) . \tag{56}$$

We remark that while (56) captures the connectivity of $(V_{\mathrm{s}}, E_{\mathrm{s}})$ for general Ising models, it ignores any long range correlations existing between the random variables X_i^u and X_i^v due to the existence of multiple paths between u and v vertices in graph $\mathcal{G}_i$.

For completeness, we list the conditions on the number of samples required in the structure learning stage of our framework under perfect pruning and when subgraphs spanned by V_{s} form an MRF in both graphs. Note that the scenario with perfect pruning is sufficient to compare our results on the average sample complexity of our algorithmic framework since we can isolate V_{s} correctly with high probability in correlation decay regime. The following theorem provides the necessary condition on the number of samples for recovering shared graph structure $(V_{\mathrm{s}}, E_{\mathrm{s}})$ in the context of tree-structured graphs.

Theorem 3. *When $\mathcal{G}_1$ and $\mathcal{G}_2$ belong to a family of tree structured Ising models and the shared structure E_{s} forms a tree, any graph decoder that achieves $\mathsf{P}(\mathcal{I}_p^{\mathrm{s}}) \leq 1/2$ must have $n_{\mathrm{L}} = \Omega\left(\frac{e^{\lambda}}{\lambda \tanh \lambda} \log q\right)$ number of samples from V_{s} vertices.*

Furthermore, for an isolated subgraph $(V_{\mathrm{s}}, E_{\mathrm{s}})$, the necessary conditions based on the results in (Sihag and Tajer, 2019b) for jointly recovering E_{s} are provided in Theorem 1.

Theorem 4. *For a pair of graphs $\mathcal{G}_1$ and $\mathcal{G}_2$ in the family of Ising models with an isolated subgraph $(V_{\mathrm{s}}, E_{\mathrm{s}})$, any graph decoder that achieves $\mathsf{P}(\mathcal{I}_p^{\mathrm{s}}) \leq \delta - \frac{1}{\log q}$ must have*

$$n_{\mathrm{L}} \geq \max\left\{ \frac{\log q}{\lambda \tanh \lambda}, \frac{\exp(\lambda d)}{\lambda d \exp(\lambda)} \log qd \right\} , \tag{57}$$

number of samples from V_{s} vertices.

D Additional Experiments

Joint versus independent learning of structurally similar graphs. In this section, we illustrate the gains in sample complexity per graph by jointly learning the structures of two graphs using Algorithm 3 that uses the multiplicative weight updates corresponding to the means of the loss functions in (14) as opposed to learning them independently using the algorithm in (Klivans and Meka, 2017) using (16). The difference from the experiments given in the main paper is that, here we aim to learn the complete structures of the two graphs, and assume $V_{\rm s}$ is known. For this purpose, we run our experiments on Ensemble 3 in Fig. 3 which has maximum degree 3. The structural similarity between the two models is quantified by a parameter $\mu \in (0,1]$ in a fashion similar to that defined in (Sihag and Tajer, 2019a, Definition 1). In this setting, Algorithm 1 without pruning step and known $V_{\rm s}$ is equivalent to Algorithm 3.

Algorithm 3 Joint structure learning algorithm for recovering E_1, E_2 when structural similarity is known (Sihag and Tajer, 2019a)

Input $V_{\rm s}$, $n = T + M$ pairs of data samples, $\beta = \log\left(1/(1+\sqrt{\log p/T})\right)$
Initialize $\kappa_i^{uv}(1) = 1/(p-1)$, $\tilde{\kappa}_i^{uv}(1) = 1/(p-1)$ and $w_i^{uv}(1) = 0$ for all $u \neq v \in V$ and $i \in \{1,2\}$
for a new pair of data sample $k \in \{1,\ldots,T\}$ **do**
    For each $u \in V$, compute $b_i^u(k) = \sum_{v \neq u, v \in V} w_i^{uv}(k) X_i^v(k)$
    for each pair $u, v \in V$, $u \neq v$ **do**
        Compute losses $\ell_i^{uv}(k)$
        if $u \in V_{\rm s}, v \in V_{\rm s}$ **then**
            Update the weights κ_i^{uv} according to (14) and $\tilde{\kappa}_i^{uv}(k+1) = \tilde{\kappa}_i^{uv}(k)\exp(\beta/2)$ for $i \in \{1,2\}$
        else
            Update the weights κ_i^{uv} according to (16) and $\tilde{\kappa}_i^{uv}(k+1) = \tilde{\kappa}_i^{uv}(k)\exp(\beta/2)$ for $i \in \{1,2\}$
        end if
    end for
    for each pair $u \neq v$ **do**
        Compute normalized weights $w_i^{uv}(k+1)$ according to (21)
    end for
    Compute estimates $\mathcal{G}_1^k$ and $\mathcal{G}_2^k$ such that for every pair $u \neq v$ in $\mathcal{G}_i^k$, an edge exists if $w_i^{uv} \geq \lambda/2$
    Compute empirical risks ϵ_i^k
end for
return Graphs $\mathcal{G}_i^t : t = \text{argmin}_k\, \epsilon_i^k$

Figure 8 illustrates the comparison of the mean performance of Algorithm 3 for recovering graph pairs with different structural similarity against recovering them independently using the algorithm in (Klivans and Meka, 2017) over 1000 random instances of graph pairs. The probability of error counts the fraction of the instances at which the true graph pair was not recovered exactly in any of the iterations when the online learning algorithm was run up to a horizon indicated on the horizontal axis.

Clearly, our algorithm outperforms the independent structure learning algorithm for $\mu = 0.25, 0.5$ and 1. When $\mu = 1$, the graph pairs are identical and therefore, Algorithm 1 is equivalent to processing the data $\mathbf{X}_1^n$ and $\mathbf{X}_2^n$ in parallel with 2 processing units that process one graph sample each in every iteration with an exchange of pairwise loss functions between the two. This indicates that Algorithm 1 outperforms by processing 2 graph samples in every iteration up to a horizon T as compared to an approach that sequentially processes 1 graph sample up to a horizon T.

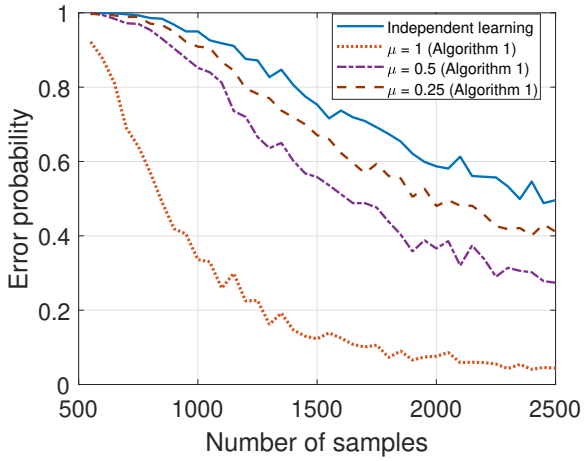


Figure 8: Error probability versus horizon (T) or the number of samples for each graph.

Our algorithm vs. naive joint learning benchmark. A different baseline from the ones presented in the main paper can be the joint learning of two graphs in their entirety, without aiming to learn only the shared structure. We can achieve this baseline by removing pruning part of our algorithm, i.e., simply replacing Step. 18 of 1 with individual weight updates. Figure 9 illustrates that our algorithm requires significantly less number of samples per vertex on both random and tree structured graph pairs.

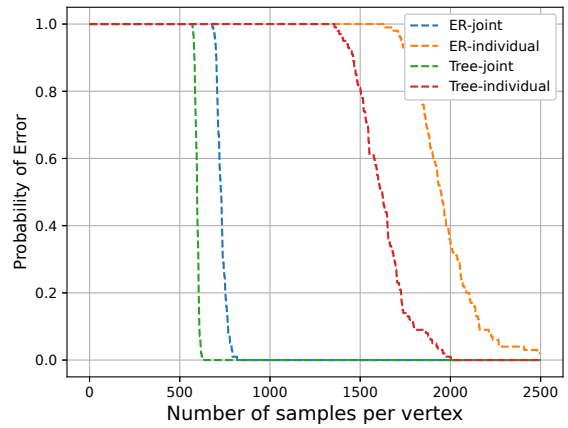


Figure 9: Error probability versus sample complexity

The effect of pruning errors. We remark that if the pruning stage output significantly deviates from the correct E_{s}, our algorithm still has the room to make the correct decisions as the structure learning algorithm runs independently of the pruning step for at least $k > \alpha \log p / \tanh^2 \lambda$ number of iterations, during which the weights for all pairwise combinations in the graph are learnt. Therefore, if pruning step makes significantly wrong decisions and terminates updating the weights for certain edges in E_{s}, we expect the degradation in performance to be controlled. We tested this on Erdős-Rényi random graph pairs that have 12 shared edges, and we intentionally stopped updates for 3 of those edges in $\hat{V}_{\mathrm{s}k}$. We observe in Fig. 10 that, there is approximately 10% increase in sample complexity of recovering the shared graph correctly, indicating that the algorithm was robust.

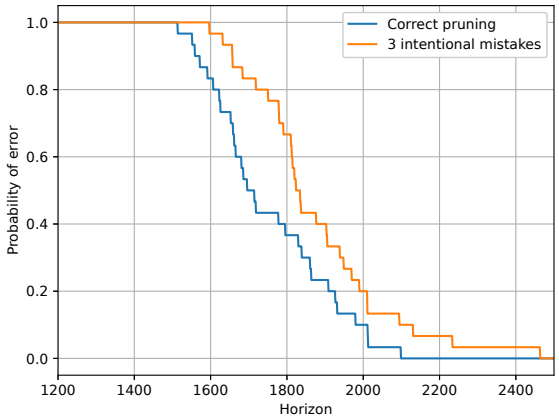


Figure 10: Effect of forced pruning errors

The effect of subgraph size. Size of the subgraph q appears in the sample complexity (20), which indicates that for a fixed target performance, doubling q increases the sample complexity by $\frac{1}{\lambda^2}\exp(\lambda d)$. Figure 11 illustrates the effect of increasing graph size for Erdős-Rényi random graphs with 200 vertices.

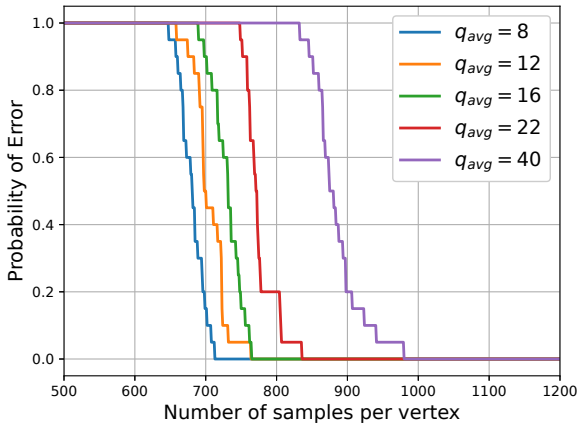
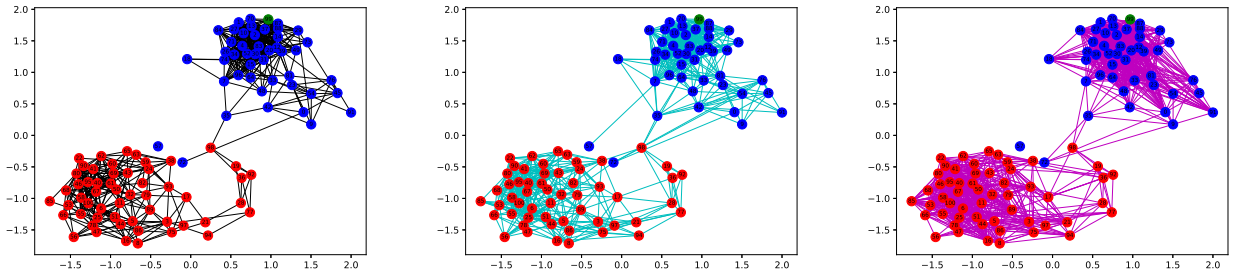


Figure 11: Error probability versus sample complexity

Application to voting records. We have tested our algorithm on senate voting data of 109-th congress (2005-2006 period) (Lewis et al., 2020) which has been previously analyzed in (Guo et al., 2015). There are 44 Democrats, 55 Republicans and 1 independent senator at this term. Each senator was linked to a vertex in the Ising model, and their 'Yes' vote was associated with the state $+1$ and the 'No' vote was associated with the state -1. Due to the bipartisanship in the U.S. Senate, the voting behavior of any senator was likely to be correlated with that of other senators with a similar political affiliation. We aimed to learn the shared structure between the graphs $\mathcal{G}_{\text{pass}}$ and $\mathcal{G}_{\text{reject}}$, where the edge structures of $\mathcal{G}_{\text{pass}}$ and $\mathcal{G}_{\text{reject}}$ represented correlations among the voting behaviors of different senators in the "Passed" and "Rejected" bills, respectively.

Figure 12a illustrates the shared structure between $\mathcal{G}_{\text{pass}}$ and $\mathcal{G}_{\text{reject}}$ obtained using our framework. Figure 12b and 12c illustrate the individual structures of $\mathcal{G}_{\text{pass}}$ and $\mathcal{G}_{\text{reject}}$ learnt using the algorithm in (Klivans and Meka, 2017). The comparison of the shared structure in Fig. 12a to that of the graphs in Fig. 12b and Fig. 12c reveals that a significant number of edges linking different senators exist in only one graph.



(a) Shared $\mathcal{G}_{\text{pass}}$ and $\mathcal{G}_{\text{reject}}$ (b) Structure of $\mathcal{G}_{\text{pass}}$ (c) Structure of $\mathcal{G}_{\text{reject}}$

Figure 12: Learned Structure of Senators of 109th Congress. Blue, Red, and Green vertices represent Democrat, Republican, and Independent senators respectively.