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 $(u,v) \in E_{\rm s}$ is higher than the upper bound of a non-edge $(u,v) \notin E_{\rm s}$, 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 $\hat{V}_{\rm s} = V_{\rm s}$. 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 $V_{\rm s}$ exactly with probability at least $(1 - 2p^{2-\alpha})$.*

B.2 Joint Learning of Sparse GLMs

Now that we have localized $V_{\rm s}$ 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 $\beta$, $\gamma$, $T$ pairs of data samples
initialize $w_i^0 = \mathbb{1}_a/a$ for $i \in \{1,2\}$
for a new pair of data sample $k \in \{1,\dots,T\}$ 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 $t \in \{1,\dots,a\}$ 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 $r_i \triangleq [r_i^1, \dots, r_i^a]$ 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

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

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

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

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 $\sum\limits_{t=1}^{a} w_i^0(t) = 1$ 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 $\sum\limits_{k=1}^{T} \mathbf{h}_i^k \cdot \mathbf{L}_i^k$ and from (41), we have

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

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

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

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 $1-\frac{\rho}{|\hat{V}_{\rm s}|^2}$,*

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

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 $(V_{\rm s}, E_{\rm s})$

In this section, we briefly comment on the necessary conditions for recovering the subgraph $(V_{\rm s}, E_{\rm s})$ 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

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

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

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

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 $\tilde{f}(\mathbf{X}_1^{\rm s}, \mathbf{X}_2^{\rm s})$ 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 $(V_{\rm s}, E_{\rm s})$ forms a tree structure in graphs with $L = 1$ or an isolated subgraph in graphs with arbitrary L, $\tilde{f}(\mathbf{X}_1^{\rm s}, \mathbf{X}_2^{\rm s})$ is given by

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

We remark that while (56) captures the connectivity of $(V_{\rm s}, E_{\rm s})$ 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 $V_{\rm s}$ 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 $V_{\rm s}$ 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 $(V_{\rm s}, E_{\rm s})$ 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 $E_{\rm s}$ forms a tree, any graph decoder that achieves $\mathsf{P}(\mathcal{I}_p^{\rm s}) \leq 1/2$ must have $n_{\rm L} = \Omega\left(\frac{e^\lambda}{\lambda \tanh\lambda} \log q\right)$ number of samples from $V_{\rm s}$ vertices.*

Furthermore, for an isolated subgraph $(V_{\rm s}, E_{\rm s})$, the necessary conditions based on the results in (Sihag and Tajer, 2019b) for jointly recovering $E_{\rm s}$ 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 $(V_{\rm s}, E_{\rm s})$, any graph decoder that achieves $\mathsf{P}(\mathcal{I}_p^{\rm s}) \leq \delta - \frac{1}{\log q}$ must have*

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

number of samples from $V_{\rm s}$ 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_{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_{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 $V_{\mathrm{s}}$, $n = T + M$ 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 $k \in \{1,\dots,T\}$ 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 $u \in V_{\mathrm{s}}, v \in V_{\mathrm{s}}$ then
            Update the weights $\kappa_i^{uv}$ 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 $\kappa_i^{uv}$ 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 $\epsilon_i^k$
end for
return Graphs $\mathcal{G}_i^t : t = \mathrm{argmin}_k\, \epsilon_i^k$
```

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.

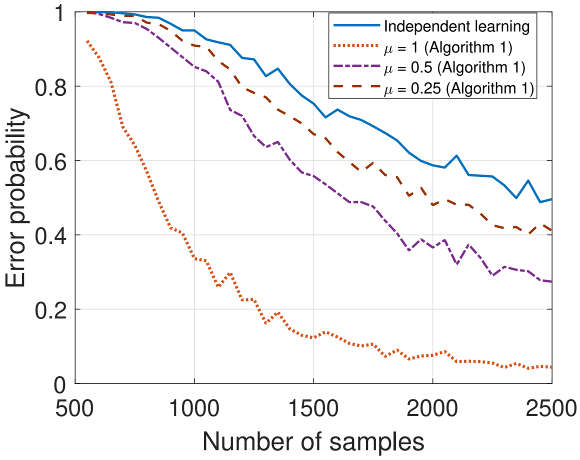


Figure 8: Error probability versus horizon (T) or the number of samples for each graph.

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.

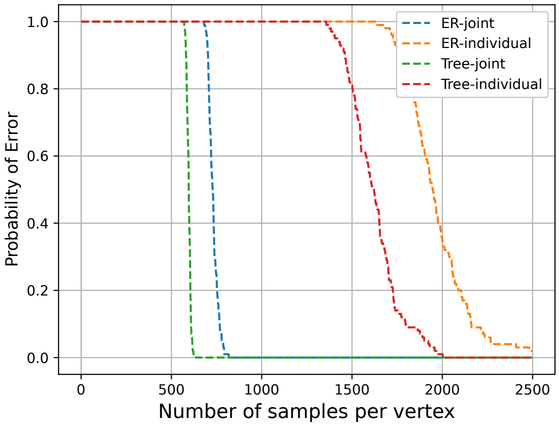


Figure 9: Error probability versus sample complexity

The effect of pruning errors. We remark that if the pruning stage output significantly deviates from the correct $E_{\rm s}$, 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 $E_{\rm s}$, 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 $\hat{V}_{{\rm s}k}$. 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.

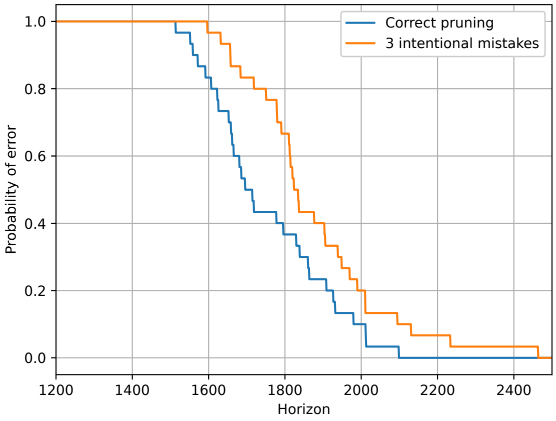


Figure 10: Effect of forced pruning errors

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.

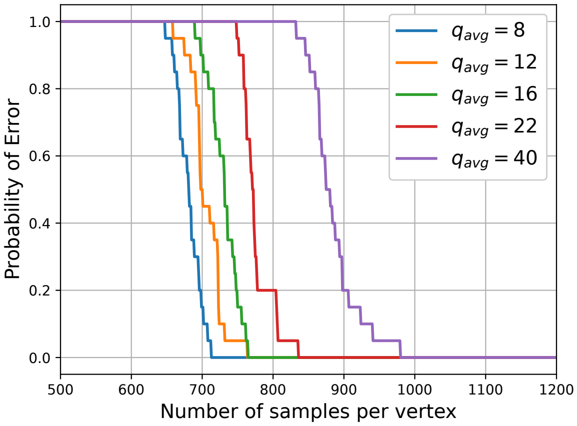
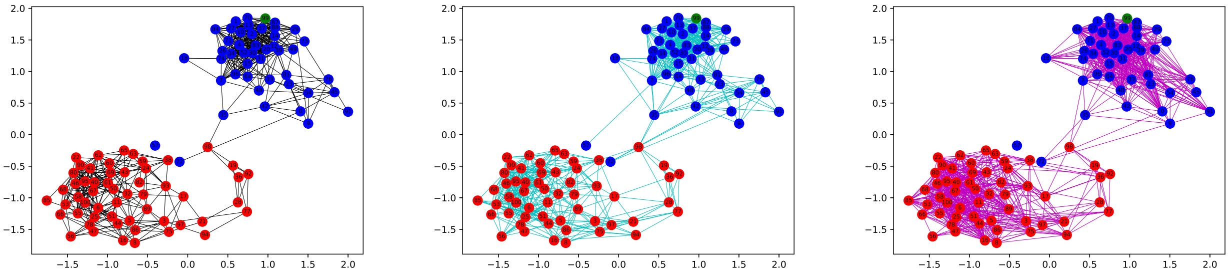


Figure 11: Error probability versus sample complexity

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.