From Boltzmann Machines to Neural Networks and Back Again

Surbhi Goel
Microsoft Research NYC
surbgoel@microsoft.com

Adam Klivans
University of Texas at Austin
klivans@cs.utexas.edu

Frederic Koehler
MIT
fkoehler@mit.edu

Abstract

Graphical models are powerful tools for modeling high-dimensional data, but learning graphical models in the presence of latent variables is well-known to be difficult. In this work we give new results for learning Restricted Boltzmann Machines, probably the most well-studied class of latent variable models. Our results are based on new connections to learning two-layer neural networks under ℓ_∞ bounded input; for both problems, we give nearly optimal results under the conjectured hardness of sparse parity with noise. Using the connection between RBMs and feedforward networks, we also initiate the theoretical study of *supervised RBMs* [1], a version of neural-network learning that couples distributional assumptions induced from the underlying graphical model with the architecture of the unknown function class. We then give an algorithm for learning a natural class of supervised RBMs with better runtime than what is possible for its related class of networks without distributional assumptions.

1 Introduction

Graphical models are a powerful framework for modelling high-dimensional distributions in a way that is interpretable and enables sophisticated forms of inference and reasoning. They are extensively used in a variety of disciplines including the natural and social sciences where they have been used to model the structure of gene regulatory networks, of connectivity in the brain, and the flocking behavior of birds [2]. In many contexts, the structure of interactions between different observed variables is unknown a priori and the goal is to infer this structure in a sample-efficient way from data. There has been decades of research on various formulations of this problem, both theoretically and empirically: for example, provable algorithms have been developed for learning tree-structured graphical models [3], for learning models on graphs of bounded tree-width [4], for learning Ising models on general graphs of bounded degree [5, 6, 7, 8] and in a variety of other contexts like Gaussian graphical models (e.g. [9]). For the most part, the main interest has been on learning under the assumption that the underlying model is sparse. Sparsity is a natural assumption since many applications are in a sample-starved regime where the learning problem is information-theoretically impossible without sparsity. Sparse models are generally considered to be more interpretable than their dense counterparts since they satisfy large numbers of conditional independence relations.

A major challenge in probabilistic inference from data is the presence of latent or confounding variables which are unobserved and may create complicated higher-order dependencies between the observed variables. Specifically in the context of learning undirected graphical models, it is well known that even if the underlying graphical model is well-behaved, if only a subset of the variables are observed then the resulting marginal distribution can still be extremely complicated, e.g. simulating the uniform distribution over satisfying assignments of an arbitrary circuit [10], which makes the learning problem computationally intractable. On the other hand, under certain assumptions we know that learning graphical models with latent variables can be both computationally

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

and statistical tractable; for example, the setting of tree-structured models with latent variables has been extensively studied in the context of phylogenetic reconstruction, see e.g. [11, 12]. However, in non-tree-structured models there are comparatively few positive results for recovering latent variable models in a computationally efficient fashion. One of the few exceptions is in the Gaussian case, where [13] gave a positive result; this setting is very special, as latent variable GGMs *do not* have higher-order interactions, but in fact are equivalent to GGMs with cliques.

Restricted Boltzmann Machines. In this work, we will focus on a latent variable model popularized in the neural network literature known as the *Restricted Boltzmann Machine* (RBM) (see e.g. [1, 14]) which has been applied to problems such as dimensionality reduction and collaborative filtering [15, 16, 17]. It is also perhaps the most canonical version of an Ising model with latent variables. The RBM describes a joint distribution over observed random variables X valued in $\{\pm 1\}^{n_1}$ and latent variables H valued in $\{\pm 1\}^{n_2}$

$$\Pr(X = x, H = h) \propto \exp\Big(\langle x, Wh\rangle + \langle b^{(1)}, x\rangle + \langle b^{(2)}, h\rangle\Big)$$

where the *weight matrix* W is an arbitrary $n_1 \times n_2$ matrix and *external fields/biases* $b^{(1)} \in \mathbb{R}^{n_1}$ and $b^{(2)} \in \mathbb{R}^{n_2}$ are arbitrary, and X is referred to as the vector of *visible unit* activations and H the vector of *hidden unit* activations. In the learning problem, we are given access to i.i.d. samples of X but do not get to observe H. It is not hard to see that in the special case where the hidden nodes are constrained to have degree 2, the class of marginal distributions on X induced by RBMs is exactly the class of Ising models (pairwise binary graphical models), so the general RBM can be thought of as a natural generalization of fully-observed Ising models, for which the learning problem is well-understood. For hidden units with larger degree, the marginal distribution can be an arbitrary Markov Random Field [18]. We also note that the parameters of the RBM are not identifiable even given an infinite number of samples [18], so our goal for learning the RBM is generally speaking to learn the distribution or related structural properties (e.g. the Markov blankets of the nodes in X).

Previous work on Learning RBMs. The most popular heuristic for learning RBMs is the *contrastive divergence* algorithm (see [1]), but there is no guarantee it will succeed. In recent work [18, 19], the first provable algorithms were developed for learning RBMs, under the assumptions that the model is (1) sparse and (2) ferromagnetic. On the other hand, it was shown in [18] that learning general sparse RBMs is computationally intractable in general, because the conjecturally hard problem of learning a *sparse parity with noise* [20] can be embedded into a sparse RBM with a constant number of hidden units. The assumption of ferromagneticity (that variables are only positively correlated, not negatively correlated) rules out this example and plays a crucial role in the analysis of these works. Without ferromagneticity, viewing the marginal on X as a general Markov Random Field allows for using prior work [8] to give learning algorithms with runtime $n^{O(d_H)}$ where d_H is the maximum degree of a hidden node. This matches the lower bound of learning sparse parity with noise mentioned previously.

To summarize, the best previous results for learning RBMs either (1) make the assumption of ferromagneticity which makes building sparse parities impossible or (2) ignore all of the structure of the RBM except the max hidden degree, and pay the price of a $n^{\Theta(d_H)}$ runtime. This leaves open the question of developing algorithms whose runtime depends on some natural notion of a *complexity* measures of the RBM.

Our Results. In this paper, we design an algorithm that is adaptive to a *norm* based complexity measure of the RBM, and often outperforms approach (2) above significantly, while not eliminating the possibility of negative correlation completely as in (1). The key idea of our approach is to develop a novel connection between learning RBMs and their historical relative, feedforward neural networks. This connection allows us to establish new results for learning RBMs, by proving new results about learning feedforward neural networks (Section 2).

Our connection also validates the idea of a so-called *supervised RBMs* as a natural distributional setting for classification with feedforward networks. Supervised RBMs, proposed by Hinton [1], treat one visible unit of the RBM as the label and the other visible units as the input to the classifier. This allows us to use the connection in the "reverse" direction — using natural structural assumptions on the RBM (like ferromagneticity) to give better results for solving supervised prediction tasks in an interesting distributional setting. Along these lines, we show that an assumption related to

ferromagneticity, but allowing for some amount of negative correlation in the RBM, allows us to learn the induced feedforward network faster than would be possible without distributional assumptions (Section 3). Lastly, we present an experimental evaluation of our "supervised RBM" algorithm on MNIST and FashionMNIST to highlight the applicability of our techniques in practice (Section 5).

2 Learning RBMs via New Results for Feedforward Networks

Relationship between RBMs and Feedforward Networks Our first result characterizes the relationship between RBMs and Feedforward networks. We show that there is a natural self-supervised prediction task in RBMs, of predicting the spin at node i given all other observed nodes, for which the Bayes-optimal predictor is *exactly given* by a two-layer feedforward network with a special family of $\tanh$-like activations.

Theorem 1. *For any visible unit i in an arbitrary RBM,*

$$\mathbb{E}[X_i|X_{\sim i}] = \tanh\left(b_i^{(1)} + \sum_j \tanh(W_{ij}) f_{\beta_{ij}}\left(b_j^{(2)} + \sum_{k\neq i} W_{kj}X_k\right)\right) \tag{1}$$

where $\beta_{ij} = |\tanh(W_{ij})|$ and $f_\beta(x) := \frac{1}{\beta}\tanh^{-1}(\beta\tanh(x))$.

Proof. Observe that the conditional distribution of (X_i, H) given $X_{\sim i} = x_{\sim i}$ is given by

$$\Pr(X_i = x_i, H = h|X_{\sim i} = x_{\sim i}) \propto \exp\left(x_i(b_i^{(1)} + \sum_j W_{ij}h_j) + \langle W_{\sim i}^t x_{\sim i} + b^{(2)}, h\rangle\right) \tag{2}$$

where $W_{\sim i}$ denotes the $(n_1 - 1) \times n_2$ dimensional matrix given by deleting row i. Since the only quadratic terms left in the potential are between the remaining visible unit X_i and the hidden units h_j, this conditional distribution is exactly an Ising model on a star graph, i.e. a tree of depth 1 with root node corresponding to X_i. For all tree-structured graphical models, the conditional distribution of the root given the leaves can be computed exactly by Belief Propagation (see e.g. [21, 22]); in the case of Ising models it's known the general BP formula can be written with hyperbolic functions as above[1]. □

Remark 1. *An analogous result can be proved in the more general setting where the spins do not have to be binary; for example in a Potts model version of the RBM where each spin is valued in a set of size q, the conditional law of X_i given the others would be given again by a two-layer network where the last layer is a softmax. In this paper we focus on the binary case for simplicity.*

Remark 2. *The family of activation functions $f_\beta(x)$ naturally interpolates between the identity activation ($\beta = 1$ where $f_\beta(x) = x$) and $\tanh$ activation at $\beta = 0$, since*

$$\lim_{\beta\to 0}\frac{1}{\beta}\tanh^{-1}(\beta\tanh(x)) = \frac{\partial}{\partial\beta}\tanh^{-1}(\beta\tanh(x))\Big|_{\beta=0} = \tanh(x).$$

The exact structure of this prediction function is crucial in what follows and does not seem to have been known in the RBM literature, though some related ideas have been used to develop better heuristics for performing inference and training in RBMs (see discussion in Appendix B).

Given this connection, we show that if we can solve the problem of learning such a neural network within sufficiently small error, then we can successfully learn the RBM. This reduces our RBM learning problem to that of learning feedforward neural networks in the setting that the input is bounded in ℓ_∞ norm.

Improved Results for Learning Feedforward Networks Subsequently, we give results for the feedforward network problem which are nearly optimal both in the terms of sample complexity (in the regime where λ is bounded) and in terms of computational complexity under the hardness of learning sparse parity with noise; some aspects of this result are new even for the well-studied case of learning neural networks with $\tanh$ activations (see Further Discussion).

[1]For the readers convenience, we include a self-contained derivation of (1) from (2) in Appendix B.1.

Theorem 2 (Informal version of Corollary 1)**.** *Suppose that Y is a random variable valued in $\{\pm 1\}$, X is a random vector such that $\|X\|_\infty \leq 1$ almost surely and*

$$\mathbb{E}[Y|X] = \tanh\left(b^{(1)} + \sum_j w_j f_{\beta_j}\left(b_j^{(2)} + \sum_k W_{jk} X_k\right)\right)$$

where $b^{(1)} \in \mathbb{R}$, $\beta_j \in [0,1]$, w is an arbitrary real vector and W is an arbitrary real matrix. Let W_j denote column *j of W and suppose $\|W_j\|_1 \leq \lambda$ for every j and some $\lambda \geq 2$. Then if we run ℓ_1-constrained regression on the degree D monomial feature map $\varphi_D(x) \mapsto \left(\prod_{i\in S} X_i\right)_{|S|\leq D}$ with appropriate ℓ_1 constraint, the result $\hat{w}$ satisfies with high probability*

$$\mathbb{E}[\ell(\hat{w} \cdot \varphi_d(X), Y)] \leq OPT + \epsilon$$

where OPT is the minimum logistic loss for any measurable function of X, as long as the number of samples m satisfies $m = \Omega((|b^{(1)}|^2 \lambda^{O(D)}) \log(2n))$ where $D = O(\lambda \log(\|w\|_1 \lambda/\epsilon))$ and the runtime of the algorithm is $poly(n^D)$.

We also show, under the standard assumption for hardness of learning sparse parity with noise, the following lower bound which shows that the runtime guarantee in our result is close to tight even in the usual setting of $\tanh$ neural networks ($\beta_j = 0$) — it is optimal up to $\log\log$ factors in the exponent in its dependence on ϵ and $\|w\|_1$, and we also show that at least a subexponential dependence (essentially $2^{\sqrt{\lambda}}$) on λ is unavoidable (assuming the dependence on other parameters in the statement is fixed, since there are e.g. trivial algorithms that run in time 2^n).

Theorem 3 (Informal version of Theorem 11)**.** *There exists families of models (one with ϵ a constant, one with $\|w\|_1$ a constant) where a runtime of $n^{\Omega\left(\frac{\log(\|w\|_1/\epsilon)}{\log\log(\|w\|_1/\epsilon)}\right)}$ is needed for any algorithm to achieve ϵ error with high probability, regardless of its sample complexity. Even in the case of $\tanh$ activations ($\beta_j = 0$ for all j), there exists a sequence of models with $\lambda = \Theta(n \log(n))$ and $\|w\|_1 = O(\sqrt{n})$ which requires runtime $n^{\Omega(\sqrt{\lambda/\log^2(\lambda)\log(n)}\log\|w\|_1)}$ to achieve error $\epsilon = 0.01$ with high probability.*

To our knowledge, the fact that $n^{\log(\|w\|_1/\epsilon)/\log\log(\|w\|_1/\epsilon)}$ runtime is required to learn this class even for $\lambda = 1$, and by the above upper bound is tight up to the $\log\log$ term, was not known before even for standard $\tanh$ networks. As far as the dependence on λ, a similar problem was studied in [23] where they proved the dependence cannot be polynomial using the result of [24] for intersection of halfspaces, based on a different assumption, though our lower bound seems to be somewhat stronger in the present context.

In particular the lower bounds on the runtime show that methods like the kernel trick cannot significantly improve the runtime compared to the simple method of writing out the feature map explicitly used in Theorem 2; however, writing out the feature map lets us use ℓ_1 regularization[2] instead of ℓ_2 which can give significant sample complexity advantages (e.g. $O(\log n)$ vs $O(n)$ for the usual sparse linear regression setups).

Structure Learning of RBMs As explained above, our reduction based on Theorem 1 lets us use the above feedforward network learning result to learn the structure of RBMs. By structure learning, we mean learning the *Markov blanket* of the each visible unit in the marginal distribution of the RBM over visible units, i.e. the minimal set of nodes S such that X_i is conditionally independent of all other X_j conditionally on X_S. We will also refer to the Markov blanket as the (two-hop) neighborhood of node i. This is a natural objective as other tasks such as distribution learning are straightforward in sparse models if the Markov blankets are known. As in the previous work on structure learning in other undirected graphical models (e., we will need some kind of quantitative nondegeneracy condition to guarantee nodes in the Markov blanket of node i are information-theoretically discoverable; it is not hard to see (e.g. using the bounds from [26]) that if two nodes are neighbors but their interaction is extremely weak then it becomes impossible to distinguish the model from the same model with the edge removed without a very large number of samples.

[2]Interestingly, recent work [25] has shown in a special case connections between the implicit bias of gradient descent in feedforward networks and ℓ_1 regularization in function space.

In Ising models and in ferromagnetic RBMs, there are simple conditions on the weight matrices which can ensure neighbors are information-theoretically discoverable. In a general RBM, there is no natural way to place constraints on the weights of the RBM to ensure this: the issue is that two nodes X_i and X_j can be independent even though they have two neighboring hidden units with non-negligible edge weights, since the effect of those hidden units can exactly cancel out so that X_i and X_j are independent or indistinguishably close to independent (a number of examples are given in [18]). For this reason, we will instead make the following assumption on the behavior of the model itself instead of on its weight matrix:

Definition 1. *We say that visible nodes i, j are η-nondegenerate two-hop neighbors if*

$$I(X_i; X_j | X_{\sim i,j}) = \mathbb{E}[\ell(\mathbb{E}[X_i | X_{\sim\{i,j\}}], X_i)] - \mathbb{E}[\ell(\mathbb{E}[X_i | X_{\sim i}], X_i)] \geq \eta$$

or if the same inequality holds with i and j interchanged. Here $I(X_i; X_j | X_{\sim i,j})$ is the conditional mutual information *between X_i and X_j conditional on $X_{\sim i,j}$, and the equality follows from Fact 1 in the Appendix and the definition of mutual information in terms of KL [27].*

Information-theoretically, this condition says that nontrivial information is gained about X_i by observing X_j, even after we have already observed $X_{\sim i,j}$. The fact that X_j is in the Markov blanket of node X_i exactly means that this quantity is nonzero. By Pinsker's inequality [27], η-nondegeneracy is also implied by a lower bound on the partial correlation $\mathrm{Cov}(X_i, X_j | X_{\sim i,j})$.

Example 1. *It is not hard to see that Ising models are equivalent to the marginal distribution of RBMs with maximum hidden node degree equal to 2. Consider an Ising model with minimum edge weight α and such that the maximum ℓ_1-norm into every node is upper bounded by λ and the external field is upper bounded by B, then $\eta \geq e^{-O(\lambda+B)}/\alpha$, see e.g. [6].*

Example 2. *In a ferromagnetic RBM with minimum edge weight α and maximum external field B, it can be shown that $\eta \geq e^{-O(\lambda_1+\lambda_2+B)}/\alpha^2$ (see [18, 19]).*

In order for the RBM to be learnable with a reasonable number of samples (since general RBMs can represent arbitrary distributions with full support on the hypercube [18]), we need to assume it has low complexity in the following sense:

Definition 2. *We say that an RBM is (λ_1, λ_2)-bounded if for any i, $\sum_j |\tanh(W_{ij})| + |b_i^{(1)}| \leq \lambda_1$ and the columns of W are bounded in ℓ_1 norm by λ_2.*

Note that λ_1 and λ_2 bound the ℓ_1 norm into the visible and hidden units, respectively. Based on our upper bounds and lower bounds for the learnability of feedforward networks, it should be less surprising that these parameters play a very different role in the computational learnability of RBMs.

Theorem 4 (Informal version of Theorem 12)**.** *Suppose all two-neighbors in a (λ_1, λ_2)-bounded RBM are η-nondegenerate. Given $m = \Omega(\lambda_2^{O(D)} \log(2n))$ i.i.d. samples from the RBM, where $D = O(\lambda_2 \log(\lambda_1 \lambda_2/\eta))$, we can recover its structure with high probability in time $poly(n^D)$.*

Based on this result we also give a result for learning the RBM in TV distance under the same assumption: see Theorem 13: the sample complexity of this method is essentially the above sample complexity plus $n^2(1 - \tanh(\lambda_1))^{-d_2}$ where d_2 is the maximum 2-hop degree; the $poly(n)$ dependence is required as even learning n bernoullis in TV requires $\Omega(n)$ sample complexity. Our algorithm encodes the distribution as a sparse Markov Random Field, but (if desired) this can easily be converted into a sparse RBM using an algorithm in [18]. Therefore we learn the distribution properly, except that the learned RBM typically has more hidden units than the original RBM (i.e. it is overparameterized).

When interpreting these result, it is crucial not to confuse the ℓ_1 norm parameters λ_1, λ_2 of visible and hidden units with the maximum degrees of these units. Typically in Ising models, we should think of the weight of a typical edge as *shrinking* as d grows so that units stay near the sensitive region of their activation and the behavior of the model does not become trivial — this means that λ_1 and λ_2 may be much smaller than d. This is consistent with practical advice in the RBM literature, see e.g. [1]. Probably the most well known sufficient condition for being able to sample in an Ising model (or RBM) is *Dobrushin's uniqueness criterion* which is equivalent to the requirement that $\lambda_1, \lambda_2 \leq 1$ and this condition is actually tight for Glauber dynamics to mix quickly in the Ising model on the complete graph (Curie-Weiss Model) [28]. We discuss this further in Remark 5; in Dobrushin's uniqueness regime and under some mild nondegeneracy conditions we expect that $\eta = \Omega(1/d^2)$

so the above algorithm has runtime $n^{\log(d)}$, which is an exponential improvement in the exponent compared to the best previously known result ($O(n^d)$ runtime by viewing the RBM as an MRF).

We also give lower bound results showing that the computational complexity of the above algorithm is essentially optimal in terms of λ_1 and η (based upon the hardness of learning sparse parity with noise) and nearly optimal in terms of λ_2 for an SQ (Statistical Query) algorithm, in the sense that any SQ algorithm needs at least sub-exponential dependence on λ_2 (given that the dependence on other parameters is not changed — e.g. obviously there is a 2^n time algorithm to learn this problem). In particular, this shows that our results for learning feedforward networks under ℓ_∞ are close to tight even in this application, where the input distribution is related to the label.

Theorem 5 (Informal version of Theorem 19)**.** *As before, λ_2 refers to the maximum ℓ_1-norm into any hidden unit and we choose parameters so that $\lambda_2 = poly(n)$ and $\lambda_1 = poly(n)$. There exists $\epsilon > 0$ so that no SQ algorithm with tolerance $n^{-\lambda_2^\epsilon}$ and access to $n^{\lambda_2^\epsilon}$ queries can structure learn an $\alpha = \Omega(1)$-nondegenerate (λ_1, λ_2)-bounded RBM.*

We also show (Theorem 16) that the η-nondegeneracy condition is required to achieve nontrivial guarantees even if we are only interested in distribution learning (i.e. in TV), assuming the hardness of learning sparse parity with noise.

3 Supervised RBMs

Since in many applications the input data to a classifier is clearly very structured (e.g. images, natural language corpuses, data on networks, etc.), it is interesting to consider the behavior of classification algorithms under structural assumptions on the data. RBMs are one (relatively simple) generative model which can generate interesting structured data. This suggests the idea of learning "supervised RBMs", as proposed by Hinton [1], where we assume the input and label are drawn from an RBM joint distribution, so that predicting the label is a feedforward network by Theorem 1; in this model the label is just a special visible unit in the RBM. Based on the previous discussion about computational lower bounds, we know that assuming the input to a feedforward network comes from the corresponding RBM does not in general make learning easier, but we know that in RBMs there are very natural assumptions we can make to avoid these computational issues. Our final result is of exactly this flavor, showing how we can learn the supervised RBM under a ferromagneticity-related condition faster than is possible if we did not have a distributional assumption.

In order to emphasize the special role of the node which we want to predict, we will adopt a modified notation where the visible unit which we want to learn to predict is labeled Y and all other visible units are still labeled X. More precisely, we model the joint distribution over input features X valued in $\{\pm 1\}^{n_1}$, latent features H valued in $\{\pm 1\}^{n_2}$ and label $Y \in \{\pm 1\}$ as,

$$\Pr[X = x, H = h, Y = y] \propto \exp\left(\langle x, Wh\rangle + \langle h, w\rangle y + \langle b^{(1)}, x\rangle + \langle b^{(2)}, h\rangle + b^{(3)} y\right)$$

where the *weight matrix* W is a non-negative $n_1 \times n_2$ matrix, w is an arbitrary n_1 dimensional vector and $b^{(1)} \in \mathbb{R}^{n_2}$, $b^{(2)} \in \mathbb{R}^{n_2}$ and $b^{(3)} \in \mathbb{R}$ are arbitrary. Given the latent variables H, w can be seen as the linear predictor for Y.

Theorem 6 (Informal Version of Theorem 21)**.** *Suppose the interaction matrix W is ferromagnetic with minimum edge weight α. Further suppose one of the RBMs induced by conditioning on $Y = 1$ or $Y = -1$ is a (λ, λ)-RBM. Then there exists an algorithm that learns the predictor Y that minimizes logistic loss up to error ϵ. The algorithm has sample complexity $m = n_1^2 \exp(\lambda)^{\exp(O(\lambda))}(1/\alpha)^{O(1)} \log(n_1/\delta)/\epsilon^2$ and has runtime $poly(m)$.*

Our main algorithm can be broken down into three main steps: (1) Use greedy maximization of conditional covariance $\mathsf{Cov}^{\mathsf{Avg}}$ to first learn the two-hop neighborhood $\mathcal{N}(i)$ of each observed variable i w.r.t. the hidden layer conditioned on the label (see Algorithm 1), (2) For each observed variable X_i, learn the conditional law of $X_i \mid X_{\mathcal{N}(i)}, Y$ using regression, and (3) Use the estimated distribution to compute $\mathbb{E}[Y|X]$. Step (1) leverages tools from [18, 19] but considers a setting where the RBM may in fact have some amount of negative correlation, as w has arbitrary signs and is allowed to have large norm. Step (2) can be achieved by simply looking at the conditional law under the empirical distribution; this is efficient as we learn small neighborhoods.

In step (3), we can make use of the following useful trick (a version of which can be found in [1]): we already have enough information to derive the law of $Y \mid X$ since we know the marginal law of Y (the fraction of $+$ and $-$ labels) and the law of $X \mid Y$. However, naively carrying out the Bayes law calculation is difficult because it involves partition functions (which are in general NP-hard to approximate, see e.g. [29]). We avoid computing the partition function by observing that if we define f_1, f_2 such that $\Pr(X, Y) \propto \exp(f_1(X)\mathbb{1}(Y = 1) + f_2(X)\mathbb{1}(Y = -1) + by)$, then the law of $Y \mid X$ follows a logistic regression model where

$$\mathbb{E}[Y \mid X] = \tanh\left(\frac{f_1(X) - f_2(X)}{2} + b\right)$$

for some constant $b \in \mathbb{R}$. Therefore if we know f_1, f_2 up to additive constants (which we can derive from the Fourier coefficients learned in (2)), we can simply fit a logistic regression model from data to learn h plus the missing constants, and we can prove this works using fundamental tools from generalization theory. We refer the reader to Appendix E for additional details.

Algorithm 1 LEARNSUPERVISEDRBMNBHD$(u, \tau, \mathcal{S})$ (Adapted from [18, 19])

1: Set $S := \phi$
2: Set $i^* = \arg\max_v \widehat{\mathsf{Cov}}_{\mathcal{S}}^{\mathsf{Avg}}(u, v|S, Y)$, and $\eta^* = \max_v \widehat{\mathsf{Cov}}_{\mathcal{S}}^{\mathsf{Avg}}(u, v|S, Y)$
3: **if** $\eta^* \geq \tau$ **then**
4: $\quad S = S \cup \{i^*\}$
5: **else**
6: $\quad$ Go to Step 8
7: Go to Step 2
8: For each $v \in S$, if $\widehat{\mathsf{Cov}}_{\mathcal{S}}^{\mathsf{Avg}}(u, v|S\backslash\{v\}, Y) < \tau$, remove v (*Pruning step*)
9: Return S

Observe that under the given distributional assumptions, our algorithm has runtime complexity polynomial in the input dimension in contrast to Theorem 2 where the run time scales as $n^{\Omega(\lambda)}$. A simple example which shows the algorithm from this Theorem will outperform any algorithm without distributional assumptions (like Theorem 2) is given in Remark 8.

4 Discussion: Comparison to Prior work on Learning Neural Networks

In the neural network learning literature, various works prove positive results that either (1) work for any distribution with norm assumptions or (2) require strong distributional assumptions. The result of Theorem 2 falls into the category (1) and the result of Theorem 6 falls into category (2).

We first discuss the relation of Theorem 2 to other previous works of type (1). Perhaps the most closely related works are [23, 30, 31, 32]. All of these works assume the input is bounded in ℓ_2 norm and give learning results based on kernel methods; of course, these results could be applied under the assumption of ℓ_∞-bounded input, by using the inequality $\|x\|_2 \leq \sqrt{n}\|x\|_\infty$ and rescaling the input to have norm 1. For comparison, the best result in the ℓ_2 setting with $\tanh$ activation is given in [32], but this result (as is essentially necessary based on the known computational hardness results) has exponential dependence on the ℓ_2 norm of the weights in the hidden units, so doing such a reduction just using norm comparison bounds gives a runtime sub-exponential in dimension. Therefore it is indeed crucial for us to give a new analysis adapting to learning with input bounded in ℓ_∞. An interesting feature of this setting (as mentioned above) is that the kernel trick does not seem to be as useful for improving the runtime as the ℓ_2 setting, where it seems genuinely better than writing out the feature map [31, 32].

Due to the generality of direction (1), it is hard to design efficient algorithms. This further motivates direction (2), however, making the right distributional assumptions which allow for efficient learning while being well-motivated in context of real world data can be very challenging. Most prior work has been limited to the Gaussian input [33, 34, 35, 36, 37, 38] or symmetric input [32, 39] assumptions which are not satisfied by real world data. The works of [40, 41] gave results for some simple tree-structured generative models. There has been some work in defining data based notions such as eigenvalue decay [42] and score function computability [43] to get efficient results. Our assumption

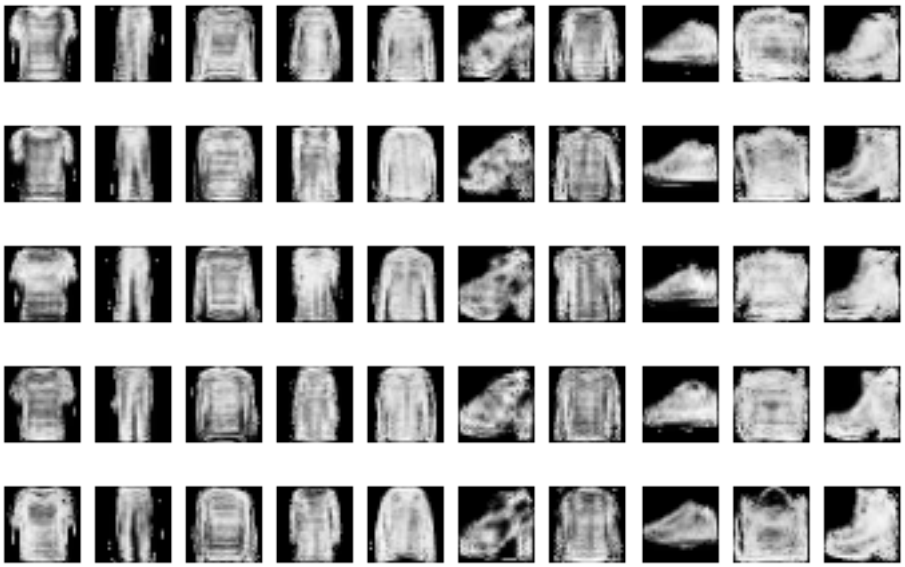

Figure 1: Five i.i.d. samples for each FashionMNIST class, drawn from the trained model by Gibbs sampling.

for Theorem 6 in contrast exploits sparsity and nonnegative correlations among the input features conditional on the output label.

5 Experiments

In this section we present some simple experiments on MNIST and FashionMNIST to confirm that our method performs reasonably well in practice. In these experiments, we implemented the supervised RBM learning algorithm from Theorem 6 which makes use of the classification labels provided in the training data set. This algorithm outputs both a classifier (which predicts the label given the image) and also a generative model (which can sample images given a label).

For classification, we allowed the logistic regression (described as "step (3)" above) to fit not just the bias term but also coefficients on the sum of Fourier coefficients for each pixel (an input of dimension $768 \times 10 = 7680$), since the runtime of the logistic regression step is almost negligible anyway. This is useful because it allows greater dynamic range in the influence of each pixel.

We observed a test accuracy of $97.22 \pm 0.16\%$ on MNIST; the training accuracy was 99.9% and we trained the logistic regression for 30 epochs (same as steps) of L-BFGS with line search enabled. For FashionMNIST, we obtained a test accuracy of $88.84 \pm 0.31\%$; the training accuracy was 92.19% and we trained the logistic regression for 45 epochs with L-BFGS as before. Overall training took a bit less than an hour each on a Kaggle notebook with a P100 GPU. Both datasets have $60,000$ training points and $10,000$ test; in both experiments we used a maximum neighborhood size of 12, and stopped adding neighbors if the conditional variance shrunk by less than 1%.

For context, we note that our accuracy on MNIST is better than what we would get using standard training methods for RBMs and logistic regression for classification; [44] reports accuracies of approximately 95% for CD and 96% using a more sophisticated TAP-based training method. The results are also around as good or better than what is achieved using many classical machine learning methods on these datasets [45]; for example, logistic regression achieves error 91.7% and 84.2% and polynomial kernel SVM achieves error 89.7% and 97.6% [45]. Of course, none of these results are as good as specialized deep convolutional networks (over 99% on MNIST). In contrast to other approaches using linear models such as kernel SVM, our approach also learns a generative model. Being able to sample from the generative model can give some insight into how the model classifies.

To evaluate the performance of the learned RBM as a generative model, we generated samples using Gibbs sampling starting from random initialization and run for 6000 steps. As is common

practice, we output the probabilities generated in the last step instead of the sampled binary values, so that the result is a normal greyscale image. We display the resulting samples in Figures 1 and 2 (for reference, see randomly sampled training datapoints in Appendix F): we note that the model successfully generates samples with diversity, as in Figure 1 the model generates handbags both with and without handles, and in Figure 2 it renders both common styles for drawing the number 4.

It is clear that the model fails to generate as detailed of patterns exhibited in real FashionMNIST images since in our training algorithm, we represent a gray pixel as a random combination of black and white, so a checkerboard pattern of black and white and a patch of grey are not well-distinguished. We do this to ensure that our setup is comparable to classic RBM training [1]. It is potentially possible to fix this by adding spins over larger alphabets (e.g. real-valued) to the model.

Broader Impact

We believe our work will be of most use to other researchers working on sparse graphical models with latent variables. We do not expect our research to disadvantage any individual. As with most machine learning tools, the proposed algorithm for classification could possibly fit to existing biases in the data. In fact, since our algorithm also learns per class distributions, a practitioner can sample from the distribution to further evaluate any biases implicitly modelled. Any practitioner using our method will need to apply the same due diligence as if they were fitting their data using a different method, such as logistic regression.

Funding Information

This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing for the Summer 2019 program on the Foundations of Deep Learning. A substantial part of the work was done while SG was a graduate student at UT Austin.

SG was supported by the JP Morgan AI PhD Fellowship. FK is supported in part by NSF award CCF-1453261 and Ankur Moitra's Packard Foundation Fellowship. AK is supported by NSF awards CCF-1909204 and CCF-1717896.

References

[1] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[2] William Bialek, Andrea Cavagna, Irene Giardina, Thierry Mora, Edmondo Silvestri, Massimiliano Viale, and Aleksandra M Walczak. Statistical mechanics for natural flocks of birds. *Proceedings of the National Academy of Sciences*, 109(13):4786–4791, 2012.

[3] C Chow and Cong Liu. Approximating discrete probability distributions with dependence trees. *IEEE transactions on Information Theory*, 14(3):462–467, 1968.

[4] David Karger, David Karger, and Nathan Srebro. Learning markov networks: Maximum bounded tree-width graphs. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 392–401. Society for Industrial and Applied Mathematics, 2001.

[5] Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356. Springer, 2008.

[6] Guy Bresler. Efficiently learning ising models on arbitrary graphs. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 771–782. ACM, 2015.

[7] Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of ising models. In *Advances in Neural Information Processing Systems*, pages 2595–2603, 2016.

[8] Adam Klivans and Raghu Meka. Learning graphical models using multiplicative weights. In *FOCS*, 2017.

[9] Nicolai Meinshausen, Peter Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34(3):1436–1462, 2006.

[10] Andrej Bogdanov, Elchanan Mossel, and Salil Vadhan. The complexity of distinguishing markov random fields. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 331–342. Springer, 2008.

[11] Joseph Felsenstein. *Inferring phylogenies*, volume 2. Sinauer associates Sunderland, MA, 2004.

[12] Constantinos Daskalakis, Elchanan Mossel, and Sébastien Roch. Optimal phylogenetic reconstruction. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 159–168, 2006.

[13] Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1935–1967, 2012.

[14] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[15] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[16] Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pages 536–543. ACM, 2008.

[17] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223, 2011.

[18] Guy Bresler, Frederic Koehler, and Ankur Moitra. Learning restricted boltzmann machines via influence maximization. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 828–839, 2019.

[19] Surbhi Goel. Learning ising and potts models with latent variables. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3557–3566, Online, 26–28 Aug 2020. PMLR.

[20] Gregory Valiant. Finding correlations in subquadratic time, with applications to learning parities and juntas. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 11–20. IEEE, 2012.

[21] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[22] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[23] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM Journal on Computing*, 40(6):1623–1646, 2011.

[24] Adam R Klivans and Alexander A Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

[25] Blake Woodworth, Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Kernel and deep regimes in overparametrized models. *arXiv preprint arXiv:1906.05827*, 2019.

[26] Narayana P Santhanam and Martin J Wainwright. Information-theoretic limits of selecting binary graphical models in high dimensions. *IEEE Transactions on Information Theory*, 58(7):4117–4134, 2012.

[27] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[28] David A Levin and Yuval Peres. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017.

[29] Allan Sly and Nike Sun. The computational hardness of counting in two-spin models on d-regular graphs. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 361–369. IEEE, 2012.

[30] Yuchen Zhang, Jason D Lee, and Michael I Jordan. ℓ_1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning*, pages 993–1001, 2016.

[31] Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. In *Conference on Learning Theory*, pages 1004–1042, 2017.

[32] Surbhi Goel, Adam R. Klivans, and Raghu Meka. Learning one convolutional layer with overlapping patches. In Jennifer G. Dy and Andreas Krause 0001, editors, *ICML*, volume 80 of *JMLR Workshop and Conference Proceedings*, pages 1778–1786. JMLR.org, 2018.

[33] Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3404–3413. JMLR. org, 2017.

[34] Mahdi Soltanolkotabi. Learning relus via gradient descent. In *Advances in Neural Information Processing Systems*, pages 2007–2017, 2017.

[35] Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 605–614. JMLR. org, 2017.

[36] Kai Zhong, Zhao Song, Prateek Jain, Peter L Bartlett, and Inderjit S Dhillon. Recovery guarantees for one-hidden-layer neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 4140–4149. JMLR. org, 2017.

[37] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in neural information processing systems*, pages 597–607, 2017.

[38] Simon Du, Jason Lee, Yuandong Tian, Aarti Singh, and Barnabas Poczos. Gradient descent learns one-hidden-layer cnn: Don't be afraid of spurious local minima. In *International Conference on Machine Learning*, pages 1339–1348, 2018.

[39] Rong Ge, Rohith Kuditipudi, Zhize Li, and Xiang Wang. Learning two-layer neural networks with symmetric inputs. In *International Conference on Learning Representations*, 2019.

[40] Elchanan Mossel. Deep learning and hierarchal generative models. *arXiv preprint arXiv:1612.09057*, 2016.

[41] Eran Malach and Shai Shalev-Shwartz. A provably correct algorithm for deep learning that actually works. *arXiv preprint arXiv:1803.09522*, 2018.

[42] Surbhi Goel and Adam Klivans. Eigenvalue decay implies polynomial-time learnability for neural networks. In *Advances in Neural Information Processing Systems*, pages 2192–2202, 2017.

[43] Weihao Gao, Ashok V Makkuva, Sewoong Oh, and Pramod Viswanath. Learning one-hidden-layer neural networks under general input distributions. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1950–1959, 2019.

[44] Marylou Gabrié, Eric W Tramel, and Florent Krzakala. Training restricted boltzmann machine via the thouless-anderson-palmer free energy. In *Advances in neural information processing systems*, pages 640–648, 2015.

[45] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[46] Alexander A Sherstov. Making polynomials robust to noise. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 747–758, 2012.

[47] András Hajnal, Wolfgang Maass, Pavel Pudlák, Mario Szegedy, and György Turán. Threshold circuits of bounded depth. *Journal of Computer and System Sciences*, 46(2):129–154, 1993.

[48] James Martens, Arkadev Chattopadhya, Toni Pitassi, and Richard Zemel. On the representational efficiency of restricted boltzmann machines. In *Advances in Neural Information Processing Systems*, pages 2877–2885, 2013.

[49] Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.

[50] Leslie Ann Goldberg and Mark Jerrum. The complexity of ferromagnetic ising with local fields. *Combinatorics, Probability and Computing*, 16(1):43–61, 2007.

[51] Max Welling and Yee Whye Teh. Approximate inference in boltzmann machines. *Artificial Intelligence*, 143(1):19–50, 2003.

[52] Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

[53] Ronald A DeVore and George G Lorentz. *Constructive approximation*, volume 303. Springer Science & Business Media, 1993.

[54] Elias M Stein and Rami Shakarchi. *Complex analysis*, volume 2. Princeton University Press, 2010.

[55] Frederic Koehler and Andrej Risteski. The comparative power of relu networks and polynomial kernels in the presence of sparse latent structure. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2019.

[56] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[57] Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

[58] Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *Journal of Statistical Physics*, 155(4):666–686, 2014.

[59] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic ising and hard-core models. *Combinatorics, Probability and Computing*, 25(4):500–559, 2016.

[60] Peter L Bartlett, Olivier Bousquet, Shahar Mendelson, et al. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.

[61] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

A Outline of the Appendix

Here we briefly outline the contents of each remaining section; each bold heading in the text below corresponds to a new section.

Appendix B. Connections between Distribution Learning and Prediction in RBMs In this section we show that if you have learned the distribution of an RBM, then you have also in principle learned how to predict the output of corresponding feedforward networks. These feedforward networks are induced from a "self-supervised" prediction task: predicting the spin at node i given observations of all other spins. This connection leverages a classical observation in probabilistic inference: inference in all tree-structured graphical models has an exact solution known as Belief Propagation (see e.g. [22, 21]); perhaps surprisingly, this observation is useful even though the RBM itself is not tree structured. Conversely, in the next subsection we give quantitative bounds showing that sufficiently good predictors for this self-supervised objective for every node i allows us to recover the distribution of the corresponding RBM.

Appendix C. Guarantees for Learning Feedforward Networks (with arbitrary distribution). In this section we prove upper and lower bounds for learning one-layer feedforward networks with f_β activations in the hidden units and inputs X drawn from an arbitrary distribution such that $\|X\|_\infty \leq 1$.

In the first two subsections, we prove the needed approximation-theoretic results about our class of activations f_β, giving approximation results with uniform guarantees over the entire interval $\beta \in [0, 1]$. In the special case of $\beta = 0$, $f_\beta = \tanh$ and the needed result has essentially already been proved in the work of [23]. As explained in the first subsection, by a classical result of Bernstein (Theorem 7 below) it turns out that analyzing approximation theory for functions analytic on $[-1, 1]$ is equivalent to analyzing the function's extension into the complex plane. We develop the needed complex-analytic estimates (which crucially are uniform in β) in the following subsection. We note that the authors of [23] did not use Bernstein's result to prove their bound; their analysis of the $\beta = 0$ case is longer because they more or less reproduce the steps from the proof of the upper bound of Bernstein's Theorem.

After solving the approximation-theoretic question, we use them in an ℓ_1-regression based algorithm for learning feedforward networks, using an explicit polynomial feature map and the logistic version of the Lasso with its corresponding nonparametric generalization bounds. We derive the needed ℓ_1-norm bound in a clean way from the approximation-theoretic results using in part a Lemma of [46], previously used in [31]. This proves Theorem 2. In the last subsection, we prove that this result is nearly optimal under the hardness of sparse parity with noise, even in the case of $\tanh$ networks, using two different ways to construct a parity out of $\tanh$ units: one is a well-known construction from [47], the other is based on Taylor series expansion and is related to the MRF-to-RBM embedding result established in [18].

Appendix D. Learning RBMs by Learning Feedforward Networks. In this section, we show how to derive structure recovery results (i.e. recovery of Markov blankets) for RBMs by using the feedforward network learning results developed in the previous section. Assuming η-nondegeneracy, we show how to learn the structure of the network by doing simple regression tests, e.g. comparing the minimal logistic loss achieved predicting node i from all other nodes to the loss when node j is excluded from the input. This proves Theorem 4. We explain in more detail in Remark 5 how this result is a significant improvement over previous results in interesting regimes where we know that the RBM can actually be sampled from in polynomial time. Based on this, we prove a result for learning the distribution: by Theorem 4 this reduces to the case where the structure is known, so by proving a good estimate (Lemma 12) on the convergence of the natural predictor of X_i given its neighbors, the empirical conditional expectation and using the tools developed in Section B.3 gives the result. A key point here is that the empirical conditional expectation converges at a much faster rate than e.g. relying on Theorem 10, which gives better sample complexity guarantees.

Finally, we again prove some computational hardness results. We establish that the algorithm's dependence is essentially optimal in terms of η and $\|w\|_1$ by using the Taylor-series based sparse parity construction from [18], related to the construction used above for $\tanh$ networks. For the dependence on λ_2, the hidden unit ℓ_1-norm, we use a third, different construction of parity from [48]

for the RBM setting; this construction is not amenable to adding noise, but we are able to prove a lower bound on the runtime in terms of λ_2 for all SQ (Statistical Query) algorithms (see e.g. [49]).

Appendix E. Learning a Feedforward Network by Learning RBMs. In this section, we prove Theorem 6, which lets us learn to predict in supervised RBMs under a natural conditional ferromagneticity condition in a provably more computationally efficient way than applying distribution-agnostic methods for learning feedforward networks like Theorem 2. In Remark 8 we give a simple example where the gap is provable and explain the (in this case) simple intuition as to how the approach of Theorem 6 uses the structure of the input data in a favorable way.

The idea of this learning algorithm is essentially to use Bayes rule to reduce computing the posterior on the label (i.e. $\Pr(Y|X)$) to computing the conditional likelihood of the observed X under the two possible values of the label. In some situations where the conditional law of $Y|X$ is very simple, this approach may be overkill as it requires to model the law of X; however, we are interested in the setting where the label Y may have a large, complicated effect on X so this approach seems perfectly reasonable. An obvious issue with using Bayes rule in this way is that even if the the RBM is already known perfectly, computing the normalizing constant for the conditional distribution under $Y = +$ or $Y = -$ in such a model is #BIS-Hard [50]. Fortunately, for our application we show that we can estimate the needed ratio of normalizing constants from the data using a simple variant of logistic regression.

What remains is to learn how to estimate the conditional log-likelihoods i.e. $\Pr(X|Y)$. Fortunately, even though under our assumptions the original RBM was not ferromagnetic, the conditional models we get by applying Bayes rule are indeed ferromagnetic so we can apply the methods developed in [19] for learning such a model. Here we need the results of [19] and not the earlier work of [18] as we expect the external fields in the resulting model to be inconsistent (have differing signs depending on the site). Once the structure is recovered, we can learn the coefficients of the log-likelihood using the results established in the previous section based on fast convergence of the empirical condition expectation, and using these coefficients we can accurately estimate $\Pr(X|Y)$ for the application of Bayes rule.

Appendix F. Additional Experimental Data. In this section we include reference images from both datasets along with samples generated by our algorithm trained on MNIST.

B Connections between Distribution Learning and Prediction in RBMs

To our knowledge, Theorem 1 has not been previously noted in the literature on RBMs. However, this is not the first time connections between RBMs and message passing algorithms for inference has been investigated: for example, the work of [51] extensively studied the use of message passing algorithms (i.e. Belief Propagation and related algorithms) for estimating the mean and covariance matrix of nodes in an RBM, and the work of [44] used the related TAP approximation to derive better alternatives to constrastive divergence for training RBMs in practice. The key conceptual difference is that in these works, their goal is to solve a much harder problem (e.g. estimating marginals and $\log Z$) which is well-known to be NP-hard in general. In contrast, for our application to learning the relevant task ends up being predicting one node from the others, which it turns out is *not* computationally difficult if we know the model — conditioning on the other nodes breaks all cycles in the graph, which is the obstacle that makes inference difficult in general.

B.1 Conditional Law Derivation

In this Appendix we give, for the reader's convenience, a self-contained derivation of the conditional law (1) described in Theorem 1 for $\mathbb{E}[X_i|X_{\sim i}]$ from (2). As described in the proof of the Theorem, the result is obtained as a special case of the Belief Propagation algorithm as described in a number of references, including [21, 22], which is derived by performing a more general version of this calculation. First recall that the joint conditional law on X_i, H condiditioned on $X_{\sim i}$ is given by (2):

$$\Pr(X_i = x_i, H = h|X_{\sim i} = x_{\sim i}) \propto \exp\left(x_i(b_i^{(1)} + \sum_j W_{ij}h_j) + \langle W_{\sim i}^t x_{\sim i} + b^{(2)}, h\rangle\right).$$

The computation proceeds by rewriting this measure with respect to a "cavity" measure where all terms involving X_i are removed. For each hidden unit j, define a corresponding probability measure

$$\mu_{H_j \to X_i}(h_j) \propto \exp\left(\sum_{k \neq i} W_{kj} x_j h_j + b_j^{(2)} h_j\right)$$

under which $\sum_j h_j \mu_{H_j \to X_i}(h_j) = \tanh(\sum_k W_{kj} x_j + b_j^{(2)})$ and rewrite the joint probability over X, H as

$$\Pr(X_i = x, H = h | X_{\sim i} = x_{\sim i}) \propto \exp\left(x_i (b_i^{(1)} + \sum_j W_{ij} h_j)\right) \prod_j \mu_{H_j \to X_i}(h_j).$$

Now we compute that

$$\begin{aligned}
&\Pr[X_i = x_i | X_{\sim i} = x_{\sim i}] \\
&= \sum_h x_i \Pr(X_i = x_i, H = h | X_{\sim i} = x_{\sim i}) \\
&\propto \sum_h \exp\left(x_i (b_i^{(1)} + \sum_j W_{ij} h_j)\right) \mu_{H \to X_i}(h) \\
&= \exp(x_i b_i^{(1)}) \prod_{j=1}^{n_2} (\cosh(W_{ij}) + \sinh(x_i W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)})) \\
&\propto \exp(x_i b_i^{(1)}) \prod_{j=1}^{n_2} (1 + x_i \tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)})) \\
&= \exp\left(x_i b_i^{(1)} + \sum_{j=1}^{n_2} \log(1 + x_i \tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)}))\right)
\end{aligned}$$

where we used $\propto$ to ignore constants of proportionality independent of x_i and in the third line we used Lemma 1 below. Therefore if we use that

$$\log(1+\beta x_i) = \frac{1}{2} \log \frac{1+\beta x_i}{1-\beta x_i} + \frac{1}{2}(\log(1+\beta x_i) + \log(1-\beta x_i)) = \tanh^{-1}(\beta x_i) + \frac{1}{2}(\log(1+\beta) + \log(1-\beta))$$

where we see the last term does not depend on x, we can compute that

$$\begin{aligned}
&\mathbb{E}[X_i = x_i | X_{\sim i} = x_{\sim i}] \\
&= \frac{\sum_{x_i} x_i \exp\left(x_i b_i^{(1)} + \sum_{j=1}^{n_2} \log(1 + x_i \tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)}))\right)}{\sum_{x_i} \exp\left(x_i b_i^{(1)} + \sum_{j=1}^{n_2} \log(1 + x_i \tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)}))\right)} \\
&= \frac{\sum_{x_i} x_i \exp\left(x_i b_i^{(1)} + \sum_{j=1}^{n_2} x_i \tanh^{-1}(\tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)}))\right)}{\sum_{x_i} \exp\left(x_i b_i^{(1)} + \sum_{j=1}^{n_2} x_i \tanh^{-1}(\tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)}))\right)} \\
&= \tanh\left(b_i^{(1)} + \sum_{j=1}^{n_2} \tanh^{-1}(\tanh(W_{ij}) \tanh(\sum_{k \neq i} W_{kj} x_j + b_j^{(2)}))\right)
\end{aligned}$$

where in the final step we used that $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$. From this we get (1) by plugging in the definition of $f_{\beta_{ij}}$.

Lemma 1. *For any $z \in \mathbb{R}$ we have the formula for moment generating function of a recentered Bernoulli:*

$$\mathbb{E}_{X \sim Ber_\pm(\tanh(z))}[\exp(\lambda X)] = \cosh(\lambda) + \sinh(\lambda) \tanh(z)$$

where $Ber_\pm(\mu)$ denotes the distribution of a $\{\pm 1\}$-valued random variable with mean μ.

Proof. First recall that $\mathbb{E}_{X\sim Rad}[\exp(\lambda X)] = \cosh(\lambda)$ and $\mathbb{E}_{X\sim Rad}[X\exp(\lambda X)] = \tanh(\lambda)$. Therefore

$$\begin{aligned}\mathbb{E}_{X\sim Ber_{\pm}(\tanh(z))}[\exp(\lambda X)] &= \mathbb{E}_{X\sim Rad}\left[e^{\lambda X}\frac{e^{zX}}{\cosh(z)}\right]\\ &= \frac{\cosh(z+\lambda)}{\cosh(z)}\\ &= \frac{\cosh(z)\cosh(\lambda)+\sinh(z)\sinh(\lambda)}{\cosh(z)}\\ &= \cosh(\lambda)+\sinh(\lambda)\tanh(z).\end{aligned}$$

□

B.2 2-layer Tanh Neural Network as Bayes-Optimal Prediction in an RBM

In particular, (1) lets us realize *any* standard 2-layer $\tanh$ neural network as the Bayes-optimal predictor in an RBM in a natural limit where the number of hidden neurons goes to infinity, but the effect of each hidden neuron is very small, so that the ℓ_1 norm of the weights going into the top neuron stays bounded by a constant. Each hidden unit in the neural network corresponds in a direct way to several duplicated hidden units in the RBM. The construction is given explicitly in the next Lemma; we will not use the statement explicitly but use it to develop intuition for (1).

Lemma 2. *Suppose that* $g(x) = \tanh\left(u_0 + \sum_{j=1}^{T} u_j \tanh\left(M_{j0} + \sum_k M_{jk}x_k\right)\right)$ *where* x *is* n*-dimensional, i.e.* g *is a 2-layer neural network with* $\tanh$ *activations. Then*

$$g(x) = \lim_{K\to\infty} \tanh\left(u_0 + \sum_{i=1}^{K}\sum_{j=1}^{T} \tanh(u_j/K) f_{|u_j/K|}\left(M_{j0} + \sum_k M_{jk}x_k\right)\right),$$

so by (1) *from Theorem 1 the restriction of* f *to* $\{\pm 1\}^n$ *is the Bayes-optimal predictor of a visible unit in an RBM with* $n+1$ *total visible units where the activations of the other visible units are known.*

Proof. This follows from the observation in Remark 2 and from Theorem 1 by building the corresponding RBM with KT hidden units. □

B.3 Distribution learning bounds from prediction bounds

In this section, we show how good estimates of the conditional prediction functions can be used in a direct way to recover the joint distribution of the RBM in total variation distance.

Algorithm 2 DISTRIBUTIONFROMPREDICTORS

1: For every i we suppose we are given $\hat{f}_i : \{\pm 1\}^n \to \mathbb{R}$ and set $\widehat{\mathcal{N}}(i)$ such that $\hat{f}_i$ is a predictor of node i from other nodes that depends only on those in the set $\widehat{\mathcal{N}}(i)$
2: Define $\mathcal{S} := \{S : \exists i, S \subset \widehat{\mathcal{N}}(i)\}$
3: **for** $S \in \mathcal{S}$ **do**
4: For all $i \in S$, define $\hat{w}_{S,i} := \mathbb{E}_{X\sim Uni(\{\pm 1\}^n)}[\tanh^{-1}(\hat{f}_i(X))X_{S\setminus i}]$.
5: Define $\hat{w}_S := \frac{1}{|S|}\sum_{i\in S}\hat{w}_{S,i}$.
6: Return the MRF with unnormalized pmf $\exp\left(\sum_{S\in\mathcal{S}}\hat{w}_S X_S\right)$.

Lemma 3 ([26])**.** *Suppose* P, Q *are distributions over random variable* X *valued in* $\{\pm 1\}^n$*. If* $P(x) \propto \exp(\sum_S p_S X_S)$ *and* $Q(x) \propto \exp(\sum_S q_S X_S)$ *then*

$$\mathbf{SKL}(P,Q) = \sum_S (p_S - q_S)(\mathbb{E}_P[X_S] - \mathbb{E}_Q[X_S]).$$

where $\mathbf{SKL}(P,Q) = \mathbf{KL}(P,Q) + \mathbf{KL}(Q,P)$ *is the symmetrized KL divergence.*

Proof. From the definition we see

$$\mathbf{SKL}(P,Q) = \mathbb{E}_P\left[\log\frac{P(x)}{Q(x)}\right] - \mathbb{E}_Q\left[\log\frac{P(x)}{Q(x)}\right] = \mathbb{E}_P\left[\sum_S (p_S - q_S)X_S\right] - \mathbb{E}_Q\left[\sum_S (p_S - q_S)X_S\right]$$

so using linearity of expectation proves the result. □

The following definition captures the level of contiguity P has with the uniform measure when looking at small sets of coordinates.

Definition 3. *For any distribution P on $\{\pm 1\}^n$ and $d \le n$ we define*

$$\delta_P(d) := \inf_{|S|\le d}\inf_{x_S} 2^{|S|}P(X_S = x_S).$$

Lemma 4. *For any function f which depends on at most d coordinates,*

$$\mathbb{E}_P[f(X)^2] \ge \delta_P(d)\mathbb{E}_{X\sim\{\pm1\}^n}[f(X)^2]$$

The following Lemma is a standard observation used in most previous works on learning Ising models including [6, 7, 8] and others.

Lemma 5. *A (λ_1,λ_2)-bounded RBM satisfies $\delta_P(d) \ge (1-\tanh(\lambda_1))^d$.*

Proof. In the $d = 1$ case this follows from the law of total expectation as $\mathbb{E}[X_i|H, X_{\sim i}] = \tanh(b_i^{(1)} + \sum_j W_{ij}H_j)$ and the term inside the $\tanh$ has magnitude at most λ_1 by definition. For general d the result follows by induction, by using the above argument for a single spin and then applying the induction hypothesis to the model where than spin is plus and where that spin is minus, since these models are also (λ_1,λ_2)-bounded RBMs. □

Lemma 6. *Let $\hat{P}$ denote the distribution returned by Algorithm* DISTRIBUTIONFROMPREDICTORS *and let P be the true distribution. Let $\log P(x) = \sum_S w_S x_S$ and $\log\hat{P}(x) = \sum_S \hat{w}_S x_S$ be the Fourier expansions of the log-likelihoods. Then*

$$\begin{aligned}\mathbf{SKL}(\hat{P},P) &\le \sum_S |w_S - \hat{w}_S| \\ &\le \sum_i \frac{2^{|\mathcal{N}(i)|/2+1}}{\sqrt{\delta_P(|\mathcal{N}(i)\cup\widehat{\mathcal{N}}(i)|)}}\sqrt{\mathbb{E}_{X'}[(\tanh^{-1}(\hat{f}_i(X')) - \tanh^{-1}(\mathbb{E}_P[X_i|X_{\sim i}]))^2]}\end{aligned}$$

where $X' \sim Uni(\{\pm1\}^n)$.

Proof. Define w_S to be the true coefficient in the true MRF potential. By Lemma 3 and Holder's inequality we know $\mathbf{SKL}(P,\hat{P}) \le 2\sum_S |\hat{w}_S - w_S|$. Then by Jensen's inequality and the Cauchy-Schwarz inequality,

$$\begin{aligned}\sum_S |\hat{w}_S - w_S| &\le \sum_S \frac{1}{|S|}\sum_{i\in S}|\hat{w}_{S,i} - w_S| \\ &= \sum_i \sum_{S:i\in S}\frac{1}{|S|}|\hat{w}_{S,i} - w_S| \\ &\le \sum_i 2^{|\mathcal{N}(i)|/2}\sqrt{\sum_{S:i\in S}(\hat{w}_{S,i} - w_S)^2}.\end{aligned}$$

Now using Plancherel's theorem [52], the fact that $f_i(x) = \tanh\left(\sum_{S:i\in S} w_S x_{S\setminus\{i\}}\right)$, and the definition of $\delta_P(d)$ gives the result. □

C Guarantees for Learning Feedforward Networks (with Arbitrary Distribution)

In this section we prove upper and lower bounds for learning one-layer feedforward networks with f_β activations in the hidden units and inputs X drawn from an arbitrary distribution such that $\|X\|_\infty \le 1$.

C.1 Preliminaries: Optimal Approximation of Analytic Functions

Identify $\mathbb{C}$ with $\mathbb{R}^2$ by taking x to be real and y to be the imaginary component of a complex number z. Define $\mathcal{E}_\rho$ to be the region bounded by the ellipse in $\mathbb{C} = \mathbb{R}^2$ centered at the origin with equation $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ with semi-axes $a = \frac{1}{2}(\rho + \rho^{-1})$ and $b = \frac{1}{2}|\rho - \rho^{-1}|$; the focii of the ellipse are ± 1. In the present context, this is sometimes referred to as a *Bernstein ellipse*. For an arbitrary function $f : [-1, 1] \to \mathbb{R}$, let $E_D(f)$ denote the error of the best polynomial approximation of degree D in infinity norm on the interval $[-1, 1]$ of f, i.e.

$$E_D(f) := \min_{P:\deg(P)\leq D} \max_{x\in[-1,1]} |f(x) - P(x)|. \tag{3}$$

The following theorem of Bernstein exactly characterizes the asymptotic rate at which $E_D(f)$ shrinks:

Theorem 7 (Theorem 7.8.1, [53])**.** *Let f be a function defined on $[-1, 1]$. Let ρ_0 be the supremum of all ρ such that f has an analytic extension on the interior of $\mathcal{E}_\rho$. Then*

$$\limsup_{D\to\infty} \sqrt[D]{E_D(f)} = \frac{1}{\rho_0}$$

where we interpret the rhs as ∞ when $\rho_0 = 0$.

For the definition of what it means for the function to be analytic on a region of the complex plane, we refer to a text on complex analysis such as [54]. For our application we need only the upper bound and we need a quantitative estimate for finite degree d. In the proof of the upper bound in [53], the following result is proved:

Theorem 8 (Quantitative Variant of Theorem 7.8.1, [53])**.** *Suppose f is analytic on the interior of $\mathcal{E}_{\rho_1}$ and $|f(z)| \leq M$ on the closure of $\mathcal{E}_{\rho_1}$. Then*

$$E_D(f) \leq \frac{2M}{\rho_1 - 1}\rho_1^{-D}.$$

This quantitative variant was previously used in [55] as part of a construction of low-degree approximations to the ReLU activation with specific properties. Note that when applying this theorem, we should center f so that the constant M is small, since adding constants to f will obviously not change $E_d(f)$.

C.2 Approximation Guarantees for f_β Family of Activations

Recall that the activations f_β were defined in Theorem 1 to be $f_\beta(x) = \frac{1}{\beta}\tanh^{-1}(\beta\tanh(x))$. Recall that if $\beta = 1$ then $f_\beta(x) = x$ so the function is analytic everywhere on $\mathbb{C}$, and if $\beta = 0$ is is $\tanh$ so it is meromorphic. For the remaining values of $\beta \in (0, 1)$, the function f_β is slightly more complicated (it has branch cuts), however we show it is still nicely behaved near the real line.

Lemma 7. *For $\beta \in [0, 1]$ the function f_β is analytic on the strip $\{x + iy : |y| < \pi/2\}$.*

Proof. Observe that

$$f'_\beta(z) = \frac{1 - \tanh^2(z)}{1 - \beta^2\tanh^2(z)}..$$

Since $\tanh$ is analytic except at points of the form $z = \frac{\pi}{2}i + \pi ki$, the only other possible poles are solutions to $\beta^2\tanh^2(z) = 1$, i.e. solutions to $\tanh(z) = \pm 1/\beta$. Recalling that $\tanh^{-1}(z) = \frac{1}{2}(\log(1 + z) - \log(1 - z))$ and taking into account the branch cut from $(-\infty, 0]$ for the logarithm, we see that the solutions to $\tanh(z) = 1/\beta$ are of the form

$$z = \frac{1}{2}\log\frac{1 + 1/\beta}{1/\beta - 1} + \frac{\pi i}{2} + k\pi i$$

and for $\tanh(z) = -1/\beta$ of the form

$$z = \frac{1}{2}\log\frac{1/\beta - 1}{1 + 1/\beta} + \frac{\pi i}{2} + k\pi i$$

for $k \in \mathbb{Z}$. In particular we see that f'_β is analytic on the strip $\{x + iy : |y| < \pi/2\}$ so f_β is as well (since the region is simply connected, this can be proved by path integration [54]). □

To get a quantitative upper bound we will need to bound (the centered version of) f_β on the Bernstein ellipse, which will require us to back away from the singularities of f'_β on the lines $y = \pm\pi/2$. The following Lemma proves that f'_β is uniformly bounded in a slightly smaller region:

Lemma 8. *For all $\beta \in [0,1]$, $|f'_\beta(z)| \le 2$ everywhere on the closed strip $\{x + iy : |y| \le \pi/4\}$.*

Proof. Observe that

$$f'_\beta(z) = \frac{1-\tanh^2(z)}{1-\beta^2\tanh^2(z)} = \frac{\cosh^2(z)-\sinh^2(z)}{\cosh^2(z)-\beta^2\sinh^2(z)}$$
$$= \frac{1}{1+(1-\beta^2)\sinh^2(z)} = \frac{1}{1+(1-\beta^2)\frac{\cosh(2z)-1}{2}}$$

using the identies $\cosh^2(x) - \sinh^2(x) = 1$ and $\sinh^2(z) = \frac{\cosh(2z)-1}{2}$. Since $\cosh(2x+2iy) = \frac{e^{2x+2iy}+e^{-2x-2iy}}{2}$ we see that under the assumption $|y| \le \pi/4$ that $\cosh(2x+2iy)$ lies in the right half plane, therefore $|1+(1-\beta^2)\frac{\cosh(2z)-1}{2}| \ge |1-(1-\beta^2)/2| \ge 1/2$ which proves the result. □

Lemma 9. *For any $\beta \in [0,1]$, arbitrary $h \in \mathbb{R}$, and any $R \ge 0$,*

$$E_D(f_\beta(Rx+h)) \le \frac{4R(1+2R)}{(1+1/2R)^D}$$

Proof. Just for this proof define $g_{\beta,h}(x) := f_\beta(Rx+h) - f_\beta(h)$. We prove this bound by application of Bernstein's theorem. By Lemma 7 we know that f_β is analytic on the strip $\{x+iy : |y| < \pi/2\}$ so in particular it is analytic on the closed strip $\{x+iy : |y| \le \pi/4\}$, and by Lemma 8 we know that $|f'_\beta| \le 2$ on the closed strip.

We now compute ρ so that $R\mathcal{E}_\rho$ is contained in the latter strip. We solve

$$\frac{1}{2}(\rho - \rho^{-1}) = \frac{\pi}{4R}$$

which gives $\rho^2 - \frac{\pi}{2R}\rho - 1 = 0$ so $\rho = \frac{\pi/2R + \sqrt{\pi^2/4R^2+4}}{2} > 1 + 1/2R$. Since $|g'_{\beta,h}(z)| \le R|f'_\beta| \le 2R$ on the closure of the ellipse, it follows by the mean-value theorem that $|g_{\beta,h}| \le 2(1+1/2R)R \le 1+2R$ on $\mathcal{E}_{1+1/2R}$ and applying Theorem 8 gives the result. □

C.3 Learning Feedforward Networks under ℓ_∞ Bounded Input

Since the final activation in our network is $\tanh$, we recall some useful facts about logistic regression and the logistic loss which we will use.

Definition 4. *The* logistic loss *is defined to be*

$$\ell(v,y) := \log(1 + e^{-2vy}).$$

We note that the factor of 2 in the exponent and the normalization differ depending on convention.

The following facts about the logistic loss which can be checked from the definition (or see a reference such as [56]):

Fact 1. *The following are true if $y \in \{\pm 1\}$ is fixed:*

1. *$\ell(v,y)$ is convex and 2-Lipschitz in v.*
2. *$\ell(v,y) = -\log \Pr(\hat{Y} = y)$ where $\hat{Y}$ is a $\{\pm 1\}$-valued random variable with expectation $\tanh(v)$.*
3. *$\frac{\partial}{\partial v}\ell(v,y) = \frac{-2ye^{-2vy}}{1+e^{-2vy}}$ and $\frac{\partial^2}{\partial v^2}\ell(v,y) = \frac{2}{1+\cosh(2v)}$.*

Furthermore if Y is a $\{\pm 1\}$-valued random variable (and v is deterministic) then

4. $\mathbb{E}_Y \ell(v, Y) = \mathbf{KL}(\mathcal{L}(Y), \mathcal{L}(\hat{Y})) + H(Y)$ *where* $\hat{Y}$ *is defined above,* $\mathcal{L}(Y)$ *denotes the law of random variable* Y*,* $\mathbf{KL}$ *denotes the Kullback-Liebler divergence and* H *denotes the Shannon entropy.*

We recall the following Theorem which states the agnostic learning guarantee for fitting ℓ_1-constrained predictors in logistic loss, i.e. the logistic version of the Lasso:

Theorem 9 (Theorem 26.15 of [56])**.** *Suppose that* X *is a random vector in* $\mathbb{R}^n$ *such that* $\|X\|_\infty \leq 1$ *almost surely and* Y *is an arbitrary* $\{\pm 1\}$*-valued random variable. Then with probability at least* $1 - \delta$*, simultaneously for all* w *with* $\|w\|_1 \leq R$ *it holds that*

$$\hat{\mathbb{E}}[\ell(w \cdot X, Y)] \leq \mathbb{E}[\ell(w \cdot X, Y)] + 4R\sqrt{\frac{2\log(2n)}{m}} + 2R\sqrt{\frac{2\log(2/\delta)}{m}}$$

where $\hat{\mathbb{E}}$ *denotes the empirical expectation over* m *i.i.d. copies* $(X_1, Y_1), \ldots, (X_m, Y_m)$ *of* (X, Y)*.*

In order to bound the ℓ_1 norm of our predictor we will need the following Lemmas:

Lemma 10 ([46], Lemma 2.13 of [31])**.** *Suppose* $p(x) = \sum_{i=0}^D \beta_i x$ *and* $|p(x)| \leq M$ *for* $x \in [-1, 1]$*, then* $\sum_{i=0}^D \beta_i^2 \leq (D+1)(4e)^{2D} M^2$*.*

Lemma 11. *Suppose that* $p(x) = \sum_{i=0}^D a_i (w \cdot x)^i = \sum_\alpha u_\alpha x^\alpha$*. Then*

$$\sum_\alpha |u_\alpha| \leq \sqrt{\sum_i a_i^2}(1 + \|w\|_1)^D.$$

Proof. For any multi-index α let $w_\alpha := \prod_{i \in \alpha} w_i$ and observe by the multinomial theorem

$$p(w \cdot x) = \sum_i a_i (w \cdot x)^i = \sum_i a_i \sum_{|\alpha| = i} \binom{i}{\alpha} w_\alpha x^\alpha.$$

Therefore by the triangle inequality, multinomial theorem, and Cauchy-Schwarz inequality

$$\sum_\alpha |u_\alpha| \leq \sum_i |a_i| \sum_{|\alpha|=i} \binom{i}{\alpha} |w_\alpha| = \sum_i |a_i| \|w\|_1^i \leq \sqrt{\sum_i a_i^2 \sum_i \|w\|_1^{2i}} \leq \sqrt{\sum_i a_i^2}(1 + \|w\|_1)^d$$

where in the last step we used $1 + x^2 + x^4 + \cdots + x^k \leq (1+x)^k$ for $x \geq 0$. □

Theorem 10. *Suppose that* Y *is a random variable valued in* $\{\pm 1\}$*,* X *is a random vector such that* $\|X\|_\infty \leq 1$ *almost surely and*

$$\mathbb{E}[Y|X] = \tanh\left(b^{(1)} + \sum_j w_j f_{\beta_j}\left(b_j^{(2)} + \sum_k W_{jk} X_k\right)\right)$$

where $b^{(1)} \in \mathbb{R}$*,* $\beta_j \in [0, 1]$*,* w *is an arbitrary real vector and* W *is an arbitrary real matrix. Let* W_j *denote* column j *of* W*. Then* ℓ_1*-constrained regression on the degree* D *monomial feature map* $\varphi_D(x) \mapsto \left(\prod_{i \in S} X_i\right)_{|S| \leq d}$ *with* ℓ_1 *constraint*

$$\|w\|_1 \leq R := |b^{(1)}| + \sqrt{D+1}(4e)^{D+1} \sum_j |w_j|(1 + \|W_j\|_1)^{D+1}$$

returns a predictor $\hat{w}$ *such that with probability at least* $1 - \delta$*,*

$$\begin{aligned}
&\mathbb{E}[\ell(\hat{w} \cdot \varphi_d(X), Y)] - \mathbb{E}[\ell(v^*(X), Y)] \\
&\leq 8 \sum_j |w_j| \frac{\|W_j\|_1 + 2\|W_j\|_1^2}{(1 + 2/\|W_j\|_1)^D} + 4R\sqrt{\frac{2D\log(2n)}{m}} + 2R\sqrt{\frac{2\log(2/\delta)}{m}}
\end{aligned}$$

where $v^*(X) := \tanh^{-1}(\mathbb{E}[Y|X]) = b^{(1)} + \sum_j w_j f_{\beta_j}\left(b_j^{(2)} + \sum_k W_{jk} X_k\right)$ *is the minimizer of the expected logistic loss over all measurable functions of* X*. The runtime is* $poly(n^D)$*.*

Proof. The fact that $v^*(X)$ is the minimizer of the logistic loss $\mathbb{E}[\ell(h(X), Y)]$ over all X-measurable functions h can be seen from Fact 1. To derive the bound we combine the approximation-theoretic guarantees developed in the previous section with the ℓ_1 guarantee for logistic Lasso.

For the approximation step, define w^* so that $w^* \cdot \varphi_d(X)$ is given by replacing each activation f_{β_j} by its best polynomial approximation P_j on the interval $[b_j^{(2)} - \|W_j\|_1, b_j^{(2)} + \|W_j\|_1]$. By the triangle inequality and Lemma 9, for any $x \in \{\pm 1\}^n$,

$$|v^*(x) - w^* \cdot \varphi_d(x)| \leq \sum_j |w_j||(f_{\beta_j} - P_j)(b_j^{(2)} + \sum_k W_{jk}x_k)| \leq 4 \sum_j \frac{|w_j| \left(\|W_j\|_1 + 2\|W_j\|_1^2\right)}{(1 + 2/\|W_j\|_1)^D}.$$

Since the logistic loss is 2-Lipschitz (Fact 1.1), this implies that

$$\mathbb{E}[\ell(w^* \cdot \varphi_D(X), Y)] \leq \mathbb{E}[\ell(v^*(X), Y)] + 8 \sum_j \frac{|w_j| \left(\|W_j\|_1 + 2\|W_j\|_1^2\right)}{(1 + 2/\|W_j\|_1)^D}. \tag{4}$$

Combining Lemma 8, Lemma 10 and Lemma 11 and using the triangle inequality shows that $\|w^*\|_1 \leq R$ where R is as specified in the Theorem statement. Then applying Theorem 9 and combining it with (4) gives the desired inequality bounding the error of the predictor $\hat{w}$. □

To simplify usage of this Theorem, we give the following slightly less precise bound which will be used from now on:

Corollary 1. *In the same setting as Theorem 10, if we assume that $\|W_j\|_1 \leq \lambda$ for every j and $\lambda \geq 2$, then with probability at least $1 - \delta$, $\mathbb{E}[\ell(\hat{w} \cdot \varphi_d(X), Y)] - \mathbb{E}[\ell(v^*(X), Y)] \leq \epsilon$ as long as the number of samples m satisfies $m = \Omega((|b^{(1)}|^2 \lambda^{O(D)}) \log(2n/\delta))$ where $D = O(\lambda \log(\|w\|_1 \lambda/\epsilon))$ and the runtime of the algorithm is $poly(n^D)$.*

Proof. In order to make the first term of the bound on $\mathbb{E}[\ell(\hat{w} \cdot \varphi_d(X), Y)] - \mathbb{E}[\ell(v^*(X), Y)]$ at most $\epsilon/2$, we can upper bound it by $O(\|w\|_1 \lambda^2/(1 + 2/\lambda)^D)$ and see that it suffices to take $D = \Omega(\lambda \log(\|w\|_1 \lambda/\epsilon))$. Then $R = |b^{(1)}| + \exp(O(D))\|w\|_1 \lambda^{D+1} = |b^{(1)}| + \lambda^{O(D)}$ so it suffices to take $m = \Omega((|b^{(1)}|^2 + \lambda^{O(D)}) \log(2n/\delta))$ □

Remark 3. *In the analysis of Theorem 10 we did not concern ourselves with the exact constants in the runtime. However, if we are interested in optimizing the runtime it should be noted that instead of getting a precise estimate of the empirical risk minimizer when computing the logistic regression, one can achieve a similar statistical guarantee by using a single pass of stochastic mirror descent/exponentiated gradient (see reference text [57]), e.g. as used in [8] where the needed high-probability guarantees can be found.*

C.4 Nearly Matching computational lower bounds

In this section, we show that the runtime guarantee of Corollary 1 is close to optima: more precisely its runtime is optimal in $\|w\|_1$ and ϵ up to a $\log \log$ factor in the exponent, and also that at least sub-exponential dependence on λ is required. We first recall the definition of this problem and a standard hardness assumption for learning sparse parity with noise. We phrase it in terms of a testing problem versus the uniform distribution, which is equivalent to a learning formulation (i.e. recovering S below), by boosting the probability of success and using a standard reduction of removing one coordinate at a time and testing (see e.g. [20]).

Definition 5. *The* k-sparse parity with noise *distribution is the following distribution on (X, Y) parameterized by $\eta \in (0, 1/2)$ and an unknown subset S of size k:*

1. *Sample $X \sim \mathrm{Unif}(\{-1, +1\}^n)$.*
2. *With probability $1/2 + \eta$, set $Y = \prod_{s \in S} X_s$, and with probability $1/2 - \eta$, set $Y = (-1) \prod_{s \in S} X_s$.*

The k-sparse parity with noise *problem is to test between the uniform and k-sparse parity with noise with sum of probability of Type I and Type II errors upper bounded by 0.01, given access to an oracle which generates samples from one of the two distributions.*

Assumption 1 (Hardness of learning sparse parity with noise)**.** *Suppose k_n is an arbitrary sequence of positive integers with $k_n = o(n^{1-\epsilon})$ for any $\epsilon > 0$ and n growing, any algorithm which solve the k-sparse parity with noise testing problem must have runtime $n^{\Omega(k_n)}$.*

The reason for the condition $k_n = o(n^{1-\epsilon})$ is simply because the number of sets of size n is 2^n, not n^n, so small correction factors in the exponent are needed when k is comparable to n. The best known algorithm for learning sparse parity with noise runs in time $n^{0.8k_n}$ [20].

Theorem 11. *In the setting of Corollary 1 and under Assumption 1, for $\lambda \leq 2$ there exists families of models (one with ϵ a constant, one with $\|w\|_1$ a constant) where a runtime of*

$$n^{\Omega\left(\frac{\log(\|w\|_1/\epsilon)}{\log\log(\|w\|_1/\epsilon)}\right)}$$

is needed for any algorithm to achieve ϵ error with high probability, regardless of its sample complexity and even in the case of $\tanh$ *activations ($\beta_j = 0$ for all j). There also exists a sequence of models with $\lambda = \Theta(n\log(n))$ and $\|w\|_1 = O(\sqrt{n})$ which requires runtime*

$$n^{\Omega(\sqrt{\lambda/\log^2(\lambda)\log(n)}\log\|w\|_1)}$$

to achieve error $\epsilon = 0.01$ with high probability.

Proof. We first show a lower bound of $n^{\Omega(\log(\|w\|_1/\epsilon))}$ for a family of models where $\lambda \leq 1$. Recall we are proving a lower bound in the $\beta_j = 0$ case where all activations are $\tanh$. The lower bound is shown by building a parity function out of $\tanh$ functions exactly using a simple taylor series expansion argument, under the assumption that the input to the network is in the hypercube $\{\pm 1\}^n$. The construction proceeds in a similar fashion to the sparse parity with noise lower bound for learning RBMs of bounded hidden degree established in [18]. We first describe the construction of a parity function on boolean inputs $x_1, \ldots, x_k$. It suffices to build this parity with a small (constant-size) coefficient, since we can repeat it to make the coefficient larger. We start from the fact that

$$\tanh(z) = 2\sum_k \frac{(-1)^k}{\pi^{2k+2}}(1 - 1/4^{k+1})\zeta(2k+2)z^{2k+1}$$

for $|z| < \pi/2$ and recall that the Riemann ζ function does not vanish on even integers [54], so every coefficient in this expansion is nonzero. Furthermore it is known that $\zeta(n) \to 1$ as $n \to \infty$, since this follows from the power series definition of $\zeta(s) = \sum \frac{1}{n^s}$, so we can write

$$\tanh(z) = \sum_k a_{2k+1} z^{2k+1}$$

where $a_{2k+1} \neq 0$ for any k and $|a_{2k+1}| = \Theta(1/\pi^{2k+2})$. From this we can see that for some constant $c \neq 0$,

$$x_1 \cdots x_{2k+1} = c\frac{(2k+1)^{2k+1}}{a_{2k+1}} \tanh\left(\frac{x_1 + \cdots + x_{2k+1}}{2k+1}\right) + p(x)$$

where $p(x)$ is of degree at most $k-1$, using that $x_i^2 = 1$ for all i on the hypercube; here the constant c (which is close to 1) is a fixed correction factor to handle the small effect of maximum-degree terms coming from expanding higher order terms in the $\tanh$ power series. We can inductively rewrite each of the highest-order coefficients of p in terms of $\tanh$ and lower order monomials: this ultimately gives us a way to write parity as a linear combination of $\tanh$ functions. Using this, we can rewrite $\tanh(\frac{1}{4}x_1 \cdots x_{2k+1})$ as a two-layer $\tanh$ network with $\|w\|_1 = k^{O(k)}$ and $\lambda \leq 1$. Taking $\epsilon = 1/16$ and using the hardness of k-sparse parity with noise, we get that the runtime for learning the corresponding network is at least $n^{\Omega(k)} = n^{\Omega(\log(\|w\|_1)/\log\log(\|w\|_1))}$.

We can similarly prove a lower bound of $n^{\Omega(\log(1/\epsilon)/\log\log(1/\epsilon))}$ for constant $\lambda, \|w\|_1$ by using the same method to convert $\tanh(\eta x_1 \cdots x_{2k+1})$ into a two-layer network and by taking $\eta = k^{-\Theta(k)}$ so that the ℓ_1 norm of the coefficients is shrunk to be at most 1. Taking $\epsilon = \Theta(\eta)$ and using the sparse parity with noise lower bound as above gives the result.

Finally, we give a lower bound showing exponential dependence on λ is necessary. We use the well-known fact that a parity can be written as a small sum of threshold functions [47]. For k even,

$$x_1 \cdots x_k = \mathbb{1}[x_1 + \cdots + x_k \geq -k] - 2(\mathbb{1}[x_1 + \cdots + x_k \geq -k+2] - \mathbb{1}[x_1 + \cdots + x_k \geq -k+4] + \cdots)$$

with a total of $k+1$ terms in the sum on the rhs. We now consider replacing each threshold function with the approximation $\mathbb{1}[a \geq b] \approx \frac{1+\tanh(\lambda'(a-b+1/2))}{2}$ for some $\lambda' > 0$. Note that the error of this approximation for a singe threshold unit and integers a, b is maximized when $a - b = 0$ where the error is $\frac{1-\tanh(\lambda'/2)}{2} = O(e^{-\lambda'})$. Therefore by Holder's inequality, the error in approximating $x_1 \cdots x_k$ by replacing all of the threshold functions is $O(ke^{-\lambda'}) = O(ke^{-\lambda/(k+1/2)})$, where we used that $\lambda = (k+1/2)\lambda'$ where λ is the hidden node ℓ_1 norm as used previously. By adding a $\tanh$ nonlinearity on top of the approximate parity, this gives an approximate construction of sparse parity with noise.

Taking $k = \sqrt{n}$ and $\lambda = \Theta(k^2 \log(n))$ we see that the resulting model is $\mathbf{TV}$-distance $n^{-\Theta(k)}$ from sparse parity with noise, so any algorithm with runtime $cn^{-\Theta(k)}$ cannot distinguish this model from sparse parity with noise with probability better than 75% for sufficiently small constant $c > 0$. From the assumed hardness of learning sparse parity with noise, any algorithm succeeding to distinguish this model from the uniform distribution with sufficiently small error probability requires runtime $n^{\Omega(k)} = n^{\sqrt{\lambda/\log^2(\lambda)\log(n)}\log\|w\|_1}$. □

Remark 4. *In the second construction in the proof of Theorem 11, based off of approximating threshold functions, the computational lower bound becomes stronger if we allow the algorithm access to less data (recall that for a fixed noise level, $\Theta(k \log n)$ samples suffice information-theoretically for sparse parity with noise). If we only allow to use $\Theta(k \log n)$ samples as information-theoretically required, then we can take $\lambda = \Theta(k(\log k + \log\log n))$ and the runtime required is $n^k = n^{\lambda/(\log\log n + \log(\lambda))}$.*

D Learning RBMs by Learning Feedforward Networks

D.1 Structure and Distribution Learning Guarantees

In this section we discuss application of the prediction guarantees from the previous section to structure and distribution learning. As motivation, recall that in undirected graphical models the *Markov blanket* or *neighborhood* of a node i, the minimal set of nodes which separate node i from the rest of the model in the underlying graph, is one of the most interesting pieces of information to learn about a node. By the Markov property, node i interacts directly only with nodes in its Markov blanket, in the sense that X_i is conditionally independent of all other nodes X_k given the values of nodes X_j for all j in the markov blanket of i. Learning the markov blanket of all nodes, equivalently learning the underlying graph of the Markov Random Field, is referred to as *structure learning*. It is also known (see e.g. [18]) that once we have performed structure learning, distribution learning (e.g. in total variation distance) becomes a conceptually straightforward task as it can typically be reduced to solving low-dimensional regression problems.

As explained in the introduction, learning the structure requires a non-degeneracy condition on neighbors (recall the definition of η-nondegeneracy from above). In the introduction, we stated that if all edges are η-nondegenerate then we can learn the structure perfectly; in the next Theorem, we state a slightly more precise result giving the result we can successfully test between non-neighbors and η-nondegenerate neighbors, without requiring nondegeneracy on the entire model. Since our guarantee holds with high probability, using the union bound it immediately gives a result for structure recovery under η-nondegeneracy.

Theorem 12. *Let i and j be two visible nodes in a (λ_1, λ_2)-bounded RBM. Let H_0 be the hypothesis that nodes i and j are not two-hop neighbors and H_1 the hypothesis that nodes i and j are η-nondegenerate two-hop neighbors. Given $\delta > 0$ and $m = \Omega(\lambda_2^{O(D)} \log(2n/\delta))$ i.i.d. samples where $D = O(\lambda_2 \log(\lambda_1\lambda_2/\eta))$, we can test in time $poly(n^D)$ between H_0 and H_1 with sum of Type I and Type II errors upper bounded by δ.*

Proof. We run the following testing procedure:

1. Run the ℓ_1 regression algorithm from Theorem 1 to predict X_i from $X_{\sim i}$ and from $X_{\sim i,j}$.
2. Repeat the previous step with i and j reversed.

3. If the decrease in prediction accuracy for removing i or j is at least $3\eta/4$ in either step 1 or step 2, reject H_0.

That this works follows by combining Theorem 1 and Corollary 1, by choosing $\epsilon = \eta/8$ under H_0 the difference in prediction error is at most 2ϵ whereas under H_1 it must be at least $\eta - 2\epsilon$. □

Assuming that all 2-hop neighbors in the RBM are η-nondegenerate, the above Theorem lets us recover the structure of the RBM (its 2-hop neighborhoods) in time $poly(n^D)$. In the following remark, we explain how large D is in the regimes where we know polynomial time sampling from the RBM is possible:

Remark 5 (Comparison to polynomial time sampling regimes)**.** *Dobrushin's uniqueness criterion is probably the most well-known sufficient condition for sampling to be possible in polynomial time in a general pairwise model. Dobrushin's condition is that for every node i, the total ℓ_1-norm of the edges touching node i is at most 1, where the mixing time guarantees for Glauber dynamics become worse as the maximum norm approaches 1 (see [28]). This condition is tight in the example of the Ising model on the complete graph (Curie-Weiss), or for the bipartite complete graph (i.e. dense RBM) with all edge weights positive and equal and an equal number of visible and hidden units.*

Under Dobrushin's uniqueness criterion on the RBM, we have that $\lambda_1, \lambda_2 \leq 1$ so $D = O(\log(1/\eta))$. As mentioned above, we cannot compute η in terms of just the edge weights for general models, but if we for example assume the model is d-regular and has all edge weights equal to $+1/d$ and no external field then it is not too hard to show that $\eta = \Omega(1/d^2)$ (see e.g. [18]), so in this case the overall runtime is $n^{\log(d)}$. We expect that under Dobrushin's condition $\eta = \Omega(1/d^2)$ except in perhaps some rare degenerate situations. This means the runtime is improved by an exponential factor in the exponent compared to what one gets by just applying the RBM to MRF reduction, since learning d-wise MRFs is known to require n^d time in general [8].

In some other interesting contexts, it is also known that polynomial time sampling can only be guaranteed when $\lambda_1, \lambda_2 = O(1)$: for antiferromagnetic Ising models on bounded degree graphs with equal edge weights the sharp result is known for every d [58, 59, 29] and embedding these Ising models as RBMs with hidden nodes of degree 2 in a straightforward way gives models with $\lambda_1, \lambda_2 = O(1)$ and $\eta = \Omega(1/d^2)$ (see Example 1 above).

For distribution learning we will need the following technical Lemma, which is proved in Appendix D.2 using the local Rademacher complexity framework [60]. Informally it says that if X is a random variable with a density with respect to the uniform measure on $\{\pm 1\}^n$ that is lower bounded by a constant, then given a number of samples m which is large with respect to the size of the domain the natural estimator of $\tanh^{-1}(\mathbb{E}[Y|X])$ has error which converges at a $1/m$ rate, which generalizes the case of estimating the (exponential-family parameterization of) mean, the $n = 0$ case, in a natural way. Since the bound depends exponentially on n, we will only apply it in settings where we expect n is small. Similar bounds are used in previous works including [5, 6] and proved using different methods, though they are not quite as optimized (e.g. deriving this result from Lemma 3.2 of [6] would give a $1/\gamma^2$ dependence); this bound can be shown to be optimal up to constants.

Lemma 12. *Suppose that X is a random variable valued in $\{\pm 1\}^n$ with $\Pr(X = x) \geq \gamma/2^n$ for every x and Y is a random variable valued in $\{\pm 1\}$. Suppose that $|\mathbb{E}[Y|X]| \leq r$ for $r < 1$. Let $\hat{\mathbb{E}}[Y|X]$ be the empirical conditional expectation of Y given X based upon m i.i.d. samples of (X, Y) and define $h(X) := \min(\max(\mathbb{E}[Y|X], r), -r)$. Then with probability at least $1 - \delta$,*

$$\mathbb{E}[(\tanh^{-1}(h(X)) - \tanh^{-1}(\mathbb{E}[Y|X]))^2] \lesssim \frac{2^n/\gamma + \log(1/\delta)}{(1 - r^2)^2 m}$$

where $\lesssim$ denotes inequality up to an absolute constant.

We present the proof of this lemma in the subsequent subsection. From this Lemma we straightforwardly get the right result for learning a sparse RBM with known 2-hop neighborhoods.

Lemma 13. *For any (λ_1, λ_2)-bounded RBM where the maximum two-hop degree of any visible node is at most d_2 and where $\|b^{(1)}\|_\infty \leq B$, for $\delta > 0$ and $m = \Omega\left(n^2 \left(\frac{2}{(1-\tanh(\lambda_1))}\right)^{d_2+1} \log(n/\delta)/\epsilon^4\right)$ we have that with probability at least $1 - \delta$, Algorithm* DISTRIBUTIONFROMSTRUCTURE *given m*

Algorithm 3 DISTRIBUTIONFROMSTRUCTURE

1: We assume for every node i we are given a recovered neighborhood $\widehat{\mathcal{N}}(i)$. $\widehat{\mathcal{N}}(i)$
2: For every node i with neighborhood $\widehat{\mathcal{N}}(i)$, let $f_i(X) := \widehat{\mathbb{E}}[X_i|X_{\widehat{\mathcal{N}}(i)}]$ be the empirical conditional expectation of X_i given $X_{\widehat{\mathcal{N}}(i)}$.
3: Return the output of Algorithm DISTRIBUTIONFROMPREDICTORS run with these f_i.

samples and $\widehat{\mathcal{N}}(i) = \mathcal{N}(i)$ for every i returns a distribution $\hat{P}$ which is ϵ-TV close to the distribution of the RBM. Furthermore, if $w_S, \hat{w}_S$ are as defined as in Lemma 6 then

$$2\mathbf{TV}(P, \hat{P})^2 \leq \mathbf{SKL}(P, \hat{P}) \leq \sum_S |w_S - \hat{w}_S| \leq \epsilon^2.$$

Proof. By Lemma 6, Lemma 5 and Lemma 12 we have

$$\begin{aligned}
\mathbf{SKL}(\hat{P}, P) &\leq \sum_S |w_S - \hat{w}_S| \\
&\leq \sum_i \frac{2^{d_2/2+1}}{(1-\tanh(\lambda_1))^{d_2/2}} \sqrt{\mathbb{E}_{X\sim Uni(\{\pm 1\}^n)}[(\tanh^{-1}(h_i(X)) - \tanh^{-1}(\mathbb{E}_P[X_i|X_{\sim i}])^2]} \\
&\leq \sum_i \frac{2^{d_2/2+1}}{(1-\tanh(\lambda_1))^{d_2}} \sqrt{\mathbb{E}_{X_{\mathcal{N}(i)}}[(\tanh^{-1}(h_i(X)) - \tanh^{-1}(\mathbb{E}_P[X_i|X_{\sim i}])^2]} \\
&\leq \sum_i \frac{2^{d_2/2+1}}{(1-\tanh(\lambda_1))^{d_2}} \sqrt{\frac{2^{d_2}/(1-\tanh(\lambda_1))^{d_2} + \log(n/\delta)}{(1-\tanh(\lambda_1)^2)^2 m}}
\end{aligned}$$

and by Pinsker's inequality $\mathbf{TV}(\hat{P}, P)^2 \leq \mathbf{SKL}(\hat{P}, P)/2$ so the result follows. □

Theorem 13. *Suppose that all visible nodes in an RBM which are neighbors in the Markov blanket sense are η-nondegenerate neighbors, and that maximum 2-hop degree of any visible node is at most d_2. Then given $\delta > 0$ and $m = \Omega(\lambda_2^{O(D)} \log(2n/\delta) + n^2 \left(\frac{2}{(1-\tanh(\lambda_1))}\right)^{d_2+1} \log(n/\delta)/\epsilon^4)$ i.i.d. samples where $D = O(\lambda_2 \log(\lambda_1\lambda_2/\eta))$ samples, Algorithm* DISTRIBUTIONFROMSTRUCTURE *run with the set of η-nondegenerate neighbors output by Theorem 12 returns with probability at least $1-\delta$ a distribution which is ϵ-TV close to the true distribution of the RBM.*

Proof. This follows by combining Theorem 12 and Lemma 13. □

Remark 6. *If we do not assume that all neighbors are η-nondegenerate, then by Theorem 16 it is impossible to get a nontrivial distribution learning guarantee assuming the hardness of learning sparse parity with noise, in the sense that the naive approach of forgetting the RBM structure entirely and using MRF learning results (e.g. [8]) cannot be improved.*

D.2 Proof of Lemma 12

We recall the statement of Lemma 12. Suppose that X is a random variable valued in $\{\pm 1\}^n$ with $\Pr(X = x) \geq \gamma/2^n$ for every x and Y is a random variable valued in $\{\pm 1\}$. Suppose that $|\mathbb{E}[Y|X]| \leq r$ for $r < 1$. Let $\hat{\mathbb{E}}[Y|X]$ be the empirical conditional expectation of Y given X based upon m i.i.d. samples of (X, Y) and define $h(X) := \min(\max(\mathbb{E}[Y|X], r), -r)$. Then with probability at least $1-\delta$,

$$\mathbb{E}[(\tanh^{-1}(h(X)) - \tanh^{-1}(\mathbb{E}[Y|X]))^2] \lesssim \frac{2^n}{\gamma(1-r^2)^2 m} + \frac{\log(1/\delta)}{(1-r^2)^2 m}$$

We will prove the result by proving the analogous result without the $\tanh^{-1}$ first, as Lemma 14. The following general result reduces this to computing the local Rademacher complexity of the corresponding function class.

Theorem 14 (Corollary 5.3 of [60]). *Suppose that $\mathcal{F}$ is a class of functions from $\mathcal{X}$ to $[-1,1]$ and $\ell(\hat{y},y)$ is a loss which satisfies:*

1. *ℓ is L-Lipschitz in $\hat{y}$.*
2. *There is a constant $B \geq 1$ such that for any random variable X supported on $\mathcal{X}$ and random variable Y on $[-1,1]$*

$$\mathbb{E}(f(X) - f^*(X))^2 \leq B\mathbb{E}[\ell(f(X),Y) - \ell(f^*(X),Y)]$$

where $f^*(X)$ is a minimizer of $\mathbb{E}[\ell(f(X),Y)]$ which we assume exists.

Then if $\psi(r)$ is a sub-root function (meaning a monotonically increasing non-negative function with $\psi(r)/\sqrt{r}$ monotonically decreasing) such that

$$\psi(r) \geq BL\mathbb{E} \sup_{f\in\mathcal{F}, L^2\mathbb{E}[(f-f^*)^2]\leq r} \frac{1}{m}\sum_{i=1}^{m} \sigma_i(f-f^*)(X_i) \tag{5}$$

where the σ_i are i.i.d. Rademacher random variables, then for any $r \geq \psi(r)$ with probability at least $1-\delta$

$$\mathbb{E}[\ell(\hat{f}(X),Y) - \ell(f^*(X),Y)] \lesssim \frac{r}{B} + \frac{(L+B)\log(1/\delta)}{m}$$

where the notation $\lesssim$ hides an absolute constant.

Lemma 14. *Under the same setup as Lemma 12,*

$$\mathbb{E}[(h(X) - \mathbb{E}[Y|X])^2] \lesssim \frac{2^n}{\gamma m} + \frac{\log(1/\delta)}{m}.$$

Proof. We consider $\mathcal{F}$ the class of arbitrary functions from $\mathcal{X}$ to $[-r,r]$ and take $\ell(\hat{y},y) := (\hat{y}-y)^2$ to be the square loss so $L=2$ and $B=1$ satisfy the conditions above. It is clear from the definition of h that it is the empirical risk minimizer for this function class and loss. Since this class is convex we can take $\psi(r)$ to be defined by the rhs of (5) (Lemma 3.4 of [60]) and it remains to compute the fixed point of ψ. Thus if we write $g := f - f^*$

$$\psi(r) = 2\mathbb{E} \sup_{f:4\mathbb{E}[g^2]\leq r} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(X_i)$$

and we observe by the assumption $\Pr(X=x) \geq \gamma/2^n$ that

$$\mathbb{E}_X[g^2] \geq \gamma \mathbb{E}_{X'\sim Uni(\{\pm 1\}^n)}[g(X')^2] = \gamma \sum_S \widehat{g}(S)^2$$

by Plancherel's Theorem [52] where $\widehat{g}(S)$ denotes the Fourier coefficient of g corresponding to set S, so that $g(x) = \sum_S \widehat{g}(S)x_S$ where $x_S = \prod_{s\in S} x_s$. Therefore by the above, the Cauchy-Schwarz inequality, and Jensen's inequality we have

$$\begin{aligned}
\psi(r) &= 2\mathbb{E} \sup_{g:4\mathbb{E}[g^2]\leq r} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(X_i) \\
&\leq 2\mathbb{E} \sup_{g:\sum_S \hat{g}(S)^2 \leq r/4\gamma} \frac{1}{m}\sum_S \hat{g}(S) \frac{1}{m}\sum_{i=1}^{m} \sigma_i (X_i)_S \\
&\leq \sqrt{r/\gamma}\mathbb{E}\frac{1}{m}\sqrt{\sum_S \left(\sum_{i=1}^{m} \sigma_i (X_i)_S\right)^2} \\
&\leq \frac{\sqrt{r}}{m\sqrt{\gamma}}\sqrt{\mathbb{E}\sum_S \left(\sum_{i=1}^{m} \sigma_i (X_i)_S\right)^2} = \frac{\sqrt{r}}{\sqrt{m\gamma}} 2^{n/2}.
\end{aligned}$$

Solving for the fixed point of $r = \frac{\sqrt{r}}{\sqrt{m\gamma}}2^{n/2}$ gives $r^* = \frac{2^n}{\gamma m}$ so the result follows from Theorem 14. □

Proof of Lemma 12. Recall that the derivative of $\tanh^{-1}$ at x is $\frac{1}{1-x^2}$. Therefore on the domain $[-r, r]$ the function $\tanh^{-1}$ is $\frac{1}{1-r^2}$ Lipschitz. Therefore by the mean value theorem,

$$\mathbb{E}[(\tanh^{-1}(h(X)) - \tanh^{-1}(\mathbb{E}[Y|X]))^2] \leq \frac{1}{(1-r^2)^2}\mathbb{E}[(h(X) - \mathbb{E}[Y|X])^2]$$

and applying Lemma 14 gives the result. □

D.3 Matching Computational Lower Bounds

In the following sequence of theorems we show that our runtime guarantees for structure learning of RBMs cannot be significantly improved. The first result relies in part on the representation of sparse parity with noise given in [18]; this embedding is constructed in a similar way to the first embedding used in Theorem 11. It shows the dependence on λ_1 and η is correct when asking for structure recovery.

Theorem 15. *In the same setup as Theorem 12 and under Assumption 1, there exists a family of instances parameterized by n going to infinity with $\lambda_2 \leq 2$ such that any algorithm which is able to achieve structure recovery for a model with all neighbors being η-nondegenerate requires runtime $n^{\Omega(\log(\lambda_1/\eta)/\log\log(\lambda_1/\eta))}$, regardless of its sample complexity.*

Proof. In [18], it was shown that for any fixed constant η (say $\eta = 1/8$), there exists an embedding of k-sparse parity with noise into an RBM where every hidden unit has incoming edges of total ℓ_1 norm upper bounded by 2 (i.e. satisfying $\lambda_1 \leq 2$) and there are $2^{O(k)}$ hidden units; it can be checked straightforwardly that for $\eta = 1/8$ that $\lambda_2 = k^{O(k)}$. Therefore if we fix $\epsilon = \eta/2$ then when assuming the hardness of k-sparse parity with noise there is a $n^{\Omega(k)}$ runtime lower bound which matches since $\lambda_2 = e^{O(k)}$.

For the tightness in ϵ, by making the parity bias η exponentially small in $k\log(k)$, it's easy to check that by repeating the construction in [18] that we can make λ_2 a constant; then to find the parity with noise one needs ϵ exponentially small in $k\log k$ as well, and the hardness assumption implies the runtime must be $n^{\Omega(k)}$. □

By tensorizing this construction, we show that the η-nondegeneracy assumption is required, even if we only care about distribution learning. More precisely, we need it to learn in TV distance with runtime better than the pessimistic $n^{O(d_h)}$ result which follows from viewing the RBM as an unstructured MRF and using the result of [8].

Theorem 16. *There exists a family of RBMs with n nodes, maximum hidden node degree d_H, and $\lambda_1, \lambda_2 = O(1)$ such that any algorithm which can learn this family of RBMs within total variation distance at most $1/4$ requires $n^{\Omega(d_H)}$ time.*

Proof. The construction in Theorem 15 shows that there exists a family of RBMs given by embedding sparse parity with noise with the desired property, except that the total variation distance is only guaranteed to be $2^{-O(d_H\log(d_H))}$. By building a larger RBM consisting of $2^{d_H\log(d_H)}$ disjoint copies of the original RBM (note that the resulting increase in n is a multiplicative factor independent of the original n), we can boost the total variation distance to be arbitrarily close to 1. □

In order to give lower bounds with respect to λ_2 for fixed η, we need a significantly more involved argument. We first recall an approximate construction of parity (with low levels of noise) from [48]:

Theorem 17 (Theorem 7 of [48])**.** *There exists an RBM network with n^2+1 hidden units and weights $poly(n, \log(1/\epsilon))$ such that the marginal distribution P on the visible units satisfies $P(x) \propto e^{f(x)}$ for some f satisfying*

$$\sup_{x\in\{\pm 1\}^n} |f(x)/C - x_1\cdots x_n| \leq \epsilon$$

where $C > 0$ satisfies $C = poly(\log(n), \log(1/\epsilon))$.

This construction is for a dense parity, but obviously we can make the parity as sparse as we want by adding additional visible units not connected to anything else. More significantly, since the above

theorem only constructs an ϵ-approximate instance of parity with noise $\eta = O(1/2 - 1/poly(n, 1/\epsilon))$, when n or $1/\epsilon$ is large it does not seem that the resulting distribution is computationally hard to distinguish from the uniform distribution, since Gaussian elimination over $\mathbb{F}_2$ has some chance of succeeding to find the parity. Since we need ϵ to be small for the model to be indistinguishable from sparse parity with noise, this appears to be a barrier to deriving a hardness result from the above Theorem. Instead, we will prove that our result cannot be significantly improved for SQ (Statistical Query) algorithms (for a reference, see [49]). In the Statistical Query model algorithms do not have access to data, but instead have access to an SQ oracle:

Definition 6. *An oracle for the statistical query model over distribution $\mathcal{D}$ over X, Y takes input (g, τ) where g is a function $g : \{\pm 1\}^n \times \{\pm 1\} \to [-1, 1]$ and τ is a tolerance, and gives output v with*

$$|\mathbb{E}_{X,Y\sim D}[g(X, Y)] - v| \leq \tau.$$

Standard arguments, i.e. implementing the needed regressions using standard gradient-based methods for convex optimization shows that our algorithm for learning RBMs can be implemented in the statistical query model (in this case, the separation of X and Y in the definition above is somewhat artificial but we will take Y to be a particular visible unit in the RBM). We will show that statistical query algorithms cannot do better than subexponential dependence on λ_2.

The following theorem statements a lower bound for learning concepts of large SQ-dimension in the Statistical Query model. The definition of SQ-dimension can be found in [49], but for our purposes the only needed fact is that the class of k-parities over the uniform distribution $\{\pm 1\}^n$ has SQ-dimension $\binom{n}{k}$ [49].

Theorem 18 ([49])**.** *Let $\mathcal{F}$ be a class of functions over $\{\pm 1\}^n$ and D a distribution such that SQ-DIM$(\mathcal{F}, D) \geq d \geq 16$. Then if all queries are made with tolerance at least $1/d^{1/3}$, then at least $d^{1/3}/2$ queries are required to learn $\mathcal{F}$ with error less than $1/2 - 1/d^3$ in the statistical query model.*

Theorem 19. *Let S be an unknown subset of $[n]$ of size k and containing n and $\mathcal{D}$ is the distribution of the RBM produced by Theorem 17 on S where the other $n - |S|$ visible units are isolated and without external field. Let $\mathcal{F}$ be the class of parities on $[n - 1]$. As before, λ_2 refers to the maximum ℓ_1-norm into any hidden unit and we choose parameters so that $\lambda_2 = poly(n)$ and $\|w\|_1 = poly(n)$. There exists $\epsilon > 0$ so that no SQ algorithm with tolerance $n^{-\lambda_2^\epsilon}$ and access to $n^{\lambda_2^\epsilon}$ queries can learn $\mathcal{F}$ with error less than $1/4$.*

Proof. In Theorem 17 we take $\epsilon = \exp(-n)$ which gives $\lambda_2 = poly(n)$. The resulting RBM is then within TV distance $\exp(-n)$ of the distribution of a parity over the uniform distribution with a small amount of label noise, so an SQ algorithm for the RBM setting implies an SQ algorithm for learning parity, and the result follows from the lower bound of Theorem 18. □

E Learning a Feedforward Network by Learning RBMs

In this section we reverse the connection between RBMs and Feedforward networks by using RBMs with certain structural assumptions as a useful *distributional assumption* for learning feedforward network. More formally, we assume our data is generated by the following Supervised RBM.

Definition 7. *A* Supervised Restricted Boltzmann Machine *is any joint distribution over random variables X valued in $\{\pm 1\}^{n_1}$, H valued in $\{\pm 1\}^{n_2}$ and label $Y \in \{\pm 1\}$ of the form*

$$\Pr[X = x, H = h, Y = y] \propto \exp\left(\langle x, Wh\rangle + \langle h, w\rangle y + \langle b^{(1)}, x\rangle + \langle b^{(2)}, h\rangle + b^{(3)} y\right)$$

where the weight matrix *W is an arbitrary $n_V \times n_H$ matrix and* external fields/biases *$b^{(1)} \in \mathbb{R}^{n_1}$, $b^{(2)} \in \mathbb{R}^{n_2}$ and $b^{(3)}$ are arbitrary, and X is referred to as the vector of* visible unit *activations and H the vector of* hidden unit *activations.*

We make the following additional assumptions on the parameters of the model.

Assumption 2 (Minimum Ferromagnetic Interaction)**.** *For all $i \in [n_1], j \in [n_2]$ either $W_{ij} = 0$ or $W_{ij} \geq \alpha$.*

We do not make any assumption on the weight w to the label. Therefore the model overall is not ferromagnetic.

Assumption 3 (Sparsity). *For all $i \in [n_1]$, $\sum_{j=1}^{n_2} W_{ij} + |b_i^{(1)}| \le \lambda$ and for either $y = -1$ or $y = 1$, for all $j \in [n_2]$ $\sum_{i=1}^{n_1} W_{ij} + |b_j^{(2)} + yw_j| \le \lambda$.*

Here the sparsity assumption implies that under the conditioning of the label to either value, the sparsity parameter is bounded. This conditional sparsity can be exploited by an algorithm for learning the conditional distribution whereas a direct regression algorithm may be unable to gain from the same.

Remark 7. *Observe that the generative model of X itself is not sparse since Y is connected to all hidden nodes however conditioned on knowing the label Y, the model is now sparse. This assumption is more reasonable than assuming sparsity directly on the model of X which may not hold.*

Assumption 4 (Balanced Label). *For $y \in \{\pm 1\}$, $\Pr[Y = y] \ge \beta$.*

The above assumption essentially rules out trivial constant learners. Using data, it is easy to check if this assumption is satisfied or not.

As before, we can compute the conditional mean function of the label as follows:

$$\mathbb{E}[Y|X = x] = \tanh\left(b^{(3)} + \sum_j \tanh^{-1}\left(\tanh(w_j)\nu_j\right)\right)$$

where $\nu_j := \tanh\left(b_j^{(2)} + \sum_i \tanh^{-1}\left(\tanh(W_{ij})X_i\right)\right) = \tanh\left(b_j^{(2)} + \sum_i W_{ij}X_i\right)$. This represents a 2-layer neural network and in the limit of infinite hidden nodes, it can represent all 2-layer $\tanh$ networks (see Lemma 2).

Assumption 5 (Boundedness). *When $\mathbb{E}[Y|X = x]$ is re-expressed as $\tanh(f^*(x) + b^*)$ for some function f^* with no constant term and $b^* \in \mathbb{R}$. $|b^*| \le B$ for some $B > 0$.*

The above assumption intuitively says that the effect on Y that does not depend on X is bounded. B can be bounded in terms of the network parameters.

Also observe that conditioned on a fixed label,

$$\Pr[X = x, H = h|Y = y] \propto \exp\left(\langle x, Wh\rangle + \langle b^{(1)}, x\rangle + \langle b^{(2)} + wy, h\rangle\right)$$

which is a sparse, ferromagnetic RBM with arbitrary external field. Thus, we capture a neural network problem with a conditional RBM distributional assumption on the input. This distributional assumption seems more natural than the Gaussian input distribution which is extensively used in prior work. Also, this assumption allows us to leverage prior known algorithms for structure learning of ferromagnetic RBMs to learn the prediction function.

E.1 Preliminaries: Structure Learning of RBMs with Ferromagnetic Interactions

Consider a RBM with the following additional assumptions:

Assumption 6 (Minimum Ferromagnetic Interaction). *For all $i \in [n_1], j \in [n_2]$ either $W_{ij} = 0$ or $W_{ij} \ge \alpha$.*

Assumption 7 (Sparsity). *For all $i \in [n_1]$, $\sum_{j=1}^{n_2} W_{ij} + |b_i^{(1)}| \le \lambda$ and for all $j \in [n_2]$, $\sum_{i=1}^{n_1} W_{ij} + |b_j^{(2)}| \le \lambda$.*

Under these assumptions, [19] has shown that a simple greedy algorithm based on covariance maximization suffices to learn the structure of the RBM. Under the further assumption of non-negative external fields, [18] previously showed a similar greedy maximization algorithm with better dependence on the sparsity parameter λ.

The crucial structural property that [19] use is their algorithm is the following strengthening of the FKG inequality,

Lemma 15 (Lemma 2 of [19]). *For any observed nodes u, v and set $S \subseteq [n_1]\backslash\{u, v\}$,*

$$\mathsf{Cov}(u, v|X_S = x_S) := \mathbb{E}[X_u X_v|X_S = x_S] - \mathbb{E}[X_u|X_S = x_S]\,\mathbb{E}[X_v|X_S = x_S] \ge \alpha^2 \exp(-12\lambda).$$

Subsequently they define *average conditional covariance* $\mathrm{Cov}^{\mathsf{Avg}}(u, v|S) = \mathbb{E}_{x_S}[\mathsf{Cov}(u, v|X_S = x_S)]$ which straightforwardly is lower bounded by an application of the above lemma. Their final algorithm essentially greedily maximizes this average conditional covariance to build the neighborhood.

Theorem 20 (Theorem 2 of [19])**.** *Consider M samples $\mathcal{S}$ drawn from a RBM with arbitrary external field satisfying the given assumptions. For $\tau = \frac{\alpha^2}{2}\exp(-12\lambda)$ and $\delta = \exp(-2\lambda)/2$, with probability $1-\zeta$,* LEARNRBMNBHD$(u, \tau, \mathcal{S})$ *outputs* exactly *the two-hop neighborhood of observed variable u for*

$$M \geq \Omega\left((\log(1/\zeta) + T^* \log(n))\,\frac{2^{2T^*}}{\tau^2\delta^{2T^*}}\right) \text{ and } T^* = \frac{8}{\tau^2}.$$

Moreover, the algorithm runs in time $O(T^* M n)$.

E.2 Prediction from Distribution Learning

Here we will present our algorithm for learning the supervised RBM followed by a proof of its correctness. Instead of learning the label function directly, we will instead first learn the underlying generative model of X conditioned on a particular value of the label and use this knowledge to predict Y.

Theorem 21. *Given a supervised RBM satisfying Assumption 2, 3, 4 and 5, there exists an algorithm with sample complexity $m = n^2\exp(\lambda)^{\exp(O(\lambda))}(1/\alpha)^{O(1)}(1/\beta)^{O(1)}\log(n/\delta)/\epsilon^2$ and runtime $poly(m)$ returns hypothesis h such that,*

$$\mathbb{E}[\ell(h(X), Y] - \mathbb{E}[\ell(h^*(X), Y] \leq \epsilon$$

where ℓ is the logistic loss and h^* is the minimizer of the logistic loss.

Remark 8. *For an example where this algorithm is better than if we have no distributional assumptions, observe that we can construct a ferromagnetic RBM where $\mathbb{E}[Y|X]$ is a sparse parity function by adapting in a straightforward way the reduction used in the proof of the part of Theorem 11 with bounded λ (the use of $\tanh$ as opposed to f_β in that construction is not fundamental, or we can use a finite version of Lemma 2), since the hidden units in that proof all have nonnegative weights. It's clear why Algorithm* LEARNSUPERVISEDRBMNBHD *is better than an algorithm which doesn't know the input distribution: under the true input distribution, the visible units involved in the parity are correlated so the algorithm can find them, which makes learning the sparse parity easy.*

Our main algorithm can be broken down into three main steps: 1) Use greedy maximization (similar to Algorithm 1 of [19]) to first learn the two-hop neighborhood $\mathcal{N}(i)$ of each observed variable i w.r.t. the hidden layer conditioned on the label, 2) For each observed variable X_i, learn the distribution for $X|Y = y$ for $y = \pm 1$, and 3) Use the estimated distribution to compute $\mathbb{E}[Y|X]$.

Structure Learning For notation simplicity, we will overload notation and represent $\mathsf{Cov}^{\mathsf{Avg}}(u, v|S, Y) = \mathbb{E}_{x_S, y}[\mathsf{Cov}(u, v|X_S = x_S, Y = y)]$ where $\mathsf{Cov}(u, v|X_S = x_S, Y = y) = \mathbb{E}[X_u X_v|X_S = x_S, Y = y] - \mathbb{E}[X_u|X_S = x_S, Y = y]\,\mathbb{E}[X_v|X_S = x_S, Y = y]$. Then for structure learning, our algorithm essentially follows Algorithm 1 of [19] with the slight modification of conditioning w.r.t. Y.

Theorem 22. *Consider m samples $\mathcal{S}$ drawn from a supervised RBM satisfying Assumption 2, 3 and 4. For $\tau = \frac{\beta\alpha^2}{2}\exp(-12\lambda)$ and $\delta = \exp(-2\lambda)/2$, with probability $1-\zeta$,* LEARNSUPERVISEDRBMNBHD$(u, \tau, \mathcal{S})$ *outputs* exactly *the two-hop neighbors of observed variable u w.r.t. the hidden layer, with*

$$m \geq \Omega\left((\log(1/\zeta) + T^* \log(n))\,\frac{2^{2T^*}}{\tau^2\beta\delta^{2T^*}}\right) \text{ and } T^* = \frac{8}{\tau^2}.$$

Moreover, the algorithm runs in time $O(T^* M n)$.

Proof. In order to apply Theorem 20 to our setting, the only two properties we need to show are 1) given the conditioning of Y, the average conditional covariance bound still holds, that is,

$\mathsf{Cov}^{\mathsf{Avg}}(u, v|S \cup \{0\})$ is lower bounded for all $S \subseteq [n_2] \backslash \{u, v\}$ for v in the two-hop neighborhood of u, 2) $\Pr[X_S = x_S, Y = y]$ for all x_S and y. We have,

$$\mathsf{Cov}^{\mathsf{Avg}}(u, v|S, Y) = \sum_{y \in \pm 1} \sum_{x_S \in \{\pm 1\}^{|S|}} \Pr[X_S = x_S, Y = y] \mathsf{Cov}(u, v|X_S = x_S, Y = y)$$

By Assumption 3, we know that either for $y = 1$ or $y = -1$ (say $y = 1$ WLOG), the resulting RBM is sparse therefore we can apply Lemma 15 to the ones conditioned on $y = 1$. Also, we know that $\mathsf{Cov}(u, v|X_S = x_S, Y = y) \geq 0$ for all x_S and y due to FKG inequality for ferromagnetic RBMs. This implies that,

$$\begin{aligned}
\mathsf{Cov}^{\mathsf{Avg}}(u, v|S, Y) &\geq \sum_{x_S \in \{\pm 1\}^{|S|}} \Pr[X_S = x_S, Y = 1] \mathsf{Cov}(u, v|X_S = x_S, Y = 1) \\
&\geq \sum_{x_S \in \{\pm 1\}^{|S|}} \Pr[X_S = x_S, Y = 1] \alpha^2 \exp(-12\lambda) \\
&\geq \Pr[Y = 1] \alpha^2 \exp(-12\lambda) \geq \beta \alpha^2 \exp(-12\lambda).
\end{aligned}$$

For the second part, let us order the elements of S of size k as $s_1, \ldots, s_k$, then we have

$$\begin{aligned}
\Pr[X_S = x_S, Y = y] = \Pr[Y = y] \times \Pr[X_{s_1} = x_{s_1}|Y = y] \times \Pr[X_{s_2} = x_{s_2}|X_{s_1} = x_{s_1}, Y = y] \times \ldots \\
\times \Pr[X_{s_k} = x_{s_k}|X_{s_1} = x_{s_1}, \ldots, X_{s_{k-1}} = x_{s_{k-1}}, Y = y]
\end{aligned}$$

Since l_1-norm to the observed nodes is bounded by λ, by Bresler's property (see [6]) we have $\Pr[X_{s_r} = x_{s_r}|X_{s_1} = x_{s_1}, \ldots, X_{s_r} = x_{s_r}, Y = y] \geq \delta$. This implies that $\Pr[X_S = x_S, Y = y] \geq \beta \delta^{|S|}$ for all values of x_S and y. Now by applying Theorem 20 with the correct parameters, we get the required result. □

Distribution Learning Given the neighborhood of each observed node, we run Algorithm DISTRIBUTIONFROMSTRUCTURE and subsequently use Lemma 13 to guarantee that we obtin the weights of the unnormalized MRFs for distributions $X|Y = y$ for $y \in \{\pm 1\}$ up to epsilon accuracy. More formally,

Lemma 16. *Let the maximum two-hop degree of any visible node is at most d_2 and $\|b^{(1)}\|_\infty \leq B$. For $\delta > 0$ and $m = \Omega\left(n^2 \left(\frac{2}{(1-\tanh(\lambda))}\right)^{d_2+1} \log(n/\delta)/\epsilon^2\right)$ we have that with probability at least $1 - \delta$, Algorithm* DISTRIBUTIONFROMSTRUCTURE *given m samples and $\widehat{\mathcal{N}}(i) = \mathcal{N}(i)$ for every i returns unnormalized MRFs of $X|Y = y$ for $y \in \{\pm 1\}$ with coefficients $\hat{f}_S^{(y)}$ that are close to the coefficients of the true unnormalized MRFs $f_S^{(y)}$, that is,*

$$\sum_S |\hat{f}_S^{(y)} - f_S^{(y)}| \leq \epsilon.$$

Constructing the Predictor Observe that the joint distribution of X and Y can be represented as,

$$\Pr[X = x, Y = y] \propto \exp\left(\sum_S f_S^{(1)} x_S \mathbb{1}[y = 1] + \sum_S f_S^{(-1)} x_S \mathbb{1}[y = -1] + b^* y\right)$$

for some b^* and coefficients of the true unnormalized MRFs $f_S^{(y)}$ corresponding to conditioning of $Y = y$. This gives us,

$$\mathbb{E}[Y|X = x] = \tanh\left(\sum_S \frac{(f_S^{(1)} - f_S^{(-1)})}{2} x_S + b\right) \approx_\varepsilon \tanh\left(\sum_S \frac{(\hat{f}_S^{(1)} - \hat{f}_S^{(-1)})}{2} x_S + b\right)$$

Since we have estimates of $f_S^{(y)}$, to learn the predictor for Y we only need to find b^* which we can find by minimizing ℓ snce it is convex. Let $h_b = \sum_S \frac{(f_S^{(1)} - f_S^{(-1)})}{2} x_S + b$ and $\hat{h}_b = \sum_S \frac{(\hat{f}_S^{(1)} - \hat{f}_S^{(-1)})}{2} x_S + b$. We minimize $\hat{E}[\ell(h_b(X), Y)]$ over b and suppose the minimizer is $\hat{b}$. By Fact 1.3, $\ell(\hat{h}_b(X), Y) \leq \ell(h_b(X), Y) + 4\epsilon$. By Fact 1.4, h_{b^*} is the minimizer of the logistic loss. Then we have,

$$\hat{\mathbb{E}}[\ell(h_b(X), Y)] \leq \hat{\mathbb{E}}[\ell(\hat{h}_{b^*}(X), Y)] + 4\epsilon \leq \hat{\mathbb{E}}[\ell(h_{b^*}(X), Y)] + 8\epsilon.$$

Last we need a generalization bound that holds for our hypothesis class. For this we bound the Rademacher complexity (see [56] for more background) of the class of functions $\ell \circ \mathcal{H}$ where $\mathcal{H} := \{h_b || b| \leq B\}$.

$$\begin{aligned}
\mathcal{R}_m(\ell \circ \mathcal{H}) &\leq 2\mathcal{R}_m(\mathcal{H}) \\
&= \mathbb{E}_\sigma \left[\sum_{b||b|\leq B} \frac{1}{m} \sum_{i=1}^{m} \sigma_i h_b(x^{(i)}) \right] \\
&= \mathbb{E}_\sigma \left[\sum_{b||b|\leq B} \frac{1}{m} \sum_{i=1}^{m} \sigma_i \sum_S (f_S^{(1)} - f_S^{(-1)}) x_S + 2b \right] \\
&= 2\mathbb{E}_\sigma \left[\sum_{b||b|\leq B} \frac{1}{m} \sum_{i=1}^{m} \sigma_i b \right] \\
&= 2B\mathbb{E}_\sigma \left[\frac{1}{m} \left| \sum_{i=1}^{m} \sigma_i \right| \right] \\
&\leq \frac{2B}{\sqrt{m}}.
\end{aligned}$$

Here the first inequality follows from the contraction lemma (see [61]) and the last from standard properties of Radmeacher variables. Now applying Theorem 26.5 from [56] we get

$$|\mathbb{E}[\ell(h_b(X), Y)] - \mathbb{E}[\ell(\hat{h}_b(X), Y)]| \leq \frac{2B}{\sqrt{m}} + c\sqrt{\frac{\log(1/\delta)}{\sqrt{m}}}$$

where c is the maximum value of logistic loss by any hypothesis in the class. Observe that by Fact 1.4, logistic loss at h_{b^*} is bounded by a constant. Hence by Lipschitzness, we know that loss anywhere will be bounded by $O(\max(1, B))$. Therefore choosing $m \geq \Omega(B^2 \log(1/\delta)/\epsilon^2)$ suffices to get within ϵ. Combining this with before we get that the loss is within $O(\varepsilon)$ of the best loss.

Proof of Theorem 21 First, the algorithm runs LEARNSUPERVISEDRBMMBHD for each node to learn the structure of the induced RBM exactly with the given samples

$$m_1 = \exp(\lambda)^{\exp(O(\lambda))}(1/\alpha)^{O(1)}(1/\beta)^{O(1)} \log(n/\delta).$$

With the structure, we run DISTRIBUTIONFROMSTRUCTURE to learn both the induced RBMs for each conditioning of the label using $m_2 \geq \Omega\left(n^2 \left(\frac{2}{(1-\tanh(\lambda))}\right)^{d_2+1} \log(n/\delta)/\epsilon^2\right)$ samples where d_2 is the max 2-hop neighborhood size. Note that the dependence on λ is greater in m_1 than m_2. Subsequently, given the unnormalized mrfs, we run a simple optimization to find the bias term of the predictor using $m_3 \geq \Omega(B^2 \log(1/\delta)/\epsilon^2)$ samples. Combining the learnt mrf and the bias term, we get our hypothesis.

Remark 9. *If the model is not ferromagnetic, it is also possible and we expect it may be advantageous in some models to still use a similar indirect approach based on Bayes rule for learning a predictor of Y, but using the result of Theorem 1 instead of the greedy structure recovery method used in this section. The disadvantage of this approach is of course that its runtime for achieving structure recovery is slower.*

F Additional Experimental Data

Figure 3 contains samples generated from the model trained on MNIST images. For reference, we also include samples from the true MNIST and FashionMNIST training sets in the same format as Figure 2 and Figure 1.

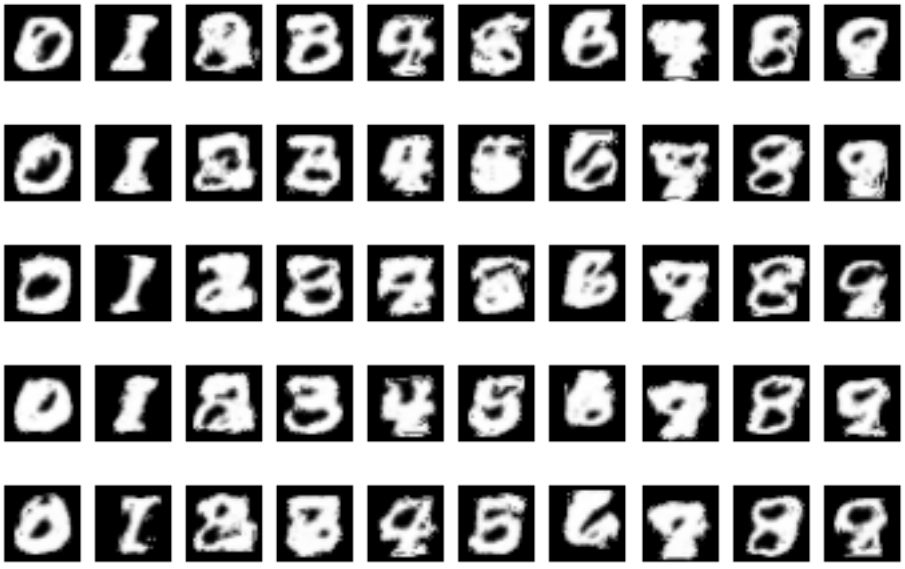

Figure 2: Five i.i.d. samples for each MNIST class, drawn from the trained model by Gibbs sampling.

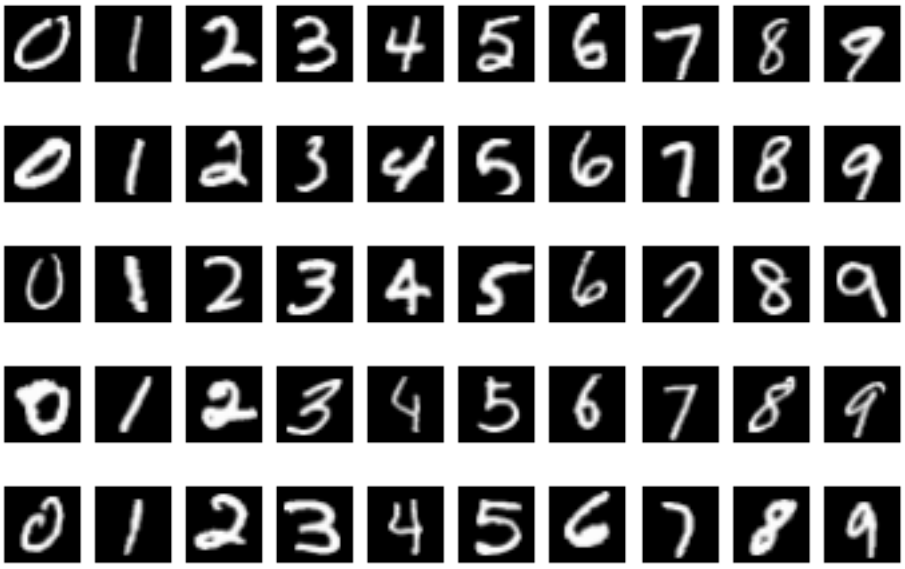

Figure 3: Reference MNIST images chosen randomly from training set.

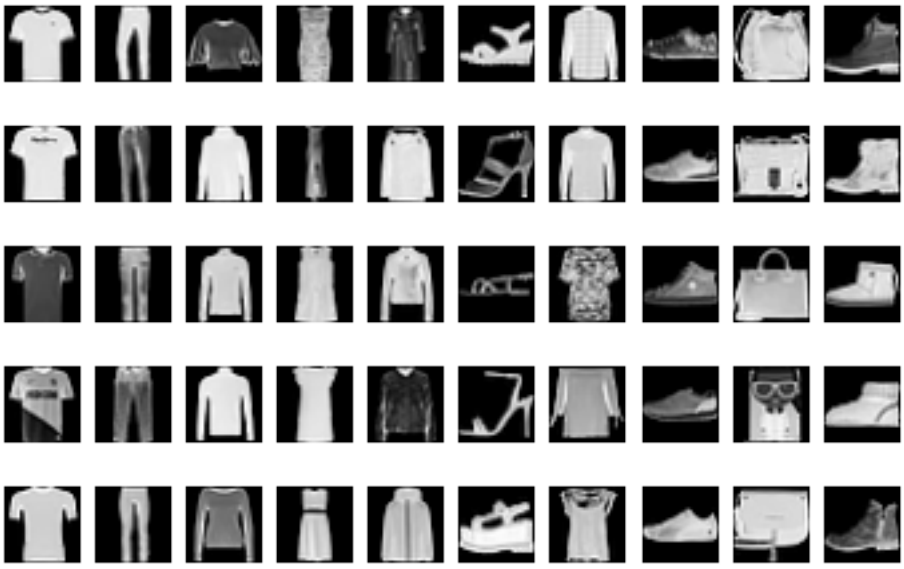

Figure 4: Reference FashionMNIST samples from training set.