A. Proof of Lemma 4.1

Proof. We will define θ so that for every x, y, $p_\theta(X_i = x_i|Y = y) = p_\lambda(X_i = x_i|H = y)$ and $p_\theta(Y = y) = p_\lambda(H = y)$.

Since the weight matrix W has dimension $d\times 1$ in this case, it is a vector, which we will denote as w. Recall that

$$p_\lambda(X_i = 1|H = y) = \sigma(a_i + w_i y),$$

hence we define

$$\psi_i \equiv \sigma(a_i + w_i)$$

and

$$\eta_i \equiv 1 - \sigma(a_i).$$

Finally, recall that

$$\begin{aligned} p_\lambda(H = 1) &= \frac{\sum_{x\in\{0,1\}^d} e^{-E_\lambda(x,1)}}{\sum_{x\in\{0,1\}^d,\, h\in\{0,1\}} e^{-E_\lambda(x,h)}} \\ &= \frac{\sum_{x\in\{0,1\}^d} e^{a^T x+b+x^T w}}{\sum_{x\in\{0,1\}^d,} e^{a^T x} + e^{a^T x+b+x^T w}}, \end{aligned}$$

where E_λ is the energy function given in equation (2), hence we set

$$\pi \equiv \frac{\sum_{x\in\{0,1\}^d} e^{a^T x+b+x^T w}}{\sum_{x\in\{0,1\}^d,} \left(e^{a^T x} + e^{a^T x+b+x^T w}\right)}. \tag{3}$$

To see that the map $\lambda \mapsto \theta$ is 1:1, note that a_i uniquely determines η_i, hence (a_i, w_i) uniquely determine (ψ_i, η_i). Lastly, rearranging equation (3) we get

$$\begin{aligned} &\pi \sum_{x\in\{0,1\}^d} \left(e^{a^T x} + e^{a^T x+b+w^T x}\right) = \sum_{x\in\{0,1\}^d} e^{a^T x+b+w^T x} \\ \Rightarrow &\pi \sum_{x\in\{0,1\}^d} e^{a^T x} = (1-\pi)e^b \sum_{x\in\{0,1\}^d} e^{a^T x+w^T x} \\ \Rightarrow &e^b = \frac{\pi}{1-\pi} \frac{\sum_{x\in\{0,1\}^d} e^{a^T x}}{\sum_{x\in\{0,1\}^d} e^{a^T x+w^T x}}, \end{aligned}$$

so that given (a, W), π is uniquely determined by b. Showing that the map $\lambda \mapsto \theta$ is a also subjective is straightforward. Hence it is a bijection. □

B. Proof of Lemma 4.2

Proof. Since $d \geq 3$ and for each i, X_i is not independent of Y, by Chang (1996), the parameter θ of the conditional independence model is identifiable. Since the map $\lambda \mapsto \theta$ in Lemma 4.1 is a bijection, there exists λ corresponding to θ, which is therefore identifiable as well. By the consistency property of the MLE (see, for example, (Casella & Berger, 2002)),

$$\lim_{n\to\infty} \hat{\lambda}_{\text{MLE}} = \lambda.$$

Since $p_\lambda(H = 1|X)$ is continuous in θ, one obtains

$$p_{\hat{\lambda}_{\text{MLE}}}(H = 1|X) \to p_\lambda(H = 1|X).$$

Finally, note that Lemma 4.1 implies, in particular, that under the map $\lambda \mapsto \theta$

$$p_\lambda(H = 1|X) = p_\theta(Y = 1|X),$$

which completes the proof. □

C. Stacking RBMs as a Variational Inference Procedure

Variational inference is a common approach to tackle complicated probability estimation problems (see, for example, (Bishop, 2006; Fox & Roberts, 2012), and a recent review (Blei et al., 2016)). Specifically, let p be a target probability distribution that we want to approximate. In variational inference we define a family of approximate distributions $\mathcal{D} = \{q_\alpha : \alpha \in \mathcal{A}\}$, and then perform optimization to find the member of $\mathcal{D}$ that is closest to p in Kullback-Leibler distance. A key idea is that the family $\mathcal{D}$ is flexible enough to contain a distribution close to p, yet simple enough to perform optimization over. For example, a popular choice is to take $\mathcal{D}$ as the collection of factorized distributions, i.e., of the form $q_\alpha(X) = \prod_i q_\alpha(X_i)$. In this section, we motivate the use of RBM-based DNN by considering a specific data generation model, and showing that training a stack of RBMs on data generated by this model is in fact a variational inference procedure.

The generative model we consider is a two layer Deep Belief Network (DBN), which played an important role in the emergence of deep learning in 2006 (Hinton et al., 2006). The DBN we consider generates data $Y \in \{0, 1\}$, $H \in \{0, 1\}^m$, $X \in \{0, 1\}^d$ via the probability distribution

$$p_\theta(X, H, Y) \equiv p_{\theta_1}(X, H)p_{\theta_2}(Y|H)$$

where X, H form a RBM (parametrized by θ_1).

We observe data $x^{(1)} \dots x^{(n)}$ from $p_\theta(X)$ and our goal is to estimate the posterior $p_\theta(y^{(i)}|x^{(i)})$ for $i = 1, \dots n$. The posterior can be written as

$$p_\theta(Y|X) = \mathbb{E}_{h\sim p_{\theta_1}(H|X)} P_{\theta_2}(Y|H = h).$$

Cueto et al. (2010) showed that as long as m is not too large comparing to d, RBMs are locally identifiable, i.e., identifiable up to order and flips of hidden units (Jason Morton, personal communication). Therefore, when training a RBM with m hidden units on $x^{(1)} \dots x^{(n)}$, by the consistency property of the MLE (Casella & Berger, 2002) the MLE $\hat{\theta}_{1\text{MLE}}$ will converge to the true parameter θ_1 as $n \to \infty$. Hence, when n is large enough, the vectors $h^{(i)}$

obtained from the (trained) RBM are in fact samples from $p_{\theta_1}(H|X = x^{(i)})$.

At the next step, the vectors $h^{(1)} \dots h^{(n)}$ are used to train a second RBM, with a single hidden node. Observe that in the data generation model considered in this section, $p_\theta(H|Y)$ does not factorize. The factorized distribution $p_\lambda(H|Y)$ that minimizes $\mathrm{KL}(p_{\theta_2}(H|Y)\|p_\lambda(H|Y))$ is given by

$$p_\lambda(H_i|Y) = p_{\theta_2}(H_i|Y)$$

Bishop (2006) (Chapter 10). By Lemma 4.1, we know that the distribution

$$p_\lambda(H, Y) = p_\theta(Y) \prod_i p_{\theta_2}(H_i|Y) \tag{4}$$

is equivalent to a RBM. Finally, by Lemma 4.2, the distribution (4) is consistently estimated by a RBM trained on vectors $h^{(1)} \dots h^{(n)}$, and is thus a variational inference procedure.

D. Stacking RBMs as an Approximation for a Directed Top-Down Model

Assume that the data is generated by a Markov chain $Y \to H \to X$, where $Y \in \{0, 1\}$, $H \in \{0, 1\}^m$, $X \in \{0, 1\}^d$. We further assume that the distributions $p_\theta(X|H)$, $p_\theta(H|Y)$ factorize, i.e.,

$$p_\theta(X|H) = \prod_{i=1}^{d} \Pr(X_i|H) \tag{5}$$

and

$$p_\theta(H|Y) = \prod_{i=1}^{m} \Pr(H_i|Y), \tag{6}$$

and are given by RBM-like conditional distributions, i.e.,

$$p_\theta(X_i = 1|H) = \sigma\left(a_i + W_{i,\cdot}H\right) \tag{7}$$

and

$$p_\theta(H_i = 1|Y) = \sigma\left(b_i + U_{i,\cdot}Y\right). \tag{8}$$

Hence the corresponding data generation probability is parametrized by $\theta = (\pi, a, b, W, U)$, where $\pi = \Pr(Y = 1)$.

This data generation process is depicted in Figure 10.

The posterior probabilities $p_\theta(Y|X)$ are given by

$$\begin{aligned} p_\theta(Y|X) &= \sum_{H \in \{0,1\}^m} p_\theta(Y|H) p_\theta(H|X) \\ &= \mathbb{E}_{h \sim p_\theta(H|X)} p_\theta(Y|H = h). \end{aligned}$$

By Section 4, we know that $p_\theta(H, Y)$ is equivalent to a RBM. Therefore, to accurately estimate the posterior, it suffices to approximate $p_\theta(H|X)$.

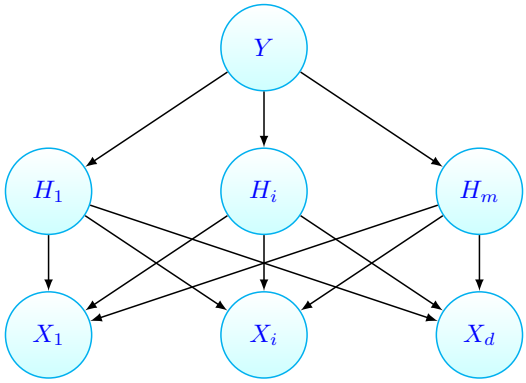

Figure 10. Data generated by a Markov Chain $Y \to H \to X$ with RBM-like conditional probabilities.

Under the data generation model described in Figure 10 and equations (5)-(8), it is evident that the joint distribution $p_\theta(X, H)$ cannot be parametrized as a RBM; indeed, $p_\theta(H|X)$ does not factorize. Hence, training a RBM on samples from $p_\theta(X)$, is a mean field approximation of $p_\theta(H|X)$. The form of $p_\theta(X, H)$ is shown in the following lemma.

Lemma D.1. *Under the data generation model described in Figure 10 and equations (5)-(8), the joint distribution $p_\theta(X, H)$ is given by*

$$p_\theta(X, H) = \exp\left(a^T X + X^T W H + b^T H\right) Z(H)$$

where

$$\begin{aligned} Z(H) = &\frac{1}{\sum_{X \in \{0,1\}^d} \exp\left(a^T X + X^T W H\right)} \\ &\times \sum_{Y \in \{0,1\}} \frac{p_\theta(Y) \exp(H^T U Y)}{\sum_{H'} \exp\left(b^T H' + H'^T U Y\right)} \end{aligned}$$

Proof. By definition,

$$\begin{aligned} p_\theta(X, H) &= \sum_{Y \in \{0,1\}} p_\theta(X, H, Y) \\ &= \sum_{Y \in \{0,1\}} p(Y) p_\theta(H|Y) p(X|H) \end{aligned} \tag{9}$$

Writing

$$p_\theta(X|H) = \frac{\exp\left(a^T X + X^T W H\right)}{\sum_{X' \in \{0,1\}^d} \exp\left(a^T X' + X'^T W H\right)}$$

and similarly

$$p_\theta(H|Y) = \frac{\exp\left(b^T H + H^T U Y\right)}{\sum_{H' \in \{0,1\}^m} \exp\left(b^T H' + H'^T U Y\right)},$$

we obtain

$$p_\theta(X|H)p_\theta(H|Y) = \frac{\exp\left(a^T X + X^T W H + b^T H + H^T U Y\right)}{\left(\sum_{X'} \exp\left(aTX' + X'^T W H\right)\right)\left(\sum_{H'} \exp\left(b^T H' + H'^T U Y\right)\right)}. \quad (10)$$

Plugging equation (10) in equation (9) we get

$$p_\theta(X,H) = \exp\left(a^T X + X^T W H + b^T H\right) \times \frac{1}{\sum_{X'} \exp\left(aTX' + X'^T W H\right)} \times \sum_{Y\in\{0,1\}} \frac{p_\theta(Y)\exp(H^T U Y)}{\sum_{H'} \exp\left(b^T H' + H'^T U Y\right)}$$

□

From lemma D.1 we see that $p_\theta(H|X)$ is close to be factorizable if $Z(H)$ is a approximately a log-linear function of H and $p_\theta(X)$ is approximately a log-linear function of X.

E. Datasets and Experimental Details

E.1. Simulated Dataset Generation Details

- **CondInd**: the label Y was sampled from a Bernoulli(0.5) distribution; The specificity η_i and sensitivity ψ_i of the variables $X_i,\ i = 1\ldots5$ were sampled uniformly from $[0.5, 1]$. The other ten X_i's were random guesses, i.e., had specificity = sensitivity = 0.5.
- **Tree15-3-1**: the label Y was sampled from a Bernoulli(0.5) distribution; each node in the intermediate and layer was generated from his parent with specificity and sensitivity sampled uniformly from $[0.8, 1]$, and in the bottom layer with specificity and sensitivity sampled uniformly from $[0.6, 1]$.
- **LayeredGraph15-5-5-1**: Data is generated from a Layered Graph with four layers of dimensions 1,5,5,15, starting at the true label Y. Each layer in the graph is generated from the above layer, and the graph has sparse connectivity (about 30% of the edges exist). For every node i and parent j we sample specificity ψ_{ij} and sensitivity η_{ij} uniformly. Finally, the value at each node was calculated as the weighted sum of the probabilities of the node being 1 given the values of the nodes in the preceding layer, normalized by the sum over the edges. The label Y was sampled from a Bernoulli(0.5) distribution.
- **TruncatedGaussian**: the label Y was sampled from a Bernoulli(0.5) distribution. One Gaussian had mean vector μ_1 were each of the 15 coordinates was sampled uniformly. The other Gaussian had mean vector $\mu_2 = -\mu_1$. Both Gaussians had identical covariance matrix, with off diagonal entries of 0.5 and diagonal entries of 1.

E.2. The Magic Datasets

An example for the correlation matrix of the 16 classifiers given the 0 class can be seen in Figure 12.

E.3. Hyper Parameters

In all experiments, we used stochastic gradient descent with minibatch size of 100. The hyper parameters we found important to tune were learning rate and the ℓ_2 penalty. In all our experiments we found that for both parameters, a value between 0.01-0.1 is satisfactory. The hyper parameters were tunes based on examination of the reconstruction error and free energies on a validation set.

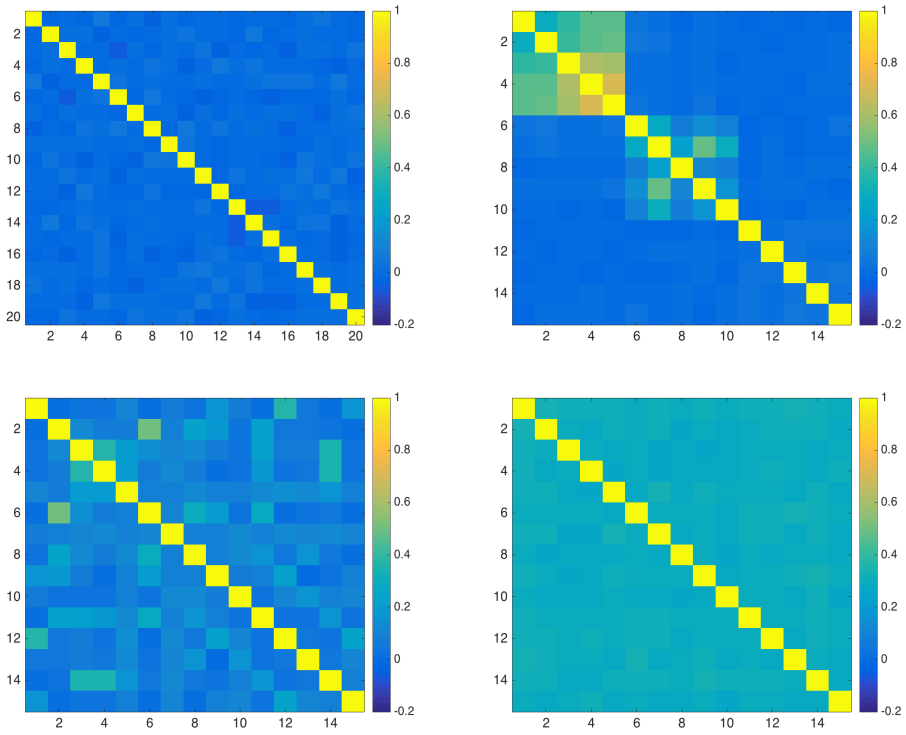

Figure 11. correlation matrices of the input data, for the $y = 0$ class in all four simulated datasets: condInd (top left), tree15-3-1 (top right), LayeredGraph (bottom left), TruncatedGaussian (bottom right).

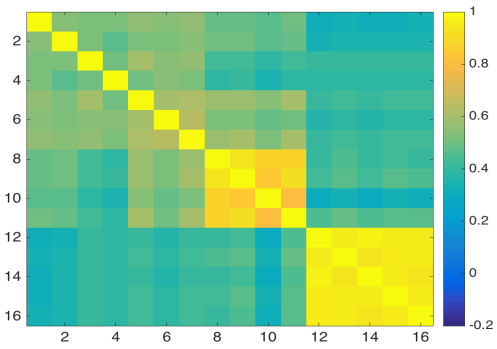

Figure 12. correlation matrix of the 16 classifiers in the Magic1 dataset, for the $y = 0$ class.