Mode-Assisted Unsupervised Learning of Restricted Boltzmann Machines

Haik Manukian, Yan Ru Pei, Sean R.B. Bearden, and Massimiliano Di Ventra

Department of Physics, University of California, San Diego, La Jolla, CA 92093

Restricted Boltzmann machines (RBMs) are a powerful class of generative models, but their training requires computing a gradient that, unlike supervised backpropagation on typical loss functions, is notoriously difficult to approximate. Here, we show that properly combining standard gradient updates with an off-gradient direction, constructed from samples of the RBM ground state (mode), improves their training dramatically over traditional gradient methods. This approach, which we call mode training, promotes faster training and stability, in addition to lower converged relative entropy (KL divergence). Along with the proofs of stability and convergence of this method, we also demonstrate its efficacy on synthetic datasets where we can compute KL divergences exactly, as well as on a larger machine learning standard, MNIST. The mode training we suggest is quite versatile, as it can be applied in conjunction with any given gradient method, and is easily extended to more general energy-based neural network structures such as deep, convolutional and unrestricted Boltzmann machines.

Boltzmann machines and their restricted version (RBMs), are generative models applied to a variety of machine learning problems. They enjoy a universal approximation theorem for discrete probability distributions, are used as building blocks for deep-belief networks, and, in no small feat, can even represent correlated states in quantum many-body systems.

Training RBMs is typically formulated as a gradient descent in the Kullbach-Leibler (KL) divergence between the data distribution defined by a dataset, and the RBM model distribution, parameterized by a set of weights and biases. This unsupervised procedure results in a computationally intractable expectation value popularly approximated by a Markov Chain Monte Carlo (MCMC) procedure dubbed “contrastive divergence” (CD). This approach faces difficulty when the model distribution represented by the RBM contains peaks of probability far away from the elements of the dataset, resulting in “spurious modes” that trap the Markov chain. The limitations of CD, the de facto algorithm for training RBMs, combined with the rapid advances in supervised learning approaches, has led to the sideline of their unsupervised learning, known also as “pretraining”, in favor of supervised backpropagation from random initial conditions.

However, many state-of-the-art neural networks have been shown to be vulnerable to what are called adversarial examples, or slight perturbations of the input that “fool” the network. In fact, one of the most popular supervised learning techniques, batch normalization, was found to contribute to this phenomenon. It is known that pretraining can be a strong regularizer resulting in better generalization for supervised models, and an improvement in their unsupervised training could lead to more robust performance in a downstream task. This motivates the search for better unsupervised training methods.

In a recent work, a memcomputing-assisted training scheme for RBM was proposed to address this unsupervised training difficulty. Memcomputing is a novel computing paradigm in which memory (time non-locality) assists in the computation of hard computational decisions and optimization problems.

In the algorithm of Ref. 11, the difficult model expectation term in the log-likelihood gradient was replaced by a sample of the mode of the RBM’s probability distribution obtained from a memcomputing solver, which led to better quality solutions versus CD in a downstream classification task. However, despite demonstrating a significant reduction in the number of pretraining iterations necessary to achieve a given level of classification accuracy, as well as a total performance gain over CD, the algorithm of Ref. 11 does not fully exploit samples of the mode, in particular it does not give rise to training advantages over standard methods in the unsupervised setting. Additionally, in the present work, we introduce i) a principled schedule for incorporating samples of the RBM ground state into pre-training, ii) an appropriate mode-driven learning rate, iii) comparisons to other state-of-the-art unsupervised pre-training approaches without the need of supervised fine-tuning, and iv) proofs of advantageous properties of the method.

We show that by appropriately combining the RBM’s mode (ground state) samples and data initiated chains (as in CD) not only improves considerably the model quality over CD and other MCMC procedures, but also improves the stability of the pretraining routine. This mode training utilizes both the dataset (as in CD) and samples of the mode of the RBM model distribution in the training process to “push down” spurious modes of the model, whenever they appear.

Superficially, this method resembles ‘mode hopping’ MCMC proposed in recent literature, where local maxima are either found with some optimization method or assumed to be known beforehand (via a dataset). However, a crucial difference between our mode training for RBMs and mode hopping algorithms is that we do not use the modal configuration to initiate a new MCMC update to improve the mixing rate. Instead, the mode itself is used to inform the weight updates directly. The difference is substantial. In fact, since higher energy states are exponentially suppressed, exposing the Markov chain to the mode will most likely get it stuck there, which requires ad hoc constructions to recover detailed balance. Our mode-training method does not suffer from these drawbacks and is thus a more computation-ally efficient way to utilize the mode to train RBMs. Furthermore, we show that under a sufficiently large
learning rate, sampling the global mode alone is capable of exploring efficiently a multi-modal energy landscape.

To realize this method in practice, one must supplement standard gradient updates with updates constructed from samples of the ground state of the RBM. Finding this ground state is equivalent to a Quadratic Unconstrained Binary Optimization (QUBO) problem, known to be NP-hard [17]. Therefore, although we can compute the ground state of RBMs exactly for small datasets, for efficient mode sampling in realistically sized cases, we employ a memcomputing solver in efficiently sampling the ground states of similar non-convex landscapes [18,19]. The details of our implementation, including computational complexity and energy comparisons with MCMC, can be found in the appendix that accompanies this work. However, in principle, one could use other optimizers for mode training.

To corroborate our method, we will show exact KL/log-likelihood achieved on small synthetic datasets and on the MNIST dataset. In all cases, we find that mode training is able to learn more accurate models than several other training methods such as CD, persistent contrastive divergence (PCD), parallel tempering (PT) used in tandem with enhanced gradient RBMs (E-RBMs), and centered RBMs (C-RBMs), as reported in Ref. [20].

The paper is organized as follows. In Section II we introduce RBMs and the standard unsupervised training procedures, and identify their main weaknesses. Section III introduces our mode-training method and its main features. Section IV contains the results of our numerical experiments. Finally, we offer our thoughts for future work in Section V.

I. TRAINING RBMS WITH MCMC

RBMs are undirected graphical models with a bipartite structure that differentiates between $n$ visible nodes, $v \in \{0,1\}^n$ and a set of $m$ latent, or ‘hidden’ nodes, $h \in \{0,1\}^m$, not directly constrained by the data $\mathcal{D}$.

These states are usually taken to be binary but can be generalized. Each state of the machine corresponds to an energy of the form

$$E(v, h) = -a^T v - b^T h - v^T Wh,$$

where the biases $a \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and weights $W \in \mathbb{R}^{n \times m}$ are the learnable parameters. Note that an RBM does not allow connections within a layer. This defines a distribution over states given by a Boltzmann-Gibbs distribution,

$$p(v, h) = \frac{e^{-E(v, h)}}{Z}.$$

The normalizing factor, $Z = \sum_v \sum_h e^{-E(v, h)}$, is the formidable partition function that involves the sum over an exponentially scaling number of states, thus making the exact computation of its value infeasible. Additionally, the bipartite structure of the RBM connectivity implies that the hidden nodes are conditionally independent given any visible nodes (and vice versa), with a closed form conditional distribution given by $p(h \mid v) = \sigma(\sum_i w_i v_i + b_j)$, where $\sigma(x) = (1 + e^{-x})^{-1}$.

We indicate the unique elements of the dataset for training and testing of the network as $\mathcal{D} = \{v_1, \cdots, v_m\} \subset \Omega$, where $\Omega = \{0,1\}^n$ is the space of all binary sequences of length $n$. We can then write the data distribution as

$$q(v) = \sum_{v_i \in \mathcal{D}} c_i \delta(v - v_i),$$

where $\delta$ is the delta function that evaluates to 1 if $v = v_i$ and 0 otherwise. This effectively defines a probability mass function (PMF) over $\Omega$ with non-zero values only for values $v_i \in \mathcal{D}$. We then call $\mathcal{D}$ the support of $q$.

Let us assume further that all data points have equal amplitude over the support, i.e., $c_i = 1/|\mathcal{D}|$. Since most real world datasets consist of unique elements with no exact repeats, this class of distributions includes all relevant ones. However, we will see in Sec. III that our method seems to work equally well also for non-uniform distributions.

Training an RBM then amounts to a search for the appropriate weights and biases, $\theta = \{W, a, b\}$, that will minimize the quantity

$$\text{KL}(q \mid \pi) = \sum_{v} q(v) \log \frac{q(v)}{p(v)}.$$

This is known as the Kullback-Leibler (KL) divergence between the data distribution, $q(v)$, and the model distribution of the RBM over the visible layer, $p(v) = \sum_h p(v, h)$, with hidden nodes traced out. The latter can be written as,

$$p(v) = \frac{1}{Z} \prod_{i=1}^n e^{a_i v_i} \prod_{j=1}^m \left(1 + e^{b_j + \sum_{i=1}^n w_{ij} v_i v_i'}\right).$$

The optimization of Eq. (4) is typically done with gradient descent over the KL divergence which leads to weight updates of the form [21]

$$\Delta w_{ij} \propto [\langle v_i h_j \rangle_q(p(h|v)) - \langle v_i h_j \rangle_p(v|h)] .$$

The first term on the rhs of Eq. (5) is an expectation with the hidden nodes driven by the data, and hence is referred to as the data term. Since the conditional distributions across the hidden nodes are factorial and known in closed form, this inference problem is easy in the RBM case. The second term on the rhs of Eq. (6) instead, is an expectation over the model distribution with no nodes fixed, and called the model term. The exact calculation of this term requires computing the partition function of the RBM, which is proved to be hard even to estimate [22]. It is this term that MCMC algorithms (including CD) attempt to approximate.

A. Contrastive Divergence

A popular method to training RBMs is CD, which is a special case of an MCMC method known as Gibbs
sampling. The Markov chain is initialized from a point in the dataset, \( v \), then the hidden and visible nodes are sequentially re-sampled a \( k \) number of times. A distorted model expectation is then computed from the \emph{reconstructed} \( v^k \). In practice, choosing some finite \( k \) introduces a bias, but empirically it is found that using \( k = 1 \) gives a sufficient signal for learning.

Since the CD chain starts from a point in the dataset (i.e., a sample from the data distribution), difficulties arise when the model distribution represented by the RBM contains modes at points where the data distribution has negligible probability. CD will have a hard time finding and hence pushing down these spurious modes. This, coupled with the prohibitively slow mixing of this MCMC method due to random-walk exploration, and typical high dimensionality of the problem, renders CD not a particularly effective method for unsupervised learning.

II. MODE INFORMED WEIGHT UPDATES

After these preliminaries we can now describe our mode-training method. In a nutshell, it consists in replacing the average in the model term of Eq. (6) with the mode of \( p(v) \) at appropriate steps of the training procedure. However, \( p(v) \) is very cumbersome to compute (see Eq. (5)), thereby adding a considerable computational burden. Instead, we sample the mode of \( p(v, h) \), the model distribution of the RBM.

The rationale for replacing the mode, \( v^* \), of \( p(v) \) with the visible configuration of the mode, \( v^* \), of \( p(v, h) \) is because the two modes are equivalent with high probability under scenarios typical for different stages of the RBM pre-training. We prove this \emph{rigorously} in the appendix, while here we provide numerical evidence of this fact.

A. Mode Correspondence of Joint and Marginal PMF

To illustrate the equivalence between the modes of \( p(v) \) and \( p(v, h) \), let us begin by expressing the joint PMF in terms of the product of the marginal PMF over the visible layer and the conditional PMF over the hidden layer \( p(v, h) = p(v)p(h|v) \). For any given visible configuration \( v \), we then have \( \arg \max_h p(v, h) = p(v) \arg \max_h p(h|v) \). We can then define the hidden “activation” of \( v \) to be

\[
r(v) = \max_h p(h|v),
\]

which allows us to write \( \max_v p(v, h) = p(v)r(v) \). Note that we can interpret \( r(v) \) as a measure of the “certainty” that the hidden nodes acquire the value \( 0 \) or \( 1 \).

It is then clear that we can write the probability amplitude of the mode of the joint PMF as \( \max_{(v, h)} p(v, h) = \max_v (\max_h p(v, h)) = \max_v (p(v)r(v)) \leq \max_v (p(v)) = p(v^+) \), where we have used the fact that \( r(v) \leq 1 \) and \( v^+ \) is the mode of the marginal PMF, \( p(v) \). If \( r(v^+) = 1 \) then we have modal equivalence of the joint and marginal PMFs.

FIG. 1. The maximal conditional probability distribution of the hidden layer, \( r(v^+) \), when driven by \( v^+ \), the mode of the marginal PMF, \( p(v) \), as a function of CD-1 training iterations. The results are averaged over an ensemble of 200 randomly initialized \( 15 \times 10 \) RBMs, with \( \pm 1 \sigma \) error bars (the shaded regions), on a shifting bar synthetic dataset. The same calculation conditioned on a random visible configuration is plotted as a baseline for comparison.

In Fig. 1 we plot the evolution of \( r(v^+) \) as a function of the number of CD-1 training iterations for a shifting bar synthetic dataset, which is small enough that we can compute the exact mode of \( p(v) \) at any iteration. The figure indeed shows that \( r(v^+) \) approaches 1 rather quickly as pre-training proceeds. The activation of a random visible configuration is being used as comparison.

In the appendix we also prove that the condition of \( r(v^+) \) being close to 1 is not necessary for establishing modal equivalence. In fact, we prove that it is still possible for the two modes to be equal even when the weights are small (thus a smaller \( r(v^+) \) value). Additionally, we show in the appendix that mode training is more effective in exploring the PMF of the model distribution for RBM instances of greater frustration. The latter is a measure of the degeneracy of the low-energy states of an RBM, and thus the difficulty of finding the ground state configuration. Since it was shown that the frustration of the RBM increases as pre-training proceeds, in order to effectively utilize the power of mode training, the frequency of mode updates should be higher at the later stages of the training than the earlier stages.

B. Optimal Mode-Training Schedule

The results from the previous subsection then suggest a schedule for the mode training routine that performs mode updates more frequently the longer the pre-training routine has elapsed.

To realize this, we use a sigmoid, \( \sigma \), to calculate the probability of replacing the data driven hidden CD term with a mode driven term at the iteration step \( n \)

\[
P_{\text{mode}}(n) = P_{\text{max}} \sigma(\alpha n + \beta).
\]
Here, $0 < P_{\text{max}} \leq 1$ is the maximum probability of employing a mode update, and $\alpha$ and $\beta$ are parameters that control how the mode updates are introduced into the pre-training. They are chosen such that the frequency of mode updates becomes dominant only when both the conditions of large weights and frustration are met (see Sec. III for the value of these parameters). Initially, $P_{\text{mode}}$ will be small, since the joint- and marginal-distribution modes are unequal, and gradually rises to its maximal value when the modes are of equal magnitude. Note that one may employ different functions to quantify the degree to which the joint- and marginal-distribution modes equalize during training. However, we have found that the sigmoid works well enough in practice.

C. Combining MCMC with mode updates

We are now ready to outline the full procedure of mode training, that combines a MCMC method with the mode updates following the schedule $[8]$. Although one may choose any variation of the MCMC method to train RBMs, for definiteness of discussion, we consider here the standard training method, CD$^2$. In this case, weight updates follow the modified KL(q||p) gradient. As discussed in Section I, it evaluates to a difference of two expectations called the data term and model term which we can write as

$$\Delta w_{ij}^{\text{CD}} = \epsilon_{\text{CD}} \left( \langle v_i h_j \rangle_{q(v)} p(h|v) - \langle v_i h_j \rangle_{p(v,h)} \right),$$

where $\epsilon_{\text{CD}}$ is the CD update learning rate, and the expectation in the second term is taken over the reconstructed distribution over a Markov chain initialized from the dataset after $k$ Gibbs samples ($k = 1$ in most cases). When driving the weights with samples of the RBM ground state with the schedule $[8]$, we use instead the following update,

$$\Delta w_{ij}^{\text{mode}} = \epsilon_{\text{mode}} \left( \langle v_i h_j \rangle_{q(v)} p(h|v) - \langle v_i h_j \rangle_{p(v,h)} \right),$$

where $\langle \rangle_p$ is the mode of the joint RBM model distribution. Note that the mode update learning rate, $\epsilon_{\text{mode}}$, may be different from the CD learning rate, $\epsilon_{\text{CD}}$.

We also stress that the updates in Eq. (10) are in an off-gradient direction. As we show now, this is the reason for the increased stability of the training over MCMC approaches, and its convergence to arbitrarily small KL divergences.

D. Stability and Convergence

The data term, which is identical in both Eq. (9) and Eq. (10), tends to increase the weights associated with the visible node configurations in the dataset, thereby increasing their relative probabilities compared to states not in the support set, $v \in \Omega D$. Instead, the model term decreases the weights/probability corresponding to highly-probable model states. CD does this poorly and often diverges, while mode training achieves this with better stability and faster convergence (see Fig. 2). We provide here an intuitive explanation of this phenomenon, while a formal treatment on this topic will be provided in the appendix.

The pre-training routine can be broken down in two phases. In the first phase, the training procedure attempts to discover the support $D$ of the data distribution $q(v)$. We call this phase the discovery phase. To better see this, consider a randomly initialized RBM with small weights. These small and uncorrelated weights give rise to RBM energies close to zero for all nodal states, or $E(v,h) \approx 0$ for all $v$ and $h$, see Eq. (1). This results in the model distribution $p(v,h)$ being almost uniform.

Therefore, we see that in the discovery phase of training, the model term plays little role in the training as it simply pushes down on the weights in a practically uniform manner, with $\langle v_i h_j \rangle_M \approx 0.25$. On the other hand, the data term drives the initial phase of the training by increasing the marginal probability of the visible states in the support, $v \in D$. We can then employ a large learning rate (say, $\epsilon_{\text{CD}} = 1$) in the beginning of the training, driving the visible layer configurations in the dataset, $D$, to high probability versus configurations outside the support. Empirically, we find that CD training performs in the discovery phase reasonably well, and is quickly able to “find” the visible states in the support.

Now, having discovered the support, we arrive at the second phase of the training where we have to bring the model distribution as close to uniformity as possible over the support in order to minimize the KL-divergence. We call this phase the matching phase of the training, where we bring the model distribution as close to the data distribution as possible. CD usually performs poorly in this phase (see Fig. 2). To see this most directly, we simply have to consider a visible state with a slightly larger probability than the other states. It should then be necessary for the model term to locate and “push down” on this state to increase the uniformity of the distribution over the support. However, for any CD approximation of the model term, this rarely happens in a timely manner as the mixing rate of the MCMC chain is far too slow to locate this state before the training diverges.

This is where samples of the mode are most effective, and can assist in the correction of the states’ amplitudes. As we have discussed in Sec. II A, finding the modal state, $v^*$, of the model distribution, $p(v,h)$, allows us to immediately locate the mode, $v^+$, of the marginal probability, $p(v)$, and “push” down on this state through an iteration of weight updates. This “push” may result in another state “popping” up and becoming the new modal state; however, often times the probability amplitude of this new state will be less than that of the previous mode (see also the appendix). This results in a training routine that “cycles” through any dominant state that emerges at each iteration, and the probability amplitude of the mode decreases as training proceeds until the proba-
FIG. 2. Median KL divergences +max/-min KL (top row) and converged logarithmic differences between model and data distributions (bottom row) of 25 randomly generated $6 \times 6$ RBM on a random uniform support set of size $n_d = 10$ for CD-1 (left column) and mode training (right column). In both cases, the learning rate was a constant $\epsilon_{CD} = 0.05$ for 100,000 iterations. The mode sampling probability, $P_{\text{mode}}$, is plotted as the dotted line in the top right.

probability amplitudes of all the states in the support become equal (see the formal demonstration of this in the appendix), which results in the desired uniform distribution over the support. This can be visualized as a “seesaw” between the dominant states, with the oscillation amplitude of this seesaw decaying to zero in time.

We outline the pseudo-code for mode training in Algorithm 1 and a visual depiction of the training side by side with CD-1 is shown in Fig. 2.

As it should now be clearer, these mode-driven updates are deviations from the gradient direction, since in general the mode over the model distribution is different from the expected value. This makes the mode-training algorithm, which mixes mode driven samples and data-driven ones, distinct from gradient descent. This is also supported by the fact that our method tends toward a particular class of distributions (uniform), when gradient descent would settle in some local minima or saddle points in the KL landscape.

Algorithm 1 Unsupervised learning of an RBM with the mode-training algorithm

1: procedure $\text{MT}(P_{\text{max}}, \alpha, \beta, (\epsilon_{CD})_{n=1}^{N})$
2: $\theta_0 \sim N(0, 0.01)$
3: for $i = 1; i \leq N; i++$ do
4: $p_{\text{mode}} \leftarrow P_{\text{max}} \sigma(\alpha_i + \beta)$
5: Sample $u \sim U[0, 1]$
6: if $u \leq p_{\text{mode}}$ then
7: $v^*, h^*, E_0 \leftarrow \text{argmin}E(v, h)$
8: $\gamma \leftarrow \frac{E_0}{(n+1)(m+1)}$
9: $\theta_i \leftarrow \theta_{i-1} + \gamma \epsilon_{CD} \Delta \theta_{\text{mode}}$ \\> Eq. 10
10: else
11: $\theta_i \leftarrow \theta_{i-1} + \epsilon_{CD} \Delta \theta_{\text{CD}}$ \\> Eq. 9
12: end if
13: end for
14: return $\theta_N$
15: end procedure

The free parameters in this method are the schedules of the mode sample using $P_{\text{mode}}(n)$ (defined by $P_{\text{max}}, \alpha$ and $\beta$ in Eq. (8)) and the CD learning rate, $\epsilon_{CD}$. With $\epsilon_{CD}$ fixed, we set $\epsilon_{\text{mode}} = \gamma \epsilon_{CD}$, where $\gamma = -E_0/[(n+1)(m+1)]$, with $E_0(<0)$ being the ground state of the corresponding RBM with nodal values $\{-1, 1\}^{n+m}$. This particular choice of $\gamma$ is an upper bound to the learning rate which minimizes the RBM energy variance over all states (see the appendix for the proof of this statement).

Finally, we find that the mode training method is not very sensitive to the parameters chosen. In fact, as long as the mode samples are incorporated after the joint and marginal mode equilibration, the training is stabilized and the learned distribution will tend
to uniformity (see also the appendix). This result reinforces the intuitive notion that the pushes on the mode provide a stabilizing quality to the training over CD (or any other MCMC approach), which can otherwise diverge when mixing rates grow too large at later times during training.

E. Importance of Representability

Note that since mode training is driven to distributions of a particular form, instead of local minima as in the case of CD or other gradient approaches, the representability of the RBM becomes important. The ability of a RBM to represent a given data distribution is given by the amount of hidden nodes, where one is guaranteed universal representability with \( n_d + 1 \) hidden nodes. In other words, one more hidden node than the number of visible configurations with non-zero probability is sufficient (but perhaps not necessary) for general representability. In practice, this bound is found to be very conservative and typically much fewer nodes are needed for a reasonable solution.

Representability can become an issue in mode training when the parameter space of the RBM does not include the uniform distribution over the support (or a reasonable approximation). Since the mode training is generally in a non-gradient direction, this means that it may settle to a worse solution than a local optimum obtainable by CD. This is a signal that more hidden nodes are required for an optimal solution.

Since most natural datasets live on a very small dimensional manifold of the total visible phase space, \( |n_d|/\Omega \ll 1 \), the amount of hidden nodes required typically scales polynomially with the size of the problem, versus the exponential scaling of the visible phase space. This makes representability not an insurmountable problem for mode training, even in full size problems. To this end, the examples of Fig. 2 and Fig. 3 show that mode training does not necessarily fail if the number of hidden nodes is less than that needed to guarantee representability.

III. RESULTS

As examples of our method, we have computed the log-likelihoods achieved with mode training across two synthetic and one realistic (MNIST) datasets, and compared the results against the best achieved log-likelihoods with CD-1, PCD-1 and PT on standard RBMs, E-RBMs, and C-RBMs. For the small synthetic datasets we could also compute the exact log-likelihoods, thus providing an even stronger comparison. For the larger MNIST case, mode sampling was done via simulation of a digital memcomputing machine based on Ref. 25. The specific details of our implementation can be found in the appendix.

For synthetic data, we use the commonly employed binary shifting bar and bars and stripes datasets, both described in Ref. 26. The former is defined by two parameters: the total length of the vector, \( L \), and the amount of consecutive elements (with periodic boundary conditions), \( B < L \), set to one, with the rest set to zero. This results in \( L \) unique elements in the dataset with uniform probability, giving a maximum likelihood of \( L \log(1/L) \). The inverted shifting bar set is obtained by swapping ones and zeros. The bars and stripes dataset is constructed by setting each row of a \( D \times D \) binary pattern to one with probability \( 1/2 \), and then rotating the resulting pattern \( 90^\circ \) with probability \( 1/2 \). This produces \( 2^{D+1} \) elements, with the all-zero and all-one patterns being twice as likely as the others.

For a direct comparison to previous work, we followed the same setup as Ref. 20. A 9×4 RBM was tested on a shifting bar dataset with \( L = 9, B = 1 \) and a \( D = 3 \) bars and stripes dataset. Both synthetic sets were trained for 50,000 parameter updates, with no mini-batching, and a constant learning rate. For the MNIST dataset, a 784×16 sized model was trained for 100 epochs, with batch sizes of 100. The mode samples in both cases are slowly incorporated into training in a probabilistic way following Eq. (8), initially with \( P_{\text{mode}} = 0 \) and driven to \( P_{\text{max}} = 0.1 \) for the shifting bar and MNIST datasets, and \( P_{\text{max}} = 0.05 \) for the bars and stripes dataset. In both cases, we chose \( \alpha = 20/N \) and \( \beta = -6 \), where \( N \) is the total number of parameter updates.

We plot an example of training progress in a moderately large synthetic problem in Fig. 5. Reported is the KL divergence (which differs from the log-likelihood by a constant factor independent of the RBM parameters) of a slightly bigger 14×10 RBM as a function of number of parameter updates on a \( L = 14, B = 7 \) shifting bar set, for both CD-1 and mode training. We consider two learning rate schedules, constant \( \epsilon_{\text{CD}} = 0.05 \) and exponential decay \( \epsilon_{\text{CD}}(n) = e^{-cn}, c = 4, n \in [0, 1], \) the fraction of completed training iterations. Additionally, every time a mode sample is taken, CD is allowed to run with \( k = 720 \), a number scaled to the equivalent computational cost of taking a mode sample. The details of the computational equivalence between a mode sample using memcomputing and iterations of CD are discussed in greater detail in the appendix. In both cases, even when computational cost is factored in, mode training leads to better solutions and proceeds in a much more stable way across runs (lower KL variance at convergence). Importantly, mode training never diverges while CD oftentimes does. Following our intuition about mode training established in Sec. 1 using larger learning rates in the CD-dominated phase accelerates the convergence of mode training.

It is known that using CD to train RBMs can result in poor models for the data distribution for which PCD and PT are recommended. We note that for the mode training employed in this paper, CD-1 was employed as the gradient approximation (except in the case for MNIST where PCD-1 was used). Impressively, in all cases tested, the mode samples were able to stabilize the CD algorithm sufficiently to overcome the other, more involved approximations (PT) and model enhancements (centering).

In addition, it is clear that mode training exhibits
FIG. 3. KL divergences achieved on the binary shifting bar dataset across 25 randomly initialized 14×10 RBMs for both CD-1 and mode training (MT). In addition, every time a mode sample is taken, CD is allowed to run with $k = 720$, a number scaled to the equivalent computational cost of taking a mode sample (see text and appendix). The bold line represents the median KL divergence across the runs, and the max/min KL divergences achieved at that training iteration define the shaded area. The plot in the top panel is with a small CD learning rate, $\epsilon_{CD} = 0.05$. The plot in the bottom panel is with an exponentially decaying $\epsilon_{CD}(n) = e^{-cn}$ with $c = 4$ and $n \in [0, 1]$ being the fraction of completed training iterations.

several desirable properties over CD (or other gradient approaches). Most significantly, it seems to perform better with larger learning rates during the gradient dominated phase, and smaller learning rates when using mode samples. CD and other gradient methods generally perform better with smaller learning rates, as their approximation to the exact gradient gets better. Irrespective, even in this regime, the mode training eventually drives the system to the uniform solution compared to the local optimum of CD. The main advantage is that with mode training, one can (and often should) use larger learning rates, resulting in fewer required iterations.

For further comparison, we report in Table I results for the shifting and inverted bar, bars and stripes, and MNIST datasets obtained with mode training and those reported in Ref. 20. The results show mode training with a standard RBM always converges to models with log-likelihoods higher than E-RBMs, and C-RBMs trained with CD-1, PCD-1, or PT. Furthermore, the mode training log-likelihood increases with an increasing number of hidden nodes (better representability). Empirically, we also find the incredible result that with sufficient representability and the proper learning rate, mode training can find solutions arbitrarily close to the exact distribution.

| S. Bar | Inv. S. Bar | Bars & Stripes | MNIST |
|--------|-------------|----------------|-------|
| CD-1   | -20.42      | -20.73         | -61.08 | -152.42 |
| PCD-1  | -21.71      | -21.64         | -57.01 | -140.43 |
| PT     | -20.57      | -20.57         | -51.99 | -142.00 |
| MT     | -19.85      | -19.86         | -50.79 | -136.42 |
| Exact  | -19.77      | -19.77         | -41.59 | –       |

TABLE I. Comparison between the best log-likelihoods achieved over 50,000 gradient updates on a 9×4 RBM across various RBM types (standard, E-RBM, C-RBM) and training techniques (CD-1, PCD-1, PT) as reported in Ref. 20 compared with mode training (MT) on a standard RBM. For each technique, the best achieved log-likelihood score across 25 runs is reported. In parenthesis are results for a 9×9 RBM. For these small datasets we can also compare with the exact result. For MNIST, networks trained had 16 hidden nodes and PCD-1 was used as the gradient update, and average log-likelihood is reported.

IV. CONCLUSIONS

In this paper we have introduced an unsupervised non-gradient training method that stabilizes gradient based methods by utilizing samples of the ground state of the RBM, and is empirically seen to get as close as desired to a given uniform data distribution. It relies on the realization that as training proceeds, the RBM becomes increasingly frustrated, leading to the modes of the visible layer distribution and joint model distribution becoming effectively equal. As a consequence, by using the mode (or ground state) of the RBM during training, our approach is able to “flatten” all modes in the model distribution that do not correspond to modes in the data distribution, reaching a steady state only when all modes are of equal magnitude. In this sense, the ground state of the RBM can be thought of as ‘supervising’ the gradient approximation during training, preventing any pathological evolution of the model distribution.

Our results are valid if the representability of the RBM is enough to include good approximations of the data distribution. Once the representability is sufficient, a properly annealed learning-rate schedule will take the KL divergence as low as desired. Increasing the number of hidden nodes increases the non-convexity of the KL-divergence landscape, easily trapping standard algorithms in sub-optimal states. In practice, after some point, increasing the number of hidden nodes will not decrease the KL divergence...
that a pre-training procedure actually converges to, as the trade-off between effective gradient update and representation quality is reached. We here claim that this point of tradeoff for our mode-assisted procedure is reached at far greater number of nodes than standard procedures, thus allowing us to find representations with far smaller KL-divergence. The mode training we suggest then provides an extremely powerful tool for unsupervised learning, which i) prevents a divergence of the model, ii) promotes a more stable learning, and iii) for data distributions uniform across some support, can lead to solutions with arbitrary high quality.

To scale our approach, one would need an efficient way to sample low-energy configurations of the RBM, a difficult optimization problem on its own. This is equivalent to a weighted MAX-SAT problem, for which there are several industrial-scale solvers available. Also, the recent successes of memcomputing on these kind of energy landscapes in large cases (million of variables) are fodder for optimism.[15][19]

Finally, fitting general discrete distributions (with modes of different height) with this technique seems also within reach. In this respect, we can point to our results on the bars and stripes dataset (a non-uniform $q(v)$) for inspiration. We have found the best log-likelihood on that set with mode training with a lower frequency of the mode sampling, $P_{\text{max}} = 0.1 \rightarrow 0.05$, compared to the shifting bar (a uniform set). This suggests that a general update, which properly weighs the mode sample in combination with the dataset samples, may extend this technique to general non-uniform probabilities, with the weight analogous to a tunable demand for uniformity.

Our method is useful from a number of perspectives. First, from the unsupervised learning point of view, it opens the door to the training of RBMs with unprecedented accuracy in a novel, non-gradient approach. Second, many unsupervised models are used as ‘feature learners’ in a downstream supervised training task (e.g., classification), where the unsupervised learning is referred to as pre-training. We suspect that starting backpropagation from an initial condition obtained through mode training would be highly advantageous. Third, the mode training we suggest can be done on models with any kind of pairwise connectivity, which include deep, convolutional, and fully-connected Boltzmann machines. We leave the analysis of these types of networks for future work.

Acknowledgments

Work supported by DARPA under grant No. HR00111990069. H.M. acknowledges support from a DoD-SMART fellowship. M.D. and Y.R.P. acknowledge partial support from the Center for Memory and Recording Research at the University of California, San Diego. All memcomputing simulations reported in this paper have been done on a single core of an AMD EPYC server.

C. Sminchisescu and M. Welling, in Artificial Intelligence and Statistics (2007) pp. 516–523.
S. Lan, J. Streets, and B. Shabababa, in AAAI (2014) pp. 1953–1959.
S. Arora and B. Barak, Computational Complexity: A Modern Approach (Cambridge University Press, 2009).
F. L. Traversa, P. Cicotti, F. Sheldon, and M. Di Ventra, Complexity 2018 (2018).
F. Sheldon, F. L. Traversa, and M. Di Ventra, arXiv preprint arXiv:1810.03712 (2018).
J. Melchior, A. Fischer, and L. Wiskott, The Journal of Machine Learning Research 17, 3387 (2016).
A. Fischer and C. Igel, in iberoamerican congress on pattern recognition (Springer, 2012) pp. 14–36.
P. M. Long and R. A. Servedio, (2010).
M. A. Carreira-Perpinan and G. E. Hinton, in Aistats, Vol. 10 (Citeseer, 2005) pp. 33–40.
Y. R. Pei, H. Manukian, and M. Di Ventra, arXiv preprint arXiv:1905.05334 (2019).
S. R. B. Bearden, F. Sheldon, and M. Di Ventra, EPL (Europhysics Letters) 127, 30005 (2019).
D. J. MacKay, Information theory, inference and learning algorithms (Cambridge university press, 2003).
G. Hinton, Momentum 9, 926 (2010).
F. L. Traversa and M. Di Ventra, Chaos: An Interdisciplinary Journal of Nonlinear Science 27, 023107 (2017).

Appendix A: Sampling with Memcomputing

The mode-training method introduced in the main text requires sampling the mode of the model distribution of a given RBM. This task can be transformed to sampling the optimum of an equivalent weighted, mixed maximum satisfiability (MAX-2-SAT) optimization problem. To obtain high-quality samples for large models, we employ the memcomputing approach, a novel computing paradigm that employs memory to both store and process information.

1. Memory Dynamics

Our implementation is based on the approach used in Ref. 25 for the satisfiability (SAT) problem, appropriately modified for the MAX-2-SAT optimization problem. For a MAX-2-SAT with \( N \) variables, \( M_1 \) 1-SAT clauses, and \( M_2 \) 2-SAT clauses we have \( i \in [1, N] \) and \( m \in [1, M_2] \). In this case, the equations used to simulate a digital memcomputing machine read

\[
\dot{v}_i = b_i + \sum_{m} \left( W_{2,m} x_m^j x_m^s G_m^i + \rho (1 - x_m^j) R_m^i \right)
\]

(A1)

\[
\dot{x}_m^j = \beta (x_m^j + \epsilon) (C_m - \frac{1}{4})
\]

(A2)

\[
\dot{x}_m^s = \alpha (1 + W_{2,m}) C_m.
\]

(A3)

The voltages, \( v_i \in [-1, 1] \), are continuous representations of the \( N \) Boolean variables of the problem, \( y_i \), with a false assignment represented as \( v_i < 0 \), a true assignment represented as \( v_i > 0 \), and \( v_i = 0 \) is ambiguous. Rather than thresholding the voltages to check the clause states, we use the clause function directly. A 2-SAT clause in Boolean form is comprised of two literals, \( \{l_{i,m}, l_{j,m}\} \), where a literal in the \( m \)-th clause, \( l_{i,m} \), is either a negated, \( \bar{y}_i \), or unnegated, \( y_i \), variable. The Boolean clause is represented as a continuous clause function,

\[
C_m(v_i, v_j) = \frac{1}{2} \min[(1 - q_m,i v_i), (1 - q_m,j v_j)].
\]

(A4)

The factor \( q_{m,i} \) contains the information about the relation between the literal in the \( m \)-th clause, \( l_{i,m} \), and its associated variable, \( y_i \); it evaluates to +1 if \( l_{i,m} = v_i \), and -1 if \( l_{i,m} = \bar{v}_i \). The function is bounded, \( C_m \in [0, 1] \), and we consider a clause to be satisfied when \( C_m(v_i, v_j) < 0.5 \). By thresholding the clause function we also avoid the ambiguity associated with \( v_i = 0 \).

Each clause has a “fast”, \( x_m^j \), and a “slow”, \( x_m^s \), memory variable that serve as indicators of the history of the state of \( C_m(v_i, v_j) \). The memory is “fast” in the sense that it contains information of the recent history of \( C_m \), and “slow” in the sense that it contains information on the entire history of \( C_m \). Both memory variables are bounded, \( x_m^j \in [0, 1] \) and \( x_m^s \in [1, 10 * M_2] \). The offset \( \epsilon = 10^{-3} \) in Eq. (A2) is used to remove spurious steady-state solutions.

The gradient-like term in Eq. (A1) is \( G_m^i = 0 \) if variable \( y_i \) is not associated with any literal in clause \( m \). Otherwise,

\[
G_m^i = q_{m,i} \frac{1}{2} (1 - q_m,j v_j),
\]

(A5)

where \( v_i \) is the value of the voltage corresponding to the other literal in the clause. The “rigidity” term in Eq. (A1) is

\[
R_m^i = \begin{cases} q_{m,i} \frac{1}{2} (1 - q_m,i v_i), & C_m(v_i, v_j) = \frac{1}{2} (1 - q_m,i v_i) \\ 0, & C_m(v_i, v_j) = \frac{1}{2} (1 - q_m,j v_j). \end{cases}
\]

(A6)

This term only influences the voltage that is closest to the satisfying assignment in the clause.

The weight of each 2-SAT clause, \( W_{2,m} \), is incorporated in the dynamics of the slow memory variable and the dynamics of voltages. The weights of the 1-SAT clauses are used to bias the voltage dynamics in Eq. (A1) as \( b_i = (W_{1,i} - W_{1,j}) / 2 \), where \( W_{1,i} \) is the weight of the 1-SAT with a literal that is equivalent to variable \( y_i \), and \( W_{1,j} \) is the weight of the 1-SAT with a literal that is the negation of variable \( y_j \). The weight is zero if no corresponding 1-SAT exists.

The parameter values used for the simulations reported in the main text are \( \alpha = 10, \beta = 0.1, \rho = 0.1 \). At \( t = 0 \), voltages are randomly initialized with \( x_m^j = 0 \) and \( x_m^s = 1 + W_{2,m} \). The equations are then numerically integrated with the forward Euler method using an adaptive time step, \( \Delta t \in [2^{-5}, 2^{-4}] \), until a total integration time of \( t = 500 \) is reached. Then, we take the configuration with the lowest number of unsatisfied clauses as the sample.
2. Computational Cost of Sampling with Dynamics

For simplicity, let us assume the form of an $N \times N$ RBM, resulting in $2N$ voltage variables and $O(N^2)$ clauses as a MAX-2-SAT instance. The time complexity of a forward Euler integration step is dominated by the sparse matrix-vector multiplication of a $2N \times N^2$ sparse matrix. Since each node connects to $N$ other nodes this matrix contains $N^2$ non-zero elements encoding the connectivity structure of the problem. This matrix multiplies the gradient vector of length $N^2$, for a total of $O(N^2)$ floating point operations per second per time step. If the maximum number of timesteps is independent of system size, the total time complexity is then $O(N^2)$. Memory complexity also scales as $O(N^2)$, since the algorithm requires the storage of $4N^2$ floating point elements, and a $2N \times N^2$ sparse matrix with $N^2$ non-zero elements.

3. Performance Comparison Between CD and Memcomputing

Here, we want to show that the RBM energy sampled by the memcomputing approach is consistently better than the one found by CD independently of the size of the RBM. Even though memcomputers are ideally realized as physical devices in hardware\cite{12,28}, here we compare their performance as numerically integrated dynamical systems versus traditional algorithmic methods (e.g., CD).

We then first compute an “exchange rate” between one iteration of the numerical integration and $k$ steps of CD, such that the resulting computational complexity (i.e., wall time on the same processor) will be essentially identical. We discover empirically, across a large range of system sizes, that this exchange rate is about 30 steps of CD per iteration of the dynamics described by Eqs. A1, A2, and A3.

We choose as our test problems a set of randomly initialized $N \times N$ RBMs, with all weights sampled from a normal distribution with $\mu = 0$ and $\sigma^2 = 0.01$. The system sizes ranged from $N = 100$ to 1000, which we chose to be large enough to observe the scaling in time. We then compute the relative energy differences, in percentage, $\Delta \epsilon\% = 100 \times \frac{(E_{\text{Mem}} - E_{\text{CD}})}{E_{\text{CD}}}$, between the energy $E_{\text{Mem}}$ obtained with the memcomputing ODEs described above, as compared to the energy $E_{\text{CD}}$ obtained using CD-$k$. For a direct comparison, we have run the memcomputing solver for $N_{\text{tot}} = 2N$ integration steps, scaled with the system size. Contrastive divergence was then run using the empirical exchange rate, $k = 30 \times N_{\text{tot}}$, resulting in the same computational cost seen in Fig. 4 (right panel).

The energy results are plotted in Fig. 4 (left panel), where the memcomputing dynamics perform very favorably in terms of energies obtained compared CD-$k$, consistently above 400%, often showing an improvement of more than 1000%. In terms of time complexity (right plot), both algorithms follow the same linear trend on a log-log plot, indicating a polynomial scaling. Indeed, the best fit asymptotic behavior of both algorithms is almost cubic. This is consistent with our complexity analysis in the last section. Since both algorithms have a leading order scaling of $O(N^2)$ for a fixed number of iterations, they would scale cubically if we allowed the number of iterations to grow as $N$, the system size. Finally, we want to stress that the set of equations used in the present work are only an example of how to implement a memcomputing solver and have not been optimized in terms of both speed and performance.
Appendix B: Stability of Mode-Assisted Training

The stability of a pre-training procedure to training neural networks is a very desirable feature. This is because the KL divergence cannot be monitored during the pre-training process for a realistically sized RBM, so it is crucial for us to ensure that the KL divergence does not diverge. In this section, we show that using the mode in the model update term will guarantee convergence to a uniform distribution, and there is an optimal learning rate that provides the largest rate of convergence, with the learning rate being easily computable.

Note that in this work, the data term of a mode-assisted update is the same as traditional CD algorithms, so the difference is entirely in the way that the model term is approximated. Therefore, we only have to focus on the model term (which is “approximated” by the mode of the joint distribution), and point out some of its key properties, in particular those pertaining to the stability of the pre-training procedure.

1. Gauging the RBM

For the sake of simplicity, we consider an $n \times m$ unbiased RBM with nodal values of $v \in \{-1, 1\}^n$ and $h \in \{-1, 1\}^m$, then the RBM energy is given by:

$$E(v, h) = -\sum_{ij} W_{ij} v_i h_j.$$  

Note that an RBM with nodal values $\{0, 1\}$ can be trivially transformed into one with nodal values $\{-1, 1\}$. For the analysis in this appendix, we will always assume (unless specifically mentioned) that an RBM is unbiased and equipped with nodal values $\{-1, 1\}$.

Since in our work, we are interested in particular to the mode of the joint distribution, which is equivalently the nodal configuration that minimizes the RBM energy, we give a special denotation to this configuration, $\{v^*, h^*\}$, and name it the ground state of the RBM energy.

Definition B.1 (Ground State Energy). Given an $n \times m$ RBM with weights $W$, we denote the ground state of this RBM to be

$$\{v^*, h^*\} = \arg \min_{\{v, h\}} [E(v, h)].$$

Furthermore, we denote the ground state energy to be

$$E_0(W) = -\sum_{ij} W_{ij} v_i^* h_j^*.$$  

Note that in practice, the ground state of an RBM can be thought of as being unique. In fact, for randomly initialized weights, the probability of having two or more minimal energy states, or degenerate ground states, is of measure zero. In theory, if there were to be multiple ground states, we can randomly select one of them to be $\{v^*, h^*\}$, and our analysis will not be affected at all.

Note that for any RBM, we can always map it to an equivalent RBM such that the ground state is $+1$. This is called a gauge operation, which we formally define as follows.

Definition B.2 (Gauged RBM). Given an $n \times m$ RBM with weights $W$ and ground state $\{v^*, h^*\}$, we define the gauge mapping $G : \mathbb{R}^{nm} \mapsto \mathbb{R}^{nm}$ such that $W' = G(W)$ satisfies the following condition:

$$W'_{ij} = W_{ij} v_i^* h_j^*.$$  

Then we call $W'$ a gauged RBM.

Remark. Note that by this definition, it is easy to see that the ground state of any gauged RBM must be $+1$. This means that the ground state energy of a gauged RBM is simply the sum of its weights

$$E = -\sum_{ij} W'_{ij}.$$  

Furthermore, note that the form of the weight update equation is invariant under conjugation. In other words, if we let $f : \mathbb{R}^{nm} \mapsto \mathbb{R}^{nm}$ denote one iteration of weight update, then it is clear that

$$f = G^{-1} \circ f \circ G.$$
This means that the dynamics of $W$ can be analyzed in terms of the dynamics of $W'$. In this section, we will always assume that the RBM is gauged.

For a gauged RBM, the change of the weight elements (under unit learning rate) as a result of an iteration of mode-informed update is:

$$\delta W_{ij} = -(v_i h_j)_{\text{mode}} = -v_i^* h_j^* = -1.$$ (B1)

Therefore, we see that every weight element is decremented by 1 uniformly across the entire weight matrix, and the energy change of the ground state energy is:

$$\delta E_0 = - \sum_{ij} \delta W_{ij} = nm.$$ (B2)

2. Metric

In order to investigate how the joint probability mass function (joint PMF), or $p(v, h)$, evolves under mode training, we have to look at how the energy changes for all nodal states. To do so, it is useful to define a distance measure between two states, to have a sense of how “far apart” the two states are. We then propose the following distance measure.

Definition B.3 (Metric). We define a spin state to be an ordered $(n + m)$-tuple given by $s = \{v, h\}$. Given two spin states, $s_1 = \{v_1, h_1\}$ and $s_2 = \{v_2, h_2\}$, we let $n_v = \frac{|v_2 - v_1|^2}{2}$ and $m_h = \frac{|h_2 - h_1|^2}{2}$, we define the distance to be

$$d(s_1, s_2) = \frac{n_v}{n} + \frac{m_h}{m} - 2 \frac{n_v m_h}{nm}.$$ (B3)

Remark. Note that $n_v$ simply counts the number of visible nodes that are different between the two states, and $m_h$ counts the number of different hidden nodes that are different. Note that the space of $s$ with this distance definition is a pseudometric space, in the sense that it is possible for the distance between two distinct points to be zero, in particular states that are related by $\mathbb{Z}_2$ symmetry (or global spin flips). This can be easily verified by letting $s_2 = -s_1$, giving us $n_v = n$ and $m_h = m$, and $d(s_1, s_2) = 0$. In this pseudometric space, the distance $d$ is a measure of how “different” two spin states are up to a $\mathbb{Z}_2$ symmetry. A formal discussion of this metric, including a proof of triangle inequality, is provided in Appendix C of our related work. The usefulness of defining the metric this way will be apparent in proposition B.1.

Remark. It is important to note that $\{n_v, m_h\}$ is not uniquely determined by $d$. To see this clearly, we rewrite Eq. (B3) in terms of the following Diophantine equation

$$(2n_v - n)(m - 2m_h) = (2d - 1)nm,$$

solving for integers $n_v \leq n$ and $m_h \leq m$. It is easy to see that this equation is over-determined by realizing that it is possible for the RHS to have multiple prime factors.

3. Energy Change

Equation (B2) gives the change in the energy of the ground state under a mode-assisted update iteration. However, to analyze the stability of the training procedure, it is necessary to look at the energy change of all states. To simplify our discussion, instead of looking at the energy of each individual state, let us consider the average energy of all the states distance $d$ from the modal configuration, which we denote as $\overline{E}(d)$. Note that the average is not the expected value over the joint PMF $p(v, h)$. Rather, it is an unweighted average (or the expected value over a uniform probability measure). It is interesting to note that this average energy only depends on the ground state energy and the distance $d$ from the ground state.

Proposition B.1 (Average Energy). The average energy of states distance $d$ from the ground state is:

$$\overline{E}(d) = (1 - 2d) E_0.$$
spins differing from the mode by \(\{n'_v, m'_h\}\).

The average energy of states with spins differing from the ground state by \(\{n'_v, m'_h\}\) can be expressed as

\[
E_{(n'_v, m'_h)}(E) = E_{(n'_v, m'_h)}\left(\sum_{ij} W_{ij} v_i h_j\right) = \left(\sum_{ij} W_{ij}\right) E_{(n'_v, m'_h)}(v_1 h_1),
\]

where in the last equality, we used the linearity of the expected value and the symmetry of the RBM. We easily see that the marginal probability distribution of a single spin is given by (with the underlying joint distribution being uniform)

\[
\Pr(v_1 = +1) = \frac{n - n'_v}{n}, \quad \Pr(v_1 = -1) = \frac{n'_v}{n},
\]

\[
\Pr(h_1 = +1) = \frac{m - m'_h}{m}, \quad \Pr(h_1 = -1) = \frac{m'_h}{m},
\]

which gives us

\[
E_{(n'_v, m'_h)}(E) = \left[(1 - 2\frac{n'_v}{n})(1 - 2\frac{m'_h}{m})\right]\left[-\sum_{ij} W_{ij}\right] = E_0(1 - 2d').
\]

Therefore, the average energy of states distance \(d'\) from the mode is also \(E(d') = E_0(1 - 2d')\). \(\square\)

Since the average energy distance \(d\) from the ground state is only dependent on \(d\), we expect this to be true also for the change in energy for a state at a distance \(d\) from the ground state, under the weight update routine given in Eq. \([B1]\).

**Proposition B.2 (Energy Change).** Given any state distance \(d\) from the ground state, the change in the energy of that state is given by

\[
\delta E(d) = nm(1 - 2d).
\]

**Proof.** Again, we only have to focus on one particular assignment of the tuple \(\{n_v, m_h\}\) which corresponds to the distance \(d\), and show that the change in energy of a state corresponding to that tuple depends only on \(d\). Without loss of generality (WLOG), we assume that the first \(n_v\) visible nodes are of value \(-1\), and the first \(m_h\) hidden nodes are of value \(-1\). Then the change in energy is given by:

\[
\delta E(d) = -\sum_{ij} \delta W_{ij} v_i h_j
\]

\[
= \sum_{ij} \delta W_{ij} v_i h_j
\]

\[
= \sum_{i=1}^{n_v} \sum_{j=1}^{m_h} v_i h_j\]

\[
= \sum_{i=n_v+1}^{n} \sum_{j=m_h+1}^{m} v_i h_j + \sum_{i=n_v+1}^{n} \sum_{j=m_h}^{m} v_i h_j + \sum_{i=n_v}^{n_v} \sum_{j=m_h}^{m} v_i h_j + \sum_{i=n_v+1}^{n} \sum_{j=m_h+1}^{m} v_i h_j
\]

\[
= n_v m_h - n_v (m - m_h) - (n - n_v)(m_h) + (n - n_v)(m - m_h)
\]

\[
= 4n_v m_h - 2n_v m - 2nm_h + nm
\]

\[
= nm(1 - 2d),
\]

where we have used the fact that \(\delta W_{ij} = -1\) from Eq. \([B1]\). \(\square\)

**Remark.** Note that the energy change is only dependent on the size of the RBM and the distance \(d\) from the ground state, so all the states at distance \(d\) experience the same energy change. Under a given learning rate \(\gamma\), the actual energy change is then

\[
\delta E(d) = \gamma nm(1 - 2d).
\]

Combining propositions \([B.1]\) and \([B.2]\) we see that the energy change can be alternatively written as

\[
\delta E(d) = \gamma nm \frac{E(d)}{E_0}. \tag{B4}
\]

At this point, it is necessary to take an intermission to look at the role that the mode update term plays in the pre-training procedure. From Eq. \([B4]\), we see that the energy change of a state distance \(d\) from the ground state is proportional to the average energy of the states at the same distance \(E(d)\). In the context of the entire pre-training procedure, this energy change can be interpreted as a constant drift term that pulls the
energy back to zero with strength proportional to the average energy of all the states of the same distance. Loosely speaking, the joint distribution will become more uniform under a suitable learning rate. This statement can be made more precise as follows.

Note that this behavior can also be achieved with standard regularization procedures such as an exponential weight decay term \( \delta W_{ij} = -W_{ij} \). However, such regularization techniques are usually undesirable as they do not induce an effective sampling of a multi-modal distribution. Our procedure, however, does not suffer from such drawbacks, and in fact promotes the effective sampling of a multi-modal distribution (see section C).

4. Approaching Uniformity

In this section, we formalize the argument that the RBM energies over all states become more uniform under a mode-assisted update iteration. To do so, we mainly focus on the energy variance across all states, and show that it must decrease under a suitable learning rate. This statement can be made more precise as follows.

Theorem B.3 (Decrease in Energy Variance). If \( 0 < \gamma < -\frac{2E_0}{nm} \), then the variance of the energies \( \text{Var}_s(E(s)) \) over all spin states decreases. The largest decrease in variance occurs when \( \gamma = -\frac{E_0}{nm} \).

Proof. We reiterate the fact that the underlying PMF for the states is assumed to be uniform, or \( f(s) = \frac{1}{2nm} \) for every nodal configuration \( s \). We can then define a random variable \( D \) with its PMF being:

\[
f_D(d) = \frac{1}{2n+m} \sum_{d(n_v,m_h)=d} \left( \frac{n}{n_v} \right) \left( \frac{m}{m_h} \right),
\]

which can be interpreted as the probability of a randomly chosen state to be a distance \( d \) from the ground state. From this PMF expression, we can easily derive the expected value and the variance of the distance of two randomly chosen states

\[
E(D) = \frac{1}{4}, \quad \text{Var}(D) = \frac{1}{4nm},
\]

where we see that the variance is small relative to the expectation value for a large system. We then use the law of total variance to write the variance of the energies over all states as

\[
\text{Var}(E(s)) = \mathbb{E}_D \left[ \text{Var}_s(E(s) \mid d(s) = D) \right] + \text{Var}_D \left[ \mathbb{E}_s(E(s) \mid d(s) = D) \right].
\]

We first begin by focusing on the first term. Note that the term \( \text{Var}_s(E(s) \mid d(s) = D) \) is the conditional variance of energies of the states distance \( D \) from the mode. If we update the energies according to Eq. (B4), then the new variance can be written as \( \text{Var}_s(E(s) + \gamma nm(1 - 2D) \mid d(s) = D) \). The term \( \gamma nm(1 - 2D) \) is dependent only on \( D \) but not the specific nodal configuration \( s \), so it is just a constant offset in the context of the conditional variance, and the variance will remain constant. Therefore, the first term of the variance decomposition is constant, and we only have to focus on the second term, which can be conveniently written as:

\[
\text{Var}_D(\overline{E}(D)) = \text{Var}_D(E_0(1 - 2D)) = 4E_0^2 \text{Var}(D) = \frac{E_0^2}{nm}.
\]

After a weight update, this variance becomes

\[
\text{Var}_D(\overline{E}(D) + \gamma nm \frac{\overline{E}(d)}{E_0}) = 4E_0^2(1 + \frac{\gamma nm}{E_0})^2 \text{Var}(D) = \frac{E_0^2}{nm}(1 + \frac{\gamma nm}{E_0})^2.
\]

In this form, it is easy to see that the variance decreases when the learning rate satisfies

\[
0 < \gamma < -\frac{2E_0}{nm},
\]

with the largest decrease being \( \delta \text{Var}_D(\overline{E}(D)) = 4E_0^2 \text{Var}_D(D) = \frac{E_0^2}{nm} \), which occurs at the learning rate \( \gamma = -E_0/nm \). This is then our optimal learning rate. \( \square \)
Remark. To avoid confusion, note that $E_0$ is negative, so $- \frac{2E_0}{\text{min}}$ is positive, so the learning rate $\gamma$ is bounded in some positive interval. Note that the two biases for the visible and hidden spins can be expressed as two ghost spins, thereby effectively adding one more spin to each layer. By taking into account the biases, we see that the largest decrease of the variance occurs when

$$\gamma \approx \frac{-E_0}{(n + 1)(m + 1)},$$

which is what we use in the main text.

There are two important things to note here. First, the learning rate, as presented in Eq. (B9) is generally very large and is only optimal in the sense that it provides the fastest convergence to a uniform joint PMF, which is desirable for a stable pre-training routine, but not necessarily optimal for minimizing the KL divergence. The practical usefulness of Eq. (B9) is to mainly provide an upper bound to the learning rate that ensures stability. It should be noted that the analysis ignores the presence of the data term (see Eq. (6) in the main text) and is only carried out over a single iteration; in other words, it may be possible that a large learning rate will force the system into a local minimum in the KL divergence rather quickly. Therefore, in the practical setting a smaller learning rate would be more beneficial. In the main paper, we then normalized this learning rate with the learning rate of CD, which results in $\epsilon_{CD} \gamma < \gamma$ (as $\epsilon_{CD} < 1$).

The second thing to note is that Eq. (B9) is not exact as the ghost spins are fixed nodes that cannot be "flipped", so theorem B.1 no longer applies, meaning that the average energy of states distance $d$ from the ground state can no longer be uniquely determined by $E_0$ and $d$ alone. Nonetheless, for large RBMs, the contribution from biases is relatively small, and the approximation is close to exact.

5. Suboptimal Updates

Before we conclude this section, we make two final remarks concerning suboptimal updates, or updates that are not informed by the global mode directly. The first remark pertains to a practical setting where locating the global mode is difficult or too computationally expensive, and only an approximate mode can be obtained, or a state with energy close to the ground state. We discuss how an update informed by this state still ensures stability. The second remark compares a mode-assisted update with an update with the model term sampled by some form of stochastic algorithm (such as CD), and we show that the latter update procedure does not ensure stability.

Note that in Eq. (B1), we transformed the weight elements such that the ground state is $v^* = +1$ and $h^* = +1$. However, this procedure is general and can be done for any given state. Given any two states, $v_1$ and $h_1$ with some associated energy $E_1$, it is always possible to gauge the RBM in a way such that $v_1 = +1$ and $h_1 = +1$. The previous proofs will still carry through for $E_1$ as long as $E_1 < 0$. This means that the mode training procedure does not hinge on the fact that the weight update has to be informed by the exact ground state, and any state sufficiently close to the ground state should suffice. However, it should be noted that using the ground state to inform the weight update provides the greatest decrease in energy variance since the maximum of $\delta \text{Var}_D(\mathcal{E}(D))$ scales quadratically with $E_0$ (see Eq. (B7)).

Note that in theorem B.3, the argument that the conditional variance of the energies conditioned on some distance $d$ from the ground state does not change is based on the fact that the weights are updated uniformly across the RBM according to Eq. (B1). However, for a stochastic algorithm, the weight updates are clearly not uniform (or even deterministic for that matter), so nothing can be said about the change of the conditional variance. It is possible for the conditional variance to increase under a stochastic update, thus pulling the energies away from uniformity if the magnitude of the increase overcomes the decrease in the second term in Eq. (B6) (the ground state variance).

To conclude this subsection, we discuss briefly the contribution of the data term in updating the weight matrix. Clearly, if we look at the gauged RBM matrix, the change in each element generated by the data term is bounded above by $+1$, meaning that its contribution cannot overcome the guaranteed $-1$ decrease generated by the mode update term. This means that it is impossible for the ground state energy to decrease even in the presence of the data term, so the mode of the joint distribution must not increase, thus the training never diverges. This effectively ensures the global stability of our mode-assisted training method.

Appendix C: Efficient Sampling of Multi-modal Distributions

So far, we have shown that our update procedure guarantees stability. However, as briefly mentioned at the end of section B.3, stability is also guaranteed by standard regularization terms such as the weight decay term, $\delta W_{ij} = -W_{ij}$. In this section, we make the crucial distinction between our procedure and standard weight
regularization procedures by pointing out the key phenomenon that our procedure is capable of efficiently exploring the landscape of a multi-modal PMF.

This property of the mode-training method is most readily analyzed from the perspective of the frustration index of the RBM instance. The frustration index can be interpreted as a measure of the difficulty of discovering the nodal ground state of a given RBM instance, and interestingly, an increase in the frustration index is correlated with an increased rate of exploration of the multi-modal distribution. Therefore, in some sense, for a given iteration of weight updates, the difficulty of finding the mode of that distribution is “compensated” by an increased efficiency of PMF exploration.

We begin by formally defining the frustration index, followed by a brief explanation of how the mode-training algorithm explores efficiently the PMF. Finally, we relate the two concepts in a cohesive manner. We provide an extensive analysis on the frustration of the RBM and its practical applications in our related work.24

1. Frustration Index

The frustration index is the ratio between the sum of unsatisfied couplings at the ground state and the sum of all coupling strengths. Formally, for a gauged RBM, it can be defined as follows

\[ f = \frac{1}{2} \left[ \frac{\sum_{ij} |W_{ij}| - \sum_{ij} W_{ij}}{\sum_{ij} |W_{ij}|} \right]. \]

This index is closely related with the degeneracy of the low-energy states. In other words, with an increase in the frustration index, the excited states will be spaced closer to the ground state in energy. Furthermore, for a highly frustrated system, the transition from the ground state to the excited states usually involves flipping a large cluster of nodes. This gives rise to a large population of local minima in the energy landscape spaced far apart in distance but close together in energy (in terms of the metric discussed in section B.2), and this property of a highly frustrated system makes it difficult for local search algorithms to locate the global minimum. This motivates the need for an algorithm that is able to learn the long-range correlations of the RBM spins, and a possible candidate of this algorithm is presented in section A.

2. Inefficiency of Weight Decay

In this section, we discuss briefly why the standard weight decay algorithm \( \delta W_{ij} = -\gamma W_{ij} \) (where \( \gamma \) is some learning rate) is not efficient in assisting local algorithms in sampling a multi-modal distribution. To begin with, we first recall that the joint distribution of the RBM is

\[ p(v, h) = \exp(-E(v, h)), \]

where \( E(v, h) = \sum_{ij} W_{ij} \) for a gauged RBM. Note that the weight decay update is a contracting affine transformation of the energies of all states, or simply a rescaling of the energies by some constant \( \beta = (1 - \gamma) < 1 \), meaning that the joint distribution transforms as

\[ p(v, h) \rightarrow p(v, h)^\beta, \]

where the normalization condition is ignored.

Of course, the distribution does become more uniform under this transformation; however, the ordering of the states with respect to their energies will not change, meaning that the ordering of the dominant modes remains invariant under this transformation. In other words, a poorly initialized Markov chain trapped under a dominant mode will still remain trapped unless \( \beta \) becomes sufficiently small; this means that a large learning rate, \( \gamma \), is required to free the Markov chain and allow efficient exploration of the joint distribution. However, a large learning rate in this context is undesirable, as it brings the RBM to uniformity in a drastic manner, which voids much of the information gained from the previous iterations of pre-training. The inefficiency of this approach boils down to the indiscriminate update of the weight matrix that is ignorant of the energy ordering of the states or the distance between them (see definition B.3).

Our mode-assisted update, on the other hand, updates the weight matrix based on the ground state configuration of the RBM, resulting in a maximal increase in energy for the ground state, and the energy change is “propagated” to the other states based on their distances from the ground state (see proposition B.2). An entirely different energy landscape will then emerge under this update procedure even under a small learning rate, and it is likely that a new ground state at a faraway distance will “pop” up. The next update iteration is
then based on this new found mode, and the process is repeated. Effectively, we are dynamically sampling the energy landscape by making large leaps between dominant states without resorting to forcing uniformity on the energies.

3. Global Mode Cycling

For the sake of simplicity, consider a gauged RBM with a joint distribution having three dominant states, with their RBM energies being $E_0 < E_1 < E_2$. The heuristic analysis in this section can be easily generalized to multi-modal distributions with arbitrary number of dominant states. We can also assume that the pairwise distances between the three modes are the same (meaning that the three modes form an equilateral triangle under the metric defined in definition B.3), which we can then denote simply as $d$.

If we assume that the learning rate is $\gamma$, then from Eq. (B4), we see that the new energies of the three states will become

$$
E_0^{(1)} = E_0^{(0)} + nm\gamma \\
E_1^{(1)} = E_1^{(0)} + nm\gamma(1 - 2d) \\
E_2^{(1)} = E_2^{(0)} + nm\gamma(1 - 2d),
$$

where we are assuming that the magnitude of the learning rate is much larger than the energy gaps, or more precisely

$$
\gamma > \frac{E_1^{(0)} - E_0^{(0)}}{2nm}, \quad \text{(C1)}
$$

where we note that the lower bound of gamma is proportional to the energy difference between the first excited state and the ground state. This guarantees that after one update, the ordering of the new energies of the states will become

$$E_1^{(1)} < E_2^{(1)} < E_0^{(1)},$$

which means that $E_1^{(1)}$ is the new ground state energy, and the next iteration of weight update will be based on state $E_1^{(1)}$, resulting in the following new energies

$$
E_1^{(2)} = E_1^{(1)} + nm\gamma \\
E_2^{(2)} = E_2^{(1)} + nm\gamma(1 - 2d) \\
E_0^{(2)} = E_0^{(1)} + nm\gamma(1 - 2d),
$$

The energies are then reordered as

$$E_2^{(2)} < E_0^{(2)} < E_1^{(2)},$$

so $E_2^{(2)}$ becomes the new ground state energy. And similarly, the third iteration will recover the original energy ordering $E_0^{(3)} < E_1^{(3)} < E_2^{(3)}$.

Therefore, we see that in general, whenever we perform a weight update, the energy ordering of the modal states will experience a left circular shift, so we are, in some sense, sampling the multiple modes in a cyclic fashion, which allows us to effectively cover a large volume of the probability measure.

4. Relationship between Frustration and Mode Sampling

Now, we discuss how an increase in the frustration index is conducive to an efficient sampling of the multi-modal distribution. We here consider simply a gauged $n \times n$ RBM, with ground state energy $E_0$. We denote the average energy of states distance $d$ from the ground state as $\overline{E}(d)$ (see proposition B.1). Under an iteration of mode update, the new energies are (see Eq. (B4))

$$
E'_0 = E_0 + n^2\gamma \\
\overline{E}(d)' = \overline{E}(d) + n^2\gamma(1 - 2d),
$$
For the sake of simplicity, consider the case where the frustration index of the RBM is zero, then all the weights can be assumed positive. Furthermore, we make the simplifying assumption that the weights are iid random variables with uniform distribution in $[0, 1]$. We then make the following claim.

**Proposition C.1.** If we update the weight matrix continuously with the mode-assisted update procedure, then the new ground state will differ from the original ground state by distance $d \sim \frac{1}{n}$ almost surely.

More formally put, if we let $E_{\text{min}}(d)$ be the minimum energy of states distance $d$ from the ground state, and $\Delta E(d) = E_{\text{min}}(d) - E_0$. Then the smallest learning rate for which a new ground state can emerge is

$$
\gamma' = \inf \{\gamma \mid 3d \in (0, 1), \ 2dn^2\gamma = \Delta E(d)\}.
$$

If we let $d'$ be the distance such that $2d'n^2\gamma = \Delta E(d')$, then

$$
\lim_{n \to \infty} \Pr(d' > \frac{1}{n}) = 0.
$$

**Proof.** Given any state distance $d$ from the ground state, we have

$$
\Pr(2dn^2\gamma > E(d) - E_0) = \frac{1}{2} \text{erfc}(\sqrt{6}(1 - 2\gamma)nd).
$$

WLOG, we can also assume that $n$ is a prime number, then the number of states distance $\frac{k}{n}$ from the ground state (where $k < n$) is $2\binom{n}{k}$, which gives us (denoting $\beta = \sqrt{6}(1 - 2\gamma)$ and $k = nd$).

$$
\Pr(2dn^2\gamma > \Delta E_{\text{min}}(d)) = 1 - \left[1 - \frac{1}{2} \text{erfc}(\sqrt{6}(1 - 2\gamma)nd)\right]^{2\binom{n}{k}} \\
\sim 1 - \exp \left[- \binom{n}{k} \frac{e^{-\beta k^2}}{\sqrt{\pi} \beta k}\right] \\
= J(n, k, \beta).
$$

Note that $\forall \epsilon \in (0, 1)$, we let $\beta'$ such that $J(n, 1, \beta') = 1 - \epsilon$, then we have

$$
\forall k \in [2, n], \ \lim_{n \to \infty} J(n, k, \beta') = 0,
$$

which proves the proposition.

This result implies that in the limit of large $n$, the new ground state is only likely going to differ from the old ground state by distance $d \sim \frac{1}{n}$, so we are only moving away from the old ground state by a very small distance. This means that a small frustration is not conducive to an efficient sampling of the phase space.

**b. Large Frustration**

A highly frustrated system is generally hard to study, so we here provide a brief heuristic argument for the efficient sampling of the PMF for a highly frustrated RBM. Recall that in the case of large frustration, the first excited state differs from the ground state by a large number of nodes (hence a large distance $d$) but by only a small amount of energy. Also recall from Eq. (C1) that the lower bound of the learning rate scales proportionally to the energy difference and inversely proportionally to the distance. Putting the two results together, we see that in order for the first excited state to become the new ground state, we only require a very small learning rate (which is conducive to a faster convergence of the KL-divergence), and furthermore, transitioning from the ground state to the new ground state effectively allows us to traverse a large distance, which allows us to efficiently sample the full PMF.

**Appendix D: Defining Modal Correspondence**

The goal of this section and the following is to show that the mode of the marginal distribution of the visible layer, $p(v)$, and the mode of the joint distribution, $p(v, h)$, are strongly correlated. We will dedicate this section to a formal definition of this notion of correspondence, and provide a full proof of correspondence in the following section. For now, we can interpret this strong correspondence as the phenomenon that there is a high chance for the mode of $p(v)$ and the mode of $p(v, h)$ to overlap in the configuration of $v$, meaning that the mode of $p(v, h)$ can be used to “approximate” the mode of $p(v)$.
1. Unnormalized PMFs

Recall that we base the analysis in this section on an $n \times m$ unbiased RBM with nodal values of $v \in \{-1, 1\}^n$ and $h \in \{-1, 1\}^m$. The discussion in this section can be easily extended to a biased RBM. To ease the burden of notation, we first begin by defining an angle variable $\theta = v \cdot W$, which allows us to rewrite the RBM energy and the joint probability mass function (PMF) as follows:

$$E = \theta \cdot h,$$
$$p(v, h) = \frac{1}{Z} e^{-E} = \frac{1}{Z} e^{\theta \cdot h},$$

where $Z$ is the partition function of the RBM. The marginal PMF of the visible layer can be obtained by fixing the visible layer and summing the joint PMF over all the hidden layer configurations:

$$p(v) = \sum_h p(v, h) = \frac{1}{Z} \sum_h e^{\theta \cdot h} = \frac{1}{Z} \prod_{j=1}^{m} 2 \cosh(\theta_j),$$  \hfill (D1)

where the last equality is obtained by factoring the sum into each individual hidden nodes.

Since we are mainly concerned with the correspondence of the modal configurations instead of the normalized probability mass, we can simply ignore the constant prefactor $\frac{1}{Z}$ as the normalization prefactor and simply look at the unnormalized PMFs:

$$P(v, h) = e^{\theta \cdot h} \quad P(v) = \prod_{j=1}^{m} 2 \cosh \theta_j,$$

where the use of the capital letter $P$ is to denote the unnormalized PMF. Note that since $p \mapsto P$ is an affine transformation, the ordering of the states in terms of their energies is invariant.

An issue we have to first address is that the nodal configuration of the joint distribution is described by the configurations of both layers $(v, h)$, while the nodal configuration of the marginal distribution is only described by the visible layer $v$. So in order to compare the nodal configurations of the two PMFs, we have to relegate $P(v, h)$ into a PMF that only depends on $v$, which we do as follows:

**Definition D.1.** Given a PMF $P(v, h)$, we denote

$$Q(v) = \max_h P(v, h),$$

**Remark.** In other words, $Q(v)$ is the maximum of the $P(v, h)$ over all $h$ under some fixed $v$. Note that the purpose of this definition is to have the mode of $Q(v)$ be the same as the mode of $P(v, h)$ “projected” onto the space of $v$. In other words, if we let $\{v^*, h^*\}$ be the mode of the joint distribution $P(v, h)$, then we have the following:

$$\arg \max_v Q(v) = \arg \max_v (\arg \max_h P(v, h)) = v^*$$

This means that the mode of the joint distribution $P(v, h)$ is the same as the mode of $Q(v)$ in the $v$ component.

**Remark.** Note that there is a bijection between the visible configurations and the angle variables given by $\theta = v \cdot W$, so we can make $Q$ depend on $\theta$ instead, or $Q(\theta)$, which is usually the form that we will be using for this section. Similarly, we can also write $P(\theta)$ as the unnormalized marginal distribution.

To simplify the analysis of modal correspondence, we first obtain a closed form expression for $Q(v)$:

**Lemma D.1.** $Q(v) = \exp(\sum_j |\theta_j|)$. 

**Proof.** Note that the expression for $P(v, h)$ can be written as $P(v, h) = \exp(\sum_j \theta_j h_j)$. It then follows that $\arg \max_h P(v, h) = \exp(\arg \max_h \sum_j \theta_j h_j) = \exp(\sum_j \arg \max_h \theta_j h_j)$. Since $h_j \in \{-1, 1\}$, it is easy to see that $\arg \max_h \theta_j h_j = |\theta_j|$. Therefore, we have $Q(v) = \arg \max_h P(v, h) = \exp(\sum_j |\theta_j|)$

Now, if we denote $v^\bullet$ as the $v$ component of the mode of $P(v, h)$ and $v^\circ$ as the mode of $P(v)$, then the question of whether the marginal mode equals to the joint mode can be succinctly expressed as

$$v^\circ \equiv v^\bullet.$$  

The equality, in fact, does not hold in the absolute sense, and it is very easy to construct pathological examples to violate the equality. However, for practical purposes, we only need this equality to hold with some non-negligible probability for an RBM with weights randomly sampled from some distribution. We then formally define the notion of correspondence as follows
Definition D.2. Given an \( n \times m \) RBM with weights \( \mathbf{w} \) sampled from some distribution \( f_{\mathbf{w}}(\mathbf{w}) \), we say that the marginal mode and joint mode of the RBM are strongly correlated if the following holds

\[
\Pr \left[ \bigwedge_{\mathbf{v} \in \{-1,+1\}^n} P(\mathbf{v}) \leq P(\mathbf{v}^*) \right] \geq 0.5,
\]

where \( \mathbf{v}^* \) is the \( \mathbf{v} \) component of the mode of \( P(\mathbf{v}, \mathbf{h}) \).

Remark. First, we recall that \( \mathbf{v}^* \) is the \( \mathbf{v} \) component of the joint distribution \( P(\mathbf{v}, \mathbf{h}) \). If \( \mathbf{v}^* \) is also the mode of the marginal distribution \( P(\mathbf{v}) \), or \( \mathbf{v}^* = \mathbf{v}^\circ \), then clearly we require that \( P(\mathbf{v}) \leq P(\mathbf{v}^*) = P(\mathbf{v}^\circ) \), for all \( \mathbf{v} \) configurations. In order to weaken the condition of exact modal correspondence, we simply require that the probability of the inequality, \( P(\mathbf{v}) \leq P(\mathbf{v}^*) \), holds for all \( \mathbf{v} \) to be greater than some arbitrary value, which we chose to be 0.5 here.

2. Trivial Cases

There are two cases where proving the modal correspondence is trivial; the two cases occur at the beginning and end of the pre-training respectively. At the beginning of the training, the frustration index is small for the RBM, and the system is trivially ferromagnetic. At end of the training, the magnitude of the weights are large, and the nodal activation of the hidden layer is almost certain.

a. Small Frustration

If the frustration index is small, we can state the following.

Proposition D.2. \( \arg \max_{\mathbf{v}} Q(\mathbf{v}) = \arg \max_{\mathbf{v}} P(\mathbf{v}) \) for an RBM with zero frustration.

Proof. We look at the gauged RBM where all weight elements are non-negative. Recall that the ground state of a gauged RBM is +1, then we have \( \arg \max_{\mathbf{v}} \{\mathbf{v}, \mathbf{h}\} P(\mathbf{v}, \mathbf{h}) = +1 \), which implies \( \arg \max_{\mathbf{v}} Q(\mathbf{v}) = +1 \). Note that \( P(\mathbf{v}) = \prod_j 2 \cosh(\theta_j) = \prod_j 2 \cosh(\sum_i W_{ij} v_i) \leq \prod_j 2 \cosh(\sum_i W_{ij}) = P(+1) \), where the inequality comes from the fact that \( W_{ij} \geq 0 \) and \( v_i \in \{-1,1\} \), so we have \( \arg \max_{\mathbf{v}} P(\mathbf{v}) = +1 \) as well. The proposition is then shown.

Remark. Note that this proposition implies directly modal correspondence as defined in definition D.2 in the absolute sense.

b. Large Weights

Near the end of the RBM training, the magnitude of the weights are usually very large (thus also the magnitude of the elements of \( \theta \), and the activation of the hidden nodes becomes increasingly certain. Intuitively speaking, this means that given any visible configuration, there is only one dominant hidden configuration corresponding to it. Therefore, the marginal distribution \( p(\mathbf{v}) \) (which involves the sum over all hidden configurations) can be effectively approximated with the joint distribution \( p(\mathbf{v}, \mathbf{h}) \). We formalize this argument as follows:

Proposition D.3. Given an \( n \times m \) weight matrix, \( \mathbf{W} \), with the joint mode \( \mathbf{v} \) satisfying

\[
\forall j \in [1, m], \quad \left| \sum_i W_{ij} v_i \right| \neq 0,
\]

and the ground state is not degenerate. Then \( \exists M > 0 \), such that for an RBM with the weight matrix, \( M\mathbf{W} \), the following is true

\[
\arg \max_{\mathbf{v}} Q(\mathbf{v}) = \arg \max_{\mathbf{v}} P(\mathbf{v}).
\]

Proof. We look at the gauged RBM so that the ground state is +1, then we set

\[
\theta_j = \sum_{ij} W_{ij} v_i > 0.
\]

Let \( \mathbf{v}' \) be the visible component of any other state, then we denote

\[
\theta_j' = \sum_{ij} W_{ij} v_i'.
\]
Then the following must be true

\[ \exists \epsilon > 0, \quad \sum_j |\theta_j| - \sum_j |\theta_j'| = \epsilon. \]

Recall from proposition [D.1] that

\[ Q(\theta) = \prod_j \exp(|\theta_j|). \]

Furthermore, we can write the marginal distribution as

\[ P(\theta) = \prod_j 2 \cosh(|\theta_j|). \]

Note that \( \lim_{x \to \infty} \frac{2 \cosh(x)}{\exp(x)} = 1 \). This implies that \( \forall \delta > 0, \exists x > 0 \) such that \( 2 \cosh(x) < (1 + \delta) \exp(x) \). If we assume that the proposition is false, then we can set \( \delta' < \exp(\epsilon/m) - 1 \) and choose \( M > 0 \) such that

\[
(1 + \delta')^m \prod_j \exp(M|\theta_j'|) \geq \prod_j 2 \cosh(M|\theta_j'|) \geq \prod_j \exp(M|\theta_j|)
\]

\[
\implies m \log(1 + \delta') + \sum_j M|\theta_j'| \geq \sum_j M|\theta_j| \implies \epsilon > \epsilon,
\]

a contradiction. Therefore, the proposition must be true.

\[
\square
\]

Appendix E: Showing Modal Correspondence

We have shown in the previous section that the modes of the joint and marginal PMF of the RBM correspond absolutely under two trivial cases: large weights and small frustration. The remaining case where the weights are small and the frustration is large is highly non-trivial, and we dedicate this entire section to showing, in the probabilistic sense, the modal correspondence as defined in definition [D.2]. The problem of showing modal correspondence can be reduced to analyzing the value Gaussian integrals over simplexes of varying sizes. Before we tackle this problem, we first formalize the notion of a random RBM.

1. Random RBM

**Definition E.1** (Random RBM). A Random RBM is an RBM with a weight matrix, \( W \), whose elements are iid normal variables with mean \( \mu = 0 \) and standard deviation \( \sigma \). Furthermore, the configuration of the visible layer is sampled uniformly from \( \{-1, +1\}^n \).

**Lemma E.1.** Given a random RBM, \( \{v, W\} \), the angle variables,

\[ \theta = v \cdot W, \]

are iid normal variables with mean 0 and variance \( \sigma_\theta^2 = n\sigma^2 \).

**Proof.** This is a three stage proof. First, we have to show the product \( W_{ij}v_i \) is a random normal variable, so the elements of \( \theta \) are also random normal variables. Second, we show that the probability distribution function (pdf) of \( \theta \) is a multivariate normal distribution. Finally, we show that the elements of \( \theta \) are uncorrelated, thus implying that they are independent.

To show that \( W_{ij}v_i \) is a random normal variable, we find the cumulative distribution function (CDF) of this product, and show that it is the CDF of a normal distribution. The CDF of the product is given by

\[
P(W_{ij}v_i \leq z) = P(W_{ij} \leq z)P(v_i = 1) + P(W_{ij} \geq -z)P(v_i = -1) = \frac{1}{2}(P(W_{ij} \leq z) + P(W_{ij} \geq -z)) = \frac{1}{2}(2P(W_{ij} \leq z)) = P(W_{ij} \leq z),
\]
which is simply the CDF of $W_{ij}$. Note that we have exploited the fact that the PDF of $W_{ij}$ is even. Therefore, $	heta_j = \sum_i W_{ij}v_i$ is the sum of $n$ random normal variables, resulting in another random normal variable $N(0, n\sigma^2)$.

To show that the pdf of $\theta$ is a multivariate normal distribution, it is sufficient to show that any linear combination of the angle variables is a normal variable. Let the linear combination be

$$
\sum_j c_j\theta_j = \sum_j c_j(\sum_i W_{ij}v_i) = \sum_i v_i(\sum_j W_{ij}c_j).
$$

If we denote $\phi_i = \sum_j W_{ij}c_j$, then the linear combination can be expressed as $\sum_i v_i\phi_i$. Note that we can show that $v_i\phi_i$ is a random normal variable by the same argument as above, then $\sum_j v_i\phi_i$ must be a random normal variable as well, as it is the sum of independent normal variables. Therefore, $\theta$ is a multivariate normal distribution.

Finally, since the PDF of $\theta$ is a multivariate normal distribution, to show that $\theta$ are independent random normal variables, it is sufficient to show that any two elements of $\theta$ are uncorrelated. For $j_1 \neq j_2$, we have

$$
\text{Cov}(\theta_{j_1}, \theta_{j_2}) = \text{Cov}(\sum_i W_{ij_1}v_i, \sum_i W_{ij_2}v_i) = \mathbb{E}(\sum_{i_1,i_2} W_{ij_1}W_{ij_2}v_{i_1}v_{i_2})
$$

$$
= \sum_{i_1,i_2} \mathbb{E}(W_{ij_1}W_{ij_2}) \mathbb{E}(v_{i_1}v_{i_2}) = \sum_i \mathbb{E}(W_{ij_1}) \mathbb{E}(W_{ij_2}) = 0,
$$

where we have used the fact that $\mathbb{E}(v_{i_1}v_{i_2}) = \delta_{i_1i_2}$. The lemma is then proved.

\begin{remark}
An important consequence of this lemma is that we can parameterize a random RBM with the angle variables $\theta$, as the distributions of $v$ and $W$ are fully captured as the distribution of $\theta$ as iid normal variables.
\end{remark}

As an RBM with large weights trivially satisfies the modal correspondence condition (see proposition D.3), we can assume the weights are small for the sake of non-triviality, and make the following approximation for $\theta$:

$$
P(\theta) = \prod_j 2 \cosh(\theta_j) \approx \prod_j (2 + \theta_j^2) \approx 2^m + 2^{m-1}(\sum_j \theta_j^2) \rightarrow \sum_j \theta_j^2,
$$

where the right arrow in the last line denotes an affine transformation which preserves the ordering of the probability masses. Similarly, we approximate $Q(v)$ as follows:

$$
Q(\theta) = \exp(\sum_j \theta_j) \approx 1 + \sum_j \theta_j \rightarrow \sum_j \theta_j.
$$

\section{Simplex Condition}

To show modal correspondence, it is convenient for us to fix $Q(\theta)$, and analyze the conditional distribution of $\theta$. In particularly, we wish to show that if $Q(\theta)$ is large, then the conditional expected value of $P(\theta)$ will also be large. First, we denote the conditional distribution of $\theta$ under a fixed $Q(\theta)$ as $f(\theta \mid Q(\theta) = \alpha)$. Recall that $Q(\theta) = \sum_j \theta_j$, so the level set of $Q(\theta)$ are composed of simplexes, one in each quadrant. Note that $\theta$ are iid normal variables, so the PDF is spherically symmetric. Furthermore, $\theta^2$ is also spherically symmetric. This means that all moments of $P(\theta)$ are invariant if we rewrite the condition as

$$
\{Q(\theta) = \alpha\} \land \{\theta \geq 0\}.
$$

\begin{lemma}
The following two conditional distributions are equivalent.

$$
f(\theta^2 \mid \{Q(\theta) = \alpha\} \land \{\theta \geq 0\}) = f(\theta^2 \mid \sum_j \theta_j = \alpha).
$$
\end{lemma}

\begin{proof}
Omitted. Follows directly from the spherical symmetry of the PDF of $\theta$.
\end{proof}

The graph of condition [E1] is a regular simplex of length $\sqrt{2}\alpha$ and dimension $m - 1$ in the first quadrant, which we can denote as $\Delta^m_{1\alpha}$, and we can write the conditional PDF as $f(\theta \mid \Delta^m_{1\alpha})$. It is convenient for us to apply an orthogonal transformation to $\theta$, and we denote the new angles as $\phi = \Theta \theta$. Note that the new angles are still independent normal random variables, since an orthogonal transformation preserves the independence of normal variables. The orthogonal transformation is chosen such that $\phi_1$ points from the origin to the centroid of the simplex. We denote $\phi^\perp$ as the components of $\phi$ other than $\phi_1$, meaning that $\phi = (\phi_1, \phi^\perp)$.
Lemma E.3. Let \( n = \frac{\alpha}{\sqrt{m}} \hat{\phi}_1 \), then
\[
\forall \theta \in \Delta^{m-1}_\alpha, \quad \theta^2 = (n^2 + \varphi^2).
\]

Proof. This can be shown by realizing that \( n \) is the displacement of the centroid from the origin, which is perpendicular to the \( m - 1 \) hyperplane the simplex is in. In other words
\[
\forall \theta \in \Delta^{m-1}_\alpha, \quad (\theta - n) \cdot n = 0.
\]

\[\boxdot\]

Remark. This lemma allows us to express the condition distribution of \( \theta \) in terms of \( \varphi \).

Lemma E.4. Let \( \varphi \) be iid normal variables with variance \( \sigma^2_{\theta} \), then
\[
f(\theta \mid \Delta^{m-1}_\alpha) = \left[ \frac{f(\varphi)}{\int_{\Delta^{m-1}_\alpha} f(\varphi)} \right] f(\varphi \mid \Delta^{m-1}_\alpha).
\]

Proof. This can be shown by realizing that if \( \theta \) are iid normal variables, then \( \varphi \) must also be iid normal variables. The intersection of the PDF of iid normal variables and a hyperplane is also a PDF of iid normal variables (with one less dimension).

The conditional expected value of \( P(\theta) \) can then be expressed as
\[
E(P(\theta) \mid Q(\theta) = \alpha) = E(\sum_j |\theta_j|^2 \mid \Delta^{m-1}_\alpha) = \frac{\alpha^2}{m} + E(\varphi^2 \mid \Delta^{m-1}_\alpha).
\]

Similarly, the conditional variance can be expressed as
\[
\text{Var}(P(\theta) \mid Q(\theta) = \alpha) = \text{Var}(\varphi^2 \mid \Delta^{m-1}_\alpha).
\]

Note that the \( k \)-th moment of \( \varphi^2 \) conditioned on the simplex is
\[
E_{\Delta^{m-1}_\alpha}(\varphi^{2k}) = \left[ \int_{\Delta^{m-1}_\alpha} f(\varphi) \varphi^{2k} \right] / \int_{\Delta^{m-1}_\alpha} f(\varphi).
\]

To lessen the burden of notation, we denote the following Gaussian integral
\[
J(\sigma, \alpha, k) = \int_{\Delta^{m-1}_\alpha} d\varphi \varphi^{2k} \exp \left[ -\frac{\varphi^2}{2\sigma^2} \right]
= \sqrt{m} \int_0^\infty d\theta \delta(\sum_j \theta_j - \alpha) \theta^{2k} \exp \left[ -\frac{\theta^2}{2\sigma^2} \right],
\]

and we note that
\[
J(\sigma, \alpha, k) = \left[ \frac{\partial}{\partial \left( \frac{\theta^2}{2\sigma^2} \right)} \right]^{2k} J(\sigma, \alpha),
\]

where the last argument of \( J \) is assumed 0 in its absence. We can then write
\[
E_{\Delta^{m-1}_\alpha}(\varphi^{2k}) = \frac{1}{J(\sigma_\theta, \alpha, 0)} \left[ \frac{\partial}{\partial \left( \frac{\theta^2}{2\sigma^2} \right)} \right]^{2k} J(\sigma_\theta, \alpha, 0).
\]

Before we proceed to evaluate this integral, we first recall that the size of the simplex is given as \( \alpha = \sum_k |\theta_j| \), which means a “typical” value of \( \alpha \) is dependent on the variance, \( \sigma^2_{\theta} \). In fact, we note that \( |\theta_j| \) is a half normal variation with mean \( \sqrt{\frac{2}{\pi}} \sigma_\theta \), then a typical size of the simplex would be
\[
\alpha = \sqrt{\frac{2}{\pi}} m \sigma_\theta.
\]

Therefore, when we make approximating assumptions on the integral \( J \), we have to keep in mind the scaling behavior of \( \alpha \) with respect to \( m \) and \( \sigma_\theta \).
3. Gaussian Integral

We now evaluate the integral $J$ (as defined in Eq. (E4)), which we use to derive asymptotic approximations for $E_{\Delta_m^{-1}}(\varphi^2)$ and $\text{Var}_{\Delta_m^{-1}}(\varphi^2)$ in the limit of large $m$.

**Proposition E.5.** We denote

$$k' = \sqrt{\frac{2}{\pi}} \left( 1 - 2 \sqrt{\frac{\log 2}{\pi - 2} + \pi \sqrt{\frac{\log 2}{\pi - 2}}} \right). \tag{E5}$$

In the limit of large $m$, we have the following linearization of $E_{\Delta_m^{-1}}(\varphi^2)$ and $\text{Var}_{\Delta_m^{-1}}(\varphi^2)$ around $k'$:

$$E_{\Delta_m^{-1}}(\varphi^2) \approx [0.727 + 0.376(k - k')] m\sigma^2,$$
$$\text{Var}_{\Delta_m^{-1}}(\varphi^2) \approx [0.887 + 0.813(k - k')] m\sigma^2.$$

**Proof.** We first note that

$$E_{\Delta_m^{-1}}(\varphi) = \frac{\sigma^2 J'}{J},$$

$$\text{Var}_{\Delta_m^{-1}}(\varphi^2) = \frac{3\sigma^2 J' + \sigma^2 J''}{J} = \frac{\sigma^2 (J')^2}{J^2}. \tag{E6}$$

where the prime symbol denotes partial derivative of $J$ with respect to $\sigma$.

We begin by transforming the integral $J(\alpha, \sigma)$ in frequency space $p$

$$J(\alpha, \sigma) = \frac{\sqrt{m}}{2\pi} \exp\left(-\frac{\alpha^2}{2m\sigma^2}\right) \times 
\int_{-\infty}^{\infty} dp \exp(-ipa) \left\{ \int_{0}^{\infty} d\theta \exp(ip\theta - \frac{\theta^2}{2\sigma^2} + \frac{\alpha\theta}{m\sigma^2}) \right\}^m \times 
\int_{-\infty}^{\infty} dp \exp(-\frac{1}{2}p^2\sigma^2m)(1 + \text{erf}\left(\frac{a + ipm\sigma^2}{\sqrt{2m\sigma^2}}\right))^m.$$

In order to approximate the error function, we denote $p' = \sqrt{\frac{2m\sigma^2}{\pi}}p$ and $\lambda = \frac{\alpha}{m\sigma}$, Note that $\lambda$ does not scale with $m$ or $\sigma$, and its typical value is $\frac{\alpha}{\sqrt{m}}$. The integral can then be written as

$$J(\alpha, \sigma) = (2^{\frac{m+1}{2}}\pi^{-\frac{1}{2}}e^{-\frac{m+1}{2}}) \times 
\int dp' \exp(-p'^2) \left( 1 + \text{erf}\left( \frac{ip'}{\sqrt{m}} + \frac{\lambda}{\sqrt{2}} \right) \right)^m. \tag{E7}$$

Note that for the argument of the error function, the real part is close to $\frac{\lambda}{\sqrt{2}} = \frac{1}{\sqrt{\pi}} \lambda < 1$, and the imaginary part approaches zero for large $m$. We then expand the error function as follows.

$$\text{erf}(x + iy) \approx \text{erf}(x) + \frac{2i}{\sqrt{\pi}} \exp(-x^2) y + \frac{2x}{\sqrt{\pi}} \exp(-x^2) y^2,$$

which gives us

$$\text{erf}\left( \frac{ip'}{\sqrt{m}} + \frac{\lambda}{\sqrt{2}} \right) \approx \text{erf}\left( \frac{\lambda}{\sqrt{2}} \right) + \frac{2}{\sqrt{\pi}} \exp\left(-\frac{\lambda^2}{4}\right) \left( \frac{ip'}{\sqrt{m}} \right) + \frac{2}{\sqrt{\pi}} \frac{\lambda}{\sqrt{2}} \exp\left(-\frac{\lambda^2}{4}\right) \left( \frac{ip'}{\sqrt{m}} \right)^2,$$

where we kept terms only up to the order of $\lim_{m \to \infty}(1 + \frac{1}{m})^m = 0$ as $m \to \infty$ for $r > 1$. We can then approximate the $m$-th power of the above result by using the fact that

\[ \lim_{m \to \infty} x^m \exp\left( \frac{a}{x} + \frac{b}{x^2} - \frac{1}{2}\frac{(b/x)^2}{x^2} \right) = 1, \]
which allows us to write
\begin{align*}
(1 + \text{erf}(\frac{i p'}{\sqrt{m}} + \frac{\lambda}{\sqrt{2}}))^m \\
\approx (1 + \text{erf}(\frac{\lambda}{\sqrt{2}}))^m \exp\left[\frac{2\sqrt{m}}{\sqrt{\pi}} C(\lambda) ip' + \sqrt{\frac{2}{\pi}} C(\lambda) \lambda p'^2 + \frac{2}{\pi} C(\lambda)^2 p'^2 \right],
\end{align*}
where we have denoted
\begin{equation*}
C(\lambda) = \frac{e^{-\lambda^2/2}}{1 + \text{erf}(\frac{\lambda}{\sqrt{2}})}.
\end{equation*}

We then make the following approximation to the integral:
\begin{equation*}
\int_{-\infty}^{\infty} dp \exp(-p^2) \exp(ap + bp^2) = \sqrt{\frac{\pi}{1 - b}} \exp\left(\frac{a^2}{4(1 - b)}\right).
\end{equation*}

We can then evaluate the integral \( J \) and perform the following linearization around \( k' \) (the reason for the choice of \( k' \) will be clear in the following subsection):
\begin{equation}
E_{\Delta^m \rightarrow (\varphi^2)} \approx (0.727 + 0.376(k - k')) m \sigma_\theta^2,
\end{equation}
\begin{equation}
\text{Var}_{\Delta^m \rightarrow (\varphi^2)} \approx (0.887 + 0.813(k - k')) m \sigma_\theta^2.
\end{equation}

\( \square \)

4. Density of States

We first briefly discuss the choice of \( k' \) as appeared in Eq. (E5). We first recall that the size of the simplex,
\begin{equation*}
\alpha = \sum_{j=1}^{m} |\theta_j|,
\end{equation*}
is the sum of \( m \) iid half-normal variables each with mean \( \sqrt{\frac{2}{\pi}} \sigma_\theta \) and variance \( (1 - \frac{2}{\pi}) \sigma_\theta^2 \). This means that in the limit of large \( m \), \( \alpha \) can be considered a normal variable with mean \( \sqrt{\frac{2}{\pi}} m \sigma_\theta \) and variance \( (1 - \frac{2}{\pi}) m \sigma_\theta^2 \). We then see that in the limit of large \( m \), \( \alpha \) is sharply peaked at its mean (as the relative standard deviation scales as \( \sqrt{\frac{1}{m}} \)), as the result of the LLN (law of large numbers). This means that the probability that \( \alpha \) deviates from its mean by some constant fraction scales as \( e^{-m} \).

However, this exponential decay is compensated by the exponential increase in the number of visible configurations, which is simply \( 2^m \). In fact, for an \( n \times n \) RBM, the contributions from the law of large numbers and entropy balance out, and a simplex whose size deviates from the typical value of \( \alpha \) can still be likely generated by some visible layer configuration. We formalize this argument as follows, where \( k = \frac{\alpha}{m \sigma_\theta} \) is taken to be a random variable with mean \( \sqrt{\frac{2}{\pi}} \) and variance \( \frac{1}{n}(1 - \frac{2}{\pi}) \).

**Definition E.2.** For a random \( n \times m \) RBM, we define its **density of states** at \( k, D(n, m, k) \), to be
\begin{equation*}
D(n, m, k) = \lim_{\delta k \to 0} \frac{E_W \left( N(k, k + \delta k) \right)}{\delta k},
\end{equation*}
where \( N(k_1, k_2) \) denotes the expected number (taken over the probability measure of the weight matrix) of visible configurations that generates a simplex whose size is from \( k m \sigma_\theta \) to \( (k + \delta k) m \sigma_\theta \).

**Remark.** For a \( n \times m \) random RBM, if we take the graphs of all the simplexes generated by all the visible configurations, \( D(n, m, k) \) is simply a measure of how “densely packed” the simplexes are at \( k \).

**Proposition E.6.** For an \( n \times n \) random RBM, we denote the density of states to be \( D(n, k) = D(n, n, k) \). If we let
\begin{align*}
k^* &= \sqrt{\frac{2}{\pi}} \left( 1 - (\pi - 2) \sqrt{\frac{\log 2}{\pi - 2}} \right), \\
k' &= \sqrt{\frac{2}{\pi}} \left( 1 + (\pi - 2) \sqrt{\frac{\log 2}{\pi - 2}} \right),
\end{align*}
Then ∀δk > 0,  
\[ \lim_{n \to \infty} D(n, k^\prime + δk) = 0 \quad \lim_{n \to \infty} D(n, k^\prime - δk) = 0. \]

Proof. We first note that
\[
D(n, k) = 2^n \frac{1}{\sqrt{2\pi (1 - \frac{k}{n})}} \exp \left[ -\frac{(k - \sqrt{\frac{2}{\pi}})}{\frac{2}{\pi} (1 - \frac{k}{n})} \right] n, \]

= \sqrt{\frac{n}{2\pi (1 - \frac{k}{n})}} \left[ \exp \left( \log(2) - \frac{(k - \sqrt{\frac{2}{\pi}})^2}{2(1 - \frac{k}{n})} \right) \right]^n.

Note that exponent evaluates to 0 at \( k = k^\prime \) or \( k = k^\prime \), and the exponent is negative if \( k > k^\prime \) or \( k < k^\prime \), so the proposition follows.

Remark. This proposition implies that for a random \( n \times n \) RBM, the largest size of the simplex generated by the visible configuration is typically \( k^\prime \), which corresponds to the mode of the joint distribution \( \nu^* \). It is convenient to denote the deviation from the size of the largest simplex as \( \kappa = k^\prime - k \), and the size difference between the smallest and largest simplexes as \( \delta \kappa = k - k^\prime \), then under this parameterization, we can write the density of states as
\[
D(n, \kappa) = \sqrt{\frac{n}{2\pi - 4}} \left[ \exp \left( ((\kappa)(\Delta \kappa - \kappa)) \right) \right] \exp \frac{n \kappa^2}{\pi}. \tag{E9}
\]

Then from Eqs. (E2), (E3), and (E8), we see that the conditional expected value and variance of \( P(\theta) \) can be linearized at \( \kappa = 0 \) as
\[
\mathbb{E}(P(\theta) \mid \Delta_{\alpha}^{m-1}) = \frac{(k\alpha)^2}{m} + \mathbb{E} \Delta_{\alpha}^{m-1}(\varphi^2 \mid k) \\
\approx (0.727 + k^2) - (0.376 + 2k^2)k_n \sigma^2 \theta
\]
\[ \sqrt{\text{Var}(P(\theta) \mid \Delta_{\alpha}^{m-1})} = \sqrt{\text{Var}(\varphi^2 \mid k)} \approx (0.942 - 0.432k)\sqrt{\pi \sigma^2}. \]

If we denote \( A = (0.376 + 2k^2) \) and \( B(k) = 0.942^2 + (0.942 - 0.432k)^2 \), then for sufficiently large \( \kappa > 0 \), we have the following approximation
\[
\Pr \left[ P(\kappa) > P(0) \right] = \frac{1}{2} \text{erfc} \left( C(n, \kappa) \right) \approx \frac{1}{2} \frac{1}{C(n, \kappa) \sqrt{\pi}} \exp \left[ -\frac{1}{2} C(n, \kappa)^2 \right]
\]
where we have denoted
\[
C(n, \kappa) = \frac{\sqrt{\pi} A \kappa}{\sqrt{2B(k)}},
\]
noting that it scales with \( \sqrt{\pi} \). Then clearly, \( \forall \delta \kappa > 0 \), we have
\[
\lim_{m \to \infty} \frac{\text{erfc}(C(n, \kappa))}{C(n, \kappa) \sqrt{\pi}} \exp \left[ -\frac{1}{2} C(n, \kappa)^2 \right],
\]
meaning that the asymptotic approximation to the erfc function is valid in the limit of large \( n \).

If we denote an instance with the random variable \( P(\theta) \) conditioned on the simplex \( \sum_j |\theta_j| = (k^\prime - \kappa)n \sigma \theta \) as \( P(\kappa) \), then we can make the following approximation to Eq. (E2).
\[
\log \left[ \prod_y \Pr(P(\nu) \leq P(\nu^*)) \right] \\
= \sum_y \log \left[ \Pr[P(\nu) \leq P(\nu^*)] \right] \\
\approx -\int_{\delta \kappa}^{\infty} d\kappa D(n, \kappa) \frac{1}{2} \frac{1}{C(n, \kappa) \sqrt{\pi}} \exp \left[ -\frac{1}{2} C(n, \kappa)^2 \right],
\]
where \( \delta \kappa > 0 \) is some small constant. We formalize this approximation as follows.
Proposition E.7. Let $\delta \kappa = \frac{1}{\pi}$, then in the limit of large $n$,

$$\int_0^{+\infty} d\kappa D(n, \kappa) \log \left[ \Pr(P(\kappa) \leq P(0)) \right] \approx -\int_0^{-\delta \kappa} d\kappa D(n, \kappa) \frac{1}{2} \frac{1}{C(n, \kappa) \sqrt{\pi}} \exp \left[ -C(n, \kappa)^2 \right].$$

Proof. We denote the following integral

$$I(n, \kappa_1, \kappa_2) = \int_{\kappa_1}^{\kappa_2} d\kappa D(n, \kappa) \log \left[ \Pr(P(\kappa) \leq P(0)) \right]. \tag{E11}$$

First thing to note is that the integrand is always negative as the density of states, $D(n, \kappa)$, is necessarily positive, and the log-likelihood is necessarily negative. We then break $I(n, -\infty, +\infty)$ into three parts:

$$I(n, 0, +\infty) = I(n, 0, +\delta \kappa) + I(n, +\delta \kappa, +\infty).$$

To prove the proposition, it is sufficient to show in the limit of large $n$ that the integral goes to zero for the first and last terms, and the asymptotic approximation for the error function is valid for the second term.

For the integral $I(n, +\delta \kappa, +\infty)$, we have $\kappa \geq +\delta \kappa$, and we can approximate the log-likelihood as

$$\log \left( \Pr(P(\kappa) \leq P(0)) \right) = \log \left( 1 - \Pr(P(\kappa) > P(0)) \right) \approx -\frac{1}{2} \text{erfc} \left( \frac{\sqrt{n}A\kappa}{\sqrt{2B(\kappa)}} \right) \rightarrow 0,$$

as $n$ goes to infinity, meaning that the erfc approximation is valid.

For the integral $I(n, 0, +\delta \kappa)$, we have $\kappa \geq +\delta \kappa$, and we obtain the following

$$\log \left( \Pr(P(\kappa) \leq P(0)) \right) \geq \log \left( \frac{1}{2} \right) \approx -0.69.$$

Recall that

$$D(n, \kappa) = \sqrt{\frac{n}{2\pi - 4}} \exp \left( (\kappa)(\Delta \kappa - \kappa) \right) \frac{n}{(\kappa - \Delta \kappa)^{n+4}},$$

which means

$$|I(n, 0, +\delta \kappa)| \geq 0.69 \int_0^{+\delta \kappa} d\kappa D(n, \kappa) \rightarrow 0,$$

as we take $n$ to infinity. The proposition is then shown.

Corollary E.7.1. For sufficiently small values of $n$, the joint and marginal modes of a random $n \times n$ RBM are strongly correlated, under definition $D.2$.

Proof. This can be shown by directly evaluating the logarithm of the integral as given in Eq. (E10), and verify that the result is greater than $\log(\frac{1}{2})$, up to a certain value of $n_{\text{max}}$. \qed
9 A. Galloway, A. Golubeva, T. Tanay, M. Moussa, and G. W. Taylor, arXiv preprint arXiv:1905.02161 (2019).
10 D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, Journal of Machine Learning Research 11, 625 (2010).
11 H. Manukian, F. L. Traversa, and M. Di Ventra, Neural Networks 110, 1 (2019).
12 M. Di Ventra and Y. V. Pershin, Nature Physics 9, 200 (2013).
13 F. L. Traversa and M. Di Ventra, Chaos: An Interdisciplinary Journal of Nonlinear Science 27, 023107 (2017).
14 M. Di Ventra and F. L. Traversa, J. Appl. Phys. 123, 180901 (2018).
15 C. Sminchisescu and M. Welling, in Artificial Intelligence and Statistics (2007) pp. 516–523.
16 S. Lan, J. Streets, and B. Shahbaba, in AAAI (2014) pp. 1953–1959.
17 S. Arora and B. Barak, Computational Complexity: A Modern Approach (Cambridge University Press, 2009).
18 F. L. Traversa, P. Cicotti, F. Sheldon, and M. Di Ventra, Complexity 2018 (2018).
19 F. Sheldon, F. L. Traversa, and M. Di Ventra, arXiv preprint arXiv:1810.03712 (2018).
20 J. Melchior, A. Fischer, and L. Wiskott, The Journal of Machine Learning Research 17, 3387 (2016).
21 A. Fischer and C. Igel, in Iberoamerican congress on pattern recognition (Springer, 2012) pp. 14–36.
22 P. M. Long and R. A. Servedio, (2010).
23 M. A. Carreira-Perpinan and G. E. Hinton, in Aistats, Vol. 10 (Citeseer, 2005) pp. 33–40.
24 Y. R. Pei, H. Manukian, and M. Di Ventra, arXiv preprint arXiv:1905.05334 (2019).
25 S. R. Bearden, F. Sheldon, and M. Di Ventra, EPL (Europhysics Letters) 127, 30005 (2019).
26 D. J. MacKay, Information theory, inference and learning algorithms (Cambridge university press, 2003).
27 G. Hinton, Momentum 9, 926 (2010).
28 F. L. Traversa and M. Di Ventra, Chaos: An Interdisciplinary Journal of Nonlinear Science 27, 023107 (2017).
29 This is a justified assumption if the system is highly frustrated, as the energy gaps near the ground state are generally very small for such system.
30 From here on, ‘iid’ will serve as the abbreviation of ‘independent and identically distributed’.