Exploring the Energy Landscape of RBMs: Reciprocal Space Insights into Bosons, Hierarchical Learning and Symmetry Breaking

J. Quetzalcóatl Toledo-Marin[1,2,*], Anindita Maiti[2], Geoffrey C. Fox[3], and Roger G. Melko[4,2]

[1]TRIUMF, Vancouver, BC V6T 2A3, Canada
[2]Perimeter Institute for Theoretical Physics, Waterloo, Ontario, N2L 2Y5, Canada
[3]University of Virginia, Computer Science and Biocomplexity Institute, 994 Research Park Blvd, Charlottesville, Virginia, 22911, USA
[4]Department of Physics and Astronomy, University of Waterloo, Ontario, N2L 3G1, Canada

March 28, 2025

Abstract

Deep generative models have become ubiquitous due to their ability to learn and sample from complex distributions. Despite the proliferation of various frameworks, the relationships among these models remain largely unexplored, a gap that hinders the development of a unified theory of AI learning. In this work, we address two central challenges: clarifying the connections between different deep generative models and deepening our understanding of their learning mechanisms. We focus on Restricted Boltzmann Machines (RBMs), a class of generative models known for their universal approximation capabilities for discrete distributions. By introducing a reciprocal space formulation for RBMs, we reveal a connection between these models, diffusion processes, and systems of coupled bosons. Our analysis shows that at initialization, the RBM operates at a saddle point, where the local curvature is determined by the singular values of the weight matrix, whose distribution follows the Marčenko-Pastur law and exhibits rotational symmetry. During training, this rotational symmetry is broken due to hierarchical learning, where different degrees of freedom progressively capture features at multiple levels of abstraction. This leads to a symmetry breaking in the energy landscape, reminiscent of Landau's theory. This symmetry breaking in the energy landscape is characterized by the singular values and the weight matrix eigenvector matrix. We derive the corresponding free energy in a mean-field approximation. We show that in the limit of infinite size RBM, the reciprocal variables are Gaussian distributed. Our findings indicate that in this regime, there will be some modes for which the diffusion process will not converge to the Boltzmann distribution. To illustrate our results, we trained replicas of RBMs with different hidden layer sizes using the MNIST dataset. Our findings not only bridge the gap between disparate generative frameworks but also shed light on the fundamental processes underpinning learning in deep generative models.

1 Introduction

Generative models are ubiquitous as they have emerged as powerful tools across multiple domains. The ongoing sprint to better models has led to a plethora of frameworks. In data-driven contexts, these models are designed to learn and replicate the underlying probability distributions of complex datasets. In particular, they have found significant applications in condensed matter and nuclear physics, where neural network-based ansätze are employed to model phase transitions and approximate ground-state wavefunctions[1, 2, 3, 4, 5, 6].

*Corresponding author: jtoledo@triumf.ca

Similarly, generative models are employed in high-energy physics to address multiple problems, *e.g.*, shower generation, jet generation and unfolding [7, 8, 9, 10]. Although generative AI techniques have achieved notable success in natural language processing and computer vision, our theoretical understanding of their learning dynamics and performance remains incomplete. Critical questions, such as the precise mechanisms through which these models capture data distributions, the robustness and fidelity of their approximations, and the criteria that determine the suitability of specific frameworks for particular datasets, remain largely unresolved. Addressing these issues is not merely an academic exercise; rather, it is essential for developing reliable and predictive generative models for complex physical systems.

Different architectures trained with different training techniques on different datasets share similar behaviors, ultimately suggesting a type of universality in generative models that goes beyond the details of each framework. In Ref. [11] the authors discuss universality in deep learning in three different directions, *i.e.*, the hierarchy of learning dynamics, of model complexity and of neural representation and relate these to parameter symmetry breaking. On the latter direction, it has been shown that representations of learned models are found to be universally aligned to different models trained on similar datasets [12]. It then begs the question *are different generative AI frameworks equivalent?* To be more precise, given the fact that different generative frameworks, *e.g.*, diffusion models, restricted Boltzmann machines (RBM), generative adversarial networks (GAN) and variational auto-encoders (VAE) among others, are capable of learning the underlying distribution of the same dataset, in spite of having different architectures, training schemes and frameworks, is there an equivalence between generative AI frameworks? It is certainly the case that generative models, in general, are trained via optimizing the log-likelihood. For instance, the decoder in a VAE and the generator in a GAN both serve the role of mapping a latent representation to the dataset space. Furthermore, VAEs [13] are trained via the evidence lower bound which is composed by a reconstruction term and the Kullback-Liebler (KL) divergence, whereas the initial GAN framework [14] used the Jensen divergence for training. However, GANs are trained in an adversarial way, such that the generator learns the "likelihood" implicitly via the discriminator's feedback, while the VAE learns the likelihood explicitly.

Diffusion models were first introduced as a type of hierarchical VAE, trained by optimizing a variational bound, similar to a VAE regularizer [15]. However, diffusion models outperform most frameworks in most tasks. Once the diffusion model is trained, the sampling process consists of first generating a Gaussian distributed random vector and then denoising it via the reverse diffusion process. This process is similar in spirit to the sampling process in an RBM.

RBMs are universal approximators of discrete distributions [16] and rely on computationally intensive Monte Carlo Markov Chain (MCMC) methods for training and sampling. By introducing a reciprocal space formulation for RBMs, we reveal a connection between RBMs, diffusion processes, and systems of coupled Bosons. Our analysis shows that at initialization, the RBM operates at a saddle point, where the local curvature is determined by the singular values of the weight matrix, whose distribution follows the Marčenko-Pastur law and exhibits rotational symmetry. During training, this rotational symmetry is broken due to hierarchical learning, where different degrees of freedom progressively capture features at multiple levels of abstraction. This leads to a symmetry breaking in the energy landscape, reminiscent of Landau's theory. This symmetry breaking in the energy landscape is characterized by the singular values and the weight matrix eigenvector matrix. We derive the corresponding free energy in a mean-field approximation. We show that in the limit of infinite size RBM, the reciprocal variables are Gaussian distributed. Our findings indicate that in this regime, there will be some modes for which the diffusion process will not converge to the Boltzmann distribution. We demonstrate these phenomena with an RBM and MNIST and further discuss how various initialization strategies can influence training dynamics. Our findings not only bridge the gap between disparate generative frameworks but also shed light on the fundamental processes underpinning learning in deep generative models.

The paper is organized as follows: The next section presents a brief introduction to RBMs and how they are typically trained; section 3 introduces the reciprocal space, the energy landscape and the singular value distribution; section 4 shows the connection between RBMs with diffusion processes and systems of coupled Bosons; section 5 presents the symmetry breaking in parameter space and its connection with symmetry breaking in the energy landscape; the last section is devoted to conclusions and outlook.

2 A brief introduction to Restricted Boltzmann Machines

Consider a dataset $\{\boldsymbol{v}^{(\alpha)}\}_{\alpha=1}^{|\mathcal{D}|}$, where each data point $\{0,1\}^N$ is an N-dimensional binary vector. Our goal is to approximate the empirical data distribution with a Boltzmann distribution, $p(\boldsymbol{v})$. This is achieved by

maximizing the log-likelihood (LL) of the model, $p(\boldsymbol{v})$, over the data set. Let us denote $P_{\mathcal{D}} = \left(\prod_{\boldsymbol{v}\in\mathcal{D}} p(\boldsymbol{v})\right)^{1/\mathcal{D}}$. Maximizing the LL corresponds to:

$$\text{argmax}_{\Theta} \ln P_{\mathcal{D}} \tag{1}$$

By design, $p(\boldsymbol{v})$ is a Boltzmann distribution *viz.*

$$p(\boldsymbol{v}) = \frac{\sum_{\{\boldsymbol{h}\}} e^{-E(\boldsymbol{v},\boldsymbol{h};\boldsymbol{a},\boldsymbol{b},\boldsymbol{W})}}{Z} \ , \tag{2}$$

where $Z = Z(\boldsymbol{a},\boldsymbol{b},\boldsymbol{W})$ is the partition function, and $E(\boldsymbol{v},\boldsymbol{h};\boldsymbol{a},\boldsymbol{b},\boldsymbol{W})$ is the energy function defined as

$$E(\boldsymbol{v},\boldsymbol{h};\boldsymbol{a},\boldsymbol{b},\boldsymbol{W}) = -\sum_{i}^{N} a_i v_i - \sum_{j}^{M} b_j h_j - \sum_{i,j} v_i W_{ij} h_j \ . \tag{3}$$

The parameters $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{W}$ are trainable fitting parameters and $\boldsymbol{h}$ is the hidden layer, with $\boldsymbol{h} \in \{0,1\}^M$. Note that the matrix $\boldsymbol{W}$ couples nodes in the visible layer, $\boldsymbol{v}$, with the nodes in the hidden layer, $\boldsymbol{h}$, and there are no explicit couplings among nodes in the visible or hidden layers, which is the same to say that the RBM is a bipartite graph (*i.e.*, restricted).

To maximize the LL, we can use stochastic gradient descent. We therefore compute the gradient of the LL with respect to some generic parameter Θ:

$$\frac{\partial \ln P_{\mathcal{D}}}{\partial \Theta} \quad = \quad \frac{1}{|D|}\sum_{\alpha=1}^{|D|} \Big\langle -\frac{\partial E}{\partial \Theta} \Big\rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\boldsymbol{\alpha})})} - \Big\langle -\frac{\partial E}{\partial \Theta} \Big\rangle_{p(\boldsymbol{v},\boldsymbol{h})} \tag{4}$$

with

$$\frac{\partial E}{\partial \Theta} = \begin{cases} -v_k, & \Theta = a_k, \\ -h_k, & \Theta = b_k, \\ -v_k h_l, & \Theta = W_{kl} \ . \end{cases} \tag{5}$$

The LL gradient simplifies to

$$\frac{\partial \ln P_{\mathcal{D}}}{\partial a_k} = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \langle v_k^{(\alpha)} \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} - \langle v_k \rangle_{p(\boldsymbol{v},\boldsymbol{h})} \tag{6a}$$

$$\frac{\partial \ln P_{\mathcal{D}}}{\partial b_k} = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \langle h_k \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} - \langle h_k \rangle_{p(\boldsymbol{v},\boldsymbol{h})} \tag{6b}$$

$$\frac{\partial \ln P_{\mathcal{D}}}{\partial W_{kl}} = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \langle v_k^{(\alpha)} h_l \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} - \langle v_k h_l \rangle_{p(\boldsymbol{v},\boldsymbol{h})} \tag{6c}$$

where index k and α, respectively, refer to the RBM node and dataset point. We further have introduced the notation:

$$\langle \bullet \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} = \frac{\sum_{\{\boldsymbol{h}\}} \bullet \ e^{-E(\boldsymbol{v}^{(\alpha)},\boldsymbol{h})}}{\sum_{\{\boldsymbol{h}\}} e^{-E(\boldsymbol{v}^{(\alpha)},\boldsymbol{h})}} \tag{7}$$

and

$$\langle \bullet \rangle_{p(\boldsymbol{v},\boldsymbol{h})} = \frac{\sum_{\{\boldsymbol{v},\boldsymbol{h}\}} \bullet \ e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\sum_{\{\boldsymbol{v},\boldsymbol{h}\}} e^{-E(\boldsymbol{v},\boldsymbol{h})}} \ . \tag{8}$$

The first terms in Eqs. (6) can be further simplified to:

$$\frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \langle v_k^{(\alpha)} \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} v_k^{(\alpha)} \ , \tag{9a}$$

$$\frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \langle h_k \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \sigma(C_k(\boldsymbol{v}^{(\alpha)})) \ , \tag{9b}$$

$$\frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} \langle v_k^{(\alpha)} h_l \rangle_{p(\boldsymbol{h}|\boldsymbol{v}^{(\alpha)})} = \frac{1}{|\mathcal{D}|}\sum_{\alpha=1}^{|\mathcal{D}|} v_k^{(\alpha)} \sigma(C_l(\boldsymbol{v}^{(\alpha)})) \ . \tag{9c}$$

Since the number of states for $\boldsymbol{v}$ and $\boldsymbol{h}$ are 2^N and 2^M, respectively, the number of terms in the sums in Eq. (8) is 2^{N+M}. This exponential dependence on the dimensionality makes computing the expectation values over $p(\boldsymbol{v}, \boldsymbol{h})$ intractable for large N and M. To overcome this limitation, importance sampling is used. Note that $q(\boldsymbol{h}|\boldsymbol{v}) = p(\boldsymbol{v}, \boldsymbol{h})/p(\boldsymbol{v})$, from which it is straightforward to show that

$$q(\boldsymbol{h}|\boldsymbol{v}) = \prod_{j=1}^{M} q(h_j|\boldsymbol{v}) \ , \tag{10}$$

where

$$q(h_j|\boldsymbol{v}) = \frac{e^{h_j C_j(\boldsymbol{v})}}{1 + e^{C_j(\boldsymbol{v})}} \tag{11}$$

and $C_j(\boldsymbol{v}) = \sum_i v_i W_{ij} + b_j$. Thus, the probability of node $h_j = 1$ is given by $q(h_j = 1|\boldsymbol{v}) = \sigma(C_j(\boldsymbol{v}))$, and similarly for $v_i = 1$ the probability yields $p(v_i = 1|\boldsymbol{h}) = \sigma(D_i(\boldsymbol{h}))$, with $D_i(\boldsymbol{h}) = \sum_j W_{ij} h_j + a_i$. The expressions $\sigma(D_i(\boldsymbol{h}))$ and $\sigma(C_j(v))$ are used for importance sampling. Note that the joint probability can be expressed as $p(\boldsymbol{v}, \boldsymbol{h}) = p(\boldsymbol{v}|\boldsymbol{h})p(\boldsymbol{h})$ and $p(\boldsymbol{v}, \boldsymbol{h}) = q(\boldsymbol{h}|\boldsymbol{v})p(\boldsymbol{v})$. Thus, we can assume a prior $p(\boldsymbol{v})$ from where we generate hidden samples $\{\boldsymbol{h}\}$ via $q(\boldsymbol{h}|\boldsymbol{v})$, each sample in $\{\boldsymbol{h}\}$ is then used to generate visible samples, $\boldsymbol{v}$. Repeating this process K times yields a sequence of block Gibbs sampling steps. We denote the overall sampling process as $\{\boldsymbol{v}, \boldsymbol{h}\} \sim \prod_{i=1}^{K} p(\boldsymbol{v}^{(i)}|\boldsymbol{h}^{(i)}) q(\boldsymbol{h}^{(i)}|\boldsymbol{v}^{(i-1)}) p(v^{(0)})$. Here the latin index i should not be mistaken with the Greek index α used to tag the dataset points. For large K, the samples can be considered to come from the stationary distribution $p(\boldsymbol{v}, \boldsymbol{h})$. We repeat the importance sampling $\mathcal{N}$ times and build the estimator for the expectation value in Eq. (8) by the arithmetic average over the $\mathcal{N}$ samples generated via block Gibbs sampling:

$$\langle \bullet \rangle_{p(\boldsymbol{v},\boldsymbol{h})} \approx \frac{1}{\mathcal{N}} \sum_{\{\boldsymbol{v},\boldsymbol{h}\} \sim \mathrm{BGS}} \bullet \ . \tag{12}$$

The Gibbs sampling number of steps is commonly on the orders of 10^2, and it has been shown that as the number of updates during training increases, the number of Gibbs sampling steps must also be increased for the RBM to reach equilibrium rather than becoming stuck in an non-equilibrium state [17].

The standard procedure to train and RBM involves partitioning the data set $\mathcal{D}$ into mini-batches $\mathcal{D}_\chi$, such that $\mathcal{D} = \cup_\chi \mathcal{D}_\chi$ and $\cap \mathcal{D}_\chi = \varnothing$. The RBM parameters are then updated according to:

$$a_k^{(t)} = a_k^{(t-1)} + \eta \frac{\partial \ln P_{\mathcal{D}_\chi}}{\partial a_k} \ , \tag{13a}$$

$$b_k^{(t)} = b_k^{(t-1)} + \eta \frac{\partial \ln P_{\mathcal{D}_\chi}}{\partial b_k} \ , \tag{13b}$$

$$W_{kl}^{(t)} = W_{kl}^{(t-1)} + \eta \frac{\partial \ln P_{\mathcal{D}_\chi}}{\partial W_{kl}} \ , \tag{13c}$$

where η is the learning rate. Three primary training strategies are discussed in the literature, each differing mainly on how the Markov chain in Eq. (12) is initialized. *Rdm-K*: For each parameter update, the initial $\boldsymbol{h}$ is randomly sampled from a Bernoulli distribution. Here K is the number of block Gibbs sampling steps. *Contrastive Divergence*: In CD, a data point from the training dataset is used as the initial $\boldsymbol{v}$ vector for the block Gibbs sampling. This method improves performance in RBMs compared to Rdm-K [17]. *Persistent contrastive divergence*: This method is similar to CD where the Markov chain is started using a data point in the data set for the first parameter update, while for the remaining parameter updates, the Markov chains are initialized using the last state in the previous parameter update. Although training with PCD can be challenging, when executed correctly it achieves superior results compared to CD [18]. Additional techniques, such as centering the RBM gradient and incorporating regularizers, have also been proposed to enhance performance [19, 20, 21]

All the numerical results regarding trained RBMs presented in this paper were obtained by training on MNIST dataset and are available in our Github repository [22]. The RBM training was done using the publicly available Julia code, as described in Ref. [18]. We trained five replicas for each RBM with hidden layer size $500, 784, 1200$ and 3000. In Fig. 1 left panel we show the log-likelihood *vs* epochs to verify the maximization of the log-likelihood. We estimated the partition function using annealed importance sampling and reverse annealed importance sampling, which correspond to an upper and lower bound of the partition function [23, 24].

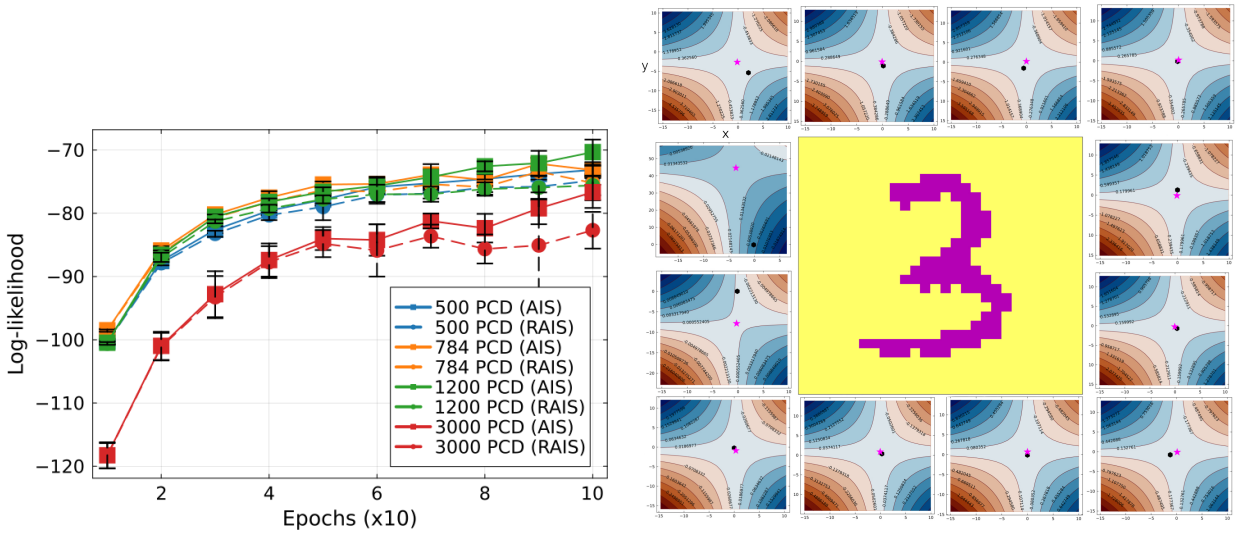

Figure 1: **Left panel)** Log-likelihood *vs* epochs for RBMs with hidden layer sizes $M = 500, 784, 1200, 3000$. The partition function was estimated using annealed importance sampling (AIS) and reverse AIS [23, 24]. Each data point corresponds to the average over five replicas and the error bars correspond to the standard deviation. **Right panel)** Image of number 3 generated from a trained RBM and, via the transformations in Eq. (17a), projected onto the energy landscapes (clockwise starting at the upper left corner) $1, 2, 3, 4, 20, 30, 50, 100, 200, 300, 498$ and 500. The magenta star marks the saddle point whereas the black pentagon corresponds to the image projection to reciprocal space.

3 Restricted Boltzmann machine in reciprocal space

Updating the weight matrix and the self-fields via Eqs. (13c) ultimately reduces the energy associated to the dataset points via the energy function (see Eq. (3)). The energy landscape described by the energy function has at most as many saddle points as there are units in the smaller partition. At initialization, the dataset points cluster around these saddle points; however, as training progresses, they deviate from the saddle points but never drift away, due to implicit constraint imposed by the binary nature of the variables. To show this, we project the RBM onto reciprocal space using singular value decomposition (SVD). By studying the eigenvector matrices and singular values of the weight matrix, we learn about the weight matrix properties and provide a physical interpretation of the singular values in the context of RBMs.

In general, the coupling matrix $\boldsymbol{W}$ is a rectangular, non-symmetric random matrix with Gaussian entries and standard deviation σ. We perform SVD on $\boldsymbol{W}$, such that $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^t = \sum_\alpha \lambda_\alpha |v_\alpha\rangle\langle h_\alpha|$; where $\boldsymbol{U}$ is a matrix composed of the eigenvectors of $\boldsymbol{W}\cdot\boldsymbol{W}^t$, *i.e.*, $\boldsymbol{U} = \sum_s |v_s\rangle\langle c_s|$, where $\{|c_s\rangle\}$ denotes the canonical basis; Σ is a rectangular matrix of size $N \times M$ containing the singular values $\{\lambda_i\}_{i=1}^{min(N,M)}$ on the diagonal; and $\boldsymbol{V}^t$ is a matrix composed of the eigenvectors of $\boldsymbol{W}^t\cdot\boldsymbol{W}$, *i.e.*, $\boldsymbol{V}^t = \sum_s |c_s\rangle\langle h_s|$. Note that due to the orthogonality of $\boldsymbol{U}$ and $\boldsymbol{V}$, we have $\boldsymbol{U}\cdot\boldsymbol{U}^t = \boldsymbol{V}\cdot\boldsymbol{V}^t = \boldsymbol{I}$. Moreover, the eigenvector matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are Haar distributed, *i.e.*, they are uniformly distributed over the group $O(N)$ and each eigenvector is uniformly distributed on the unit hypersphere [25]. The training process breaks this symmetry, as we will show later.

Using the matrices $\boldsymbol{U}$ and $\boldsymbol{V}^t$, we can express the energy function, Eq. (3), as:

$$E(\boldsymbol{v},\boldsymbol{h}) \to E(\boldsymbol{x},\boldsymbol{y}) = -\langle x|a_0\rangle - \langle b_0|y\rangle - \langle x|\boldsymbol{\Sigma}|y\rangle \ . \tag{14}$$

Expressing the energy in terms of the singular value decomposition of the coupling matrix recasts the model in terms of quasi-discrete variables with single pairwise interactions. We say quasi-discrete variable because each x_i (y_j) can take on values formed by summing subsets of elements in column i-th (j-th) of matrix U (V). The single pairwise interactions allows us to decompose the energy as a sum of individual pairwise energy terms:

$$E(\boldsymbol{x},\boldsymbol{y}) = \sum_i E_i(x_i, y_i) \ , \tag{15}$$

such that

$$E_i(x_i, y_i) = \begin{cases} -a_{0i}x_i - b_{0i}y_i - \lambda_i x_i y_i, & 1 \leq i \leq min(N, M) \\ -a_{0i}x_i, & min(N, M) < i \leq max(N, M) \end{cases} \tag{16}$$

where

$$\begin{cases} \boldsymbol{a_0} = \boldsymbol{U}^t \boldsymbol{a} \\ \boldsymbol{b_0} = \boldsymbol{V}^t \boldsymbol{b} \\ \boldsymbol{x} = \boldsymbol{U}^t \boldsymbol{v} \\ \boldsymbol{y} = \boldsymbol{V}^t \boldsymbol{h} \end{cases} \tag{17a}$$

It is instructive to consider a dynamical system described by the Hamiltonian $E_i(x_i, y_i)$ [26]. In this scenario, the conjugate momenta are $p_{x_i} = a_{0i} + \lambda_i y_i$ and $p_{y_i} = b_{0i} + \lambda_i x_i$, respectively. Setting these momenta to zero yields the fixed point of the system. From the Hessian matrix we find that the eigenvalues correspond to $\pm\lambda_i$, indicating that the fixed point is a saddle point. Each of the $E_{i(\leq min(N,M))}$ corresponds to a hyperbolic paraboloid with saddle point at $(x_{0i}, y_{0i}) = (-b_{0i}/\lambda_i, -a_{0i}/\lambda_i)$ where the energy is equal to $b_{0i}a_{0i}/\lambda_i$. In other words, for each energy landscape $E_{i(\leq min(N,M))}$ the phase curves are such that there is always a direction where the energy decreases and tends to $-\infty$.

In Fig. 1 we show the projection of an image of number 3 generated from a trained RBM using the transformations in Eq. (17a). The energy landscapes are ordered clockwise starting from the upper left corner for indices $i = 1, 2, 3, 4, 20, 30, 50, 100, 200, 300, 498$ and 500. The magenta star marks the saddle point, whereas the black pentagon corresponds to the image projection to reciprocal space.

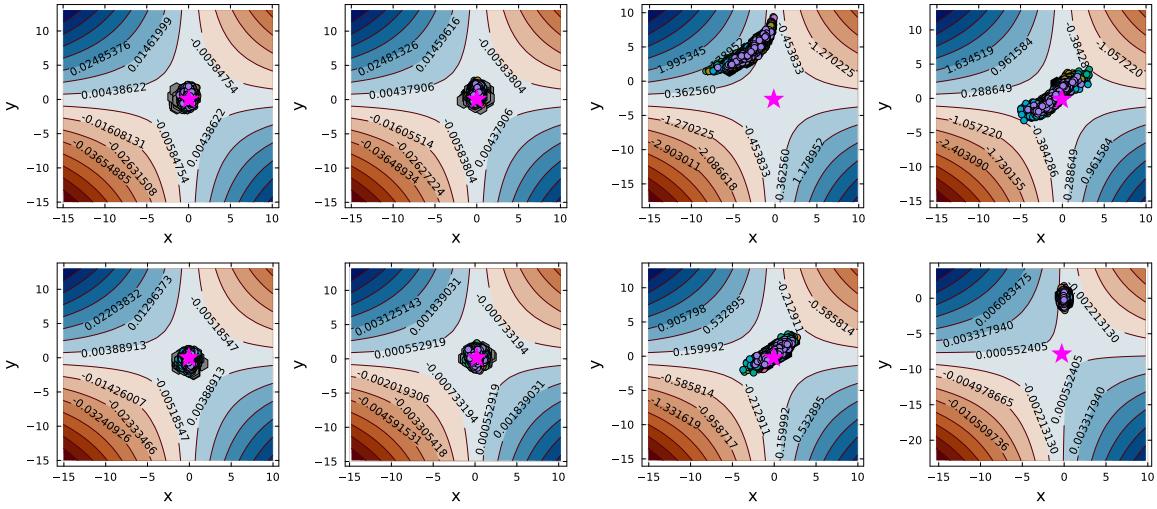

Figure 2: Each subpanel shows the xy-plane contour energy E_i for the singular value index 1,2,3 and 498. The colored data points correspond to MNIST test data projected onto the reciprocal space. The gray hexagons correspond to Gibbs sampled data. The magenta star marks the saddle point. **Left panel)** Randomly initialized RBM. **Right panel)** Trained RBM.

We further introduce a new set of variables $\{u_i\}_{i=1}^{\min(N,M)}, \{w_i\}_{i=1}^{\min(N,M)}$ which relate to x_i, y_i for $1 \leq i \leq min(N, M)$ via a translation to the saddle points and a rotation, namely,

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \begin{pmatrix} x_{0i} \\ y_{0i} \end{pmatrix} + \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} u_i \\ w_i \end{pmatrix} \tag{18}$$

Setting $\theta = \pi/4$ aligns the principal axis of the hyperbolic paraboloid with the uw-plane axis, leading to the energy function having the following form:

$$E_i(u_i, w_i) = \frac{a_{0i}b_{0i}}{\lambda_i} - \frac{\lambda_i}{2}(u_i^2 - w_i^2)\ . \tag{19}$$

We can now give a physical interpretation of the weight matrix singular values. The singular values can be interpreted as the vibrational modes, since the curvature of each mode per energy landscape is $\pm\lambda$. In

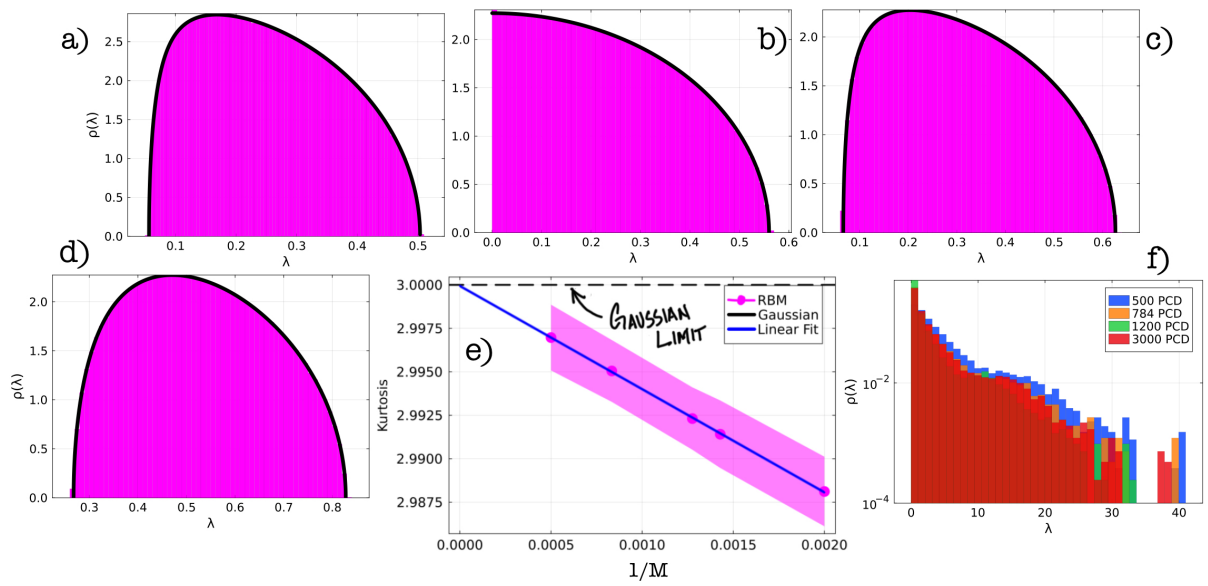

Figure 3: Singular values probability density function for a random RBM with 784 visible nodes and with **a)** 500, **b)** 784, **c)** 1200 and **d)** 3000 hidden nodes. **e)** Kurtosis of the reciprocal variable y for hidden layer sizes $M = 500, 700, 784, 1200, 2000$ for randomly initialized RBMs, with 784 visible nodes. Each data point was generated from averaging over the kurtosis of the hidden nodes with non-zero singular values. The kurtosis for each node was computed from projecting $7 \cdot 10^6$ binary vector samples to reciprocal space. The ribbon corresponds to the standard deviation over the hidden layer. The dashed line marks the target for Gaussianity. The blue curve is a linear fit extrapolated to hidden layer size infinity where the reciprocal variable becomes Gaussian. The linear fit residue is on the order of $\mathcal{O}(10^{-4})$ and has been included. **f)** Distribution of singular values for trained RBM with different hidden layer sizes and different training methods.

addition to modulating the stiffness of the mode, the singular values modulate the saddle point energy and its location. In Fig. 2 we show four energy landscapes in the xy-plane corresponding to the singular value indices $1, 2, 3$ and 498 in the case of a trained RBM, and we also project the test set onto these landscapes. We do this for an untrained and a trained RBM. In the former, the dataset points cluster on the saddle point, whereas for the trained RBM this is not always the case although the points do not drift away from the saddle point.

In practice, several empirical guidelines are used for initializing RBMs, many of which are supported by experimental evidence. For example, selecting the number of nodes in the hidden layer remains largely heuristic. A common recommendation is to limit the hidden layer to no more nodes than the visible layer, as exceeding this threshold may slow down training due to the increase in the number of operations per matrix multiplication. An additional reason to not have the hidden layer be much greater the the visible nodes is related to the singular values distribution. As we have shown previously, the singular values influence the stability of the energy landscape. The singular value spectrum is characterized by the Marčenko–Pastur law [25]:

$$\varrho(\lambda) = \frac{\sqrt{(\lambda_+^2 - \lambda^2)(\lambda^2 - \lambda_-^2)}}{\pi q \lambda} \qquad \lambda_- < \lambda < \lambda_+ \,, \tag{20}$$

with $\lambda_\pm = \sqrt{\max(N, M)\sigma^2} \pm \sqrt{q}$ and $q = \min(N, M)\sigma^2$. In Fig. 3 we plot Eq. (20) for various RBM sizes. One can demonstrate that the maximum of $\varrho(\lambda)$ occurs at $\lambda_{max} = \sqrt{\lambda_+ \lambda_-}$, implying $\lambda_{max} = \sigma\sqrt{|N - M|}$. This distribution increases with the difference between visible and hidden nodes as well as with increasing standard deviation σ. Consequently, a significant imbalance between the numbers of visible and hidden nodes and/or a broad range of coupling values accentuates the singular value distribution, thereby impacting the stability of the energy landscapes at initialization.

It is interesting to note that while the energy is minimized as $(x_i, y_i) \to \pm(\infty, \infty)$, the dataset clusters around the saddle point, suggesting an implicit constraint during training. This property ultimately comes from the

constraints imposed by the binary space. In the next section we obtain these constraints as a potential in reciprocal space.

4 Partition function in reciprocal space

In the previous section we presented the RBM in reciprocal space, we gave a physical interpretation of the singular values and we described the general structure of the energy landscape of the RBM. We also mentioned that the implicit constraints imposed by the binary nature of the variables confine the dataset points to regions near the saddle points. In this section we treat the reciprocal variables as continuous while enforcing the constraints imposed by the binary nature of the variables as a potential acting on these degrees of freedom. This approach allows us to map an RBM to a diffusion process and, subsequently, to a Schrödinger equation describing the interaction of Bosons.

We first construct the partition function of the reciprocal space in terms of the variables $\{u_i\}_{i=1}^{\min(\mathrm{N},\mathrm{M})}, \{w_i\}_{i=1}^{\min(\mathrm{N},\mathrm{M})}$ and $\{x_j\}_{j=\min(N,M)+1}^{\max(N,M)}$. We follow a similar path to the replica method in the case of a single replica. Applying the replica method to RBMs has been done previously in Ref. [27].

To simplify the notation let us assume that $M < N$. Let us consider $u_{1:M}$, $w_{1:M}$ and $x_{M+1:N}$ continuous variables subject to the constraints given by Eq. (17a) *viz.*,

$$Z = 2^{N+M} \int du_{1:M} dw_{1:M} dx_{M+1:N} e^{-\beta E(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x})} \rho(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x}) \tag{21}$$

with

$$\begin{aligned}
\rho(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x}) \;\equiv\; & \langle \prod_{i=1}^{M} \prod_{j=M+1}^{N} \delta(u_i - \frac{1}{\sqrt{2}}(\sum_k U_{ik}^t v_k - \sum_k V_{ik}^t h_k - x_{0i} + y_{0i})) \\
& \delta(w_i - \frac{1}{\sqrt{2}}(\sum_k U_{ik}^t v_k + \sum_k V_{ik}^t h_k - x_{0i} - y_{0i})) \delta(\sum_k U_{jk}^t v_k - x_j) \rangle_{(\boldsymbol{v},\boldsymbol{h})} \\
= \; & \frac{1}{2^{N+M}} \sum_{\{\boldsymbol{v},\boldsymbol{h}\}} \prod_{i=1}^{M} \prod_{j=M+1}^{N} \delta(u_i - \frac{1}{\sqrt{2}}(\sum_k U_{ik}^t v_k - \sum_k V_{ik}^t h_k - x_{0i} + y_{0i})) \\
& \delta(w_i - \frac{1}{\sqrt{2}}(\sum_k U_{ik}^t v_k + \sum_k V_{ik}^t h_k - x_{0i} - y_{0i})) \delta(\sum_k U_{jk}^t v_k - x_j)
\end{aligned} \tag{22}$$

By expressing the partition function as in (21), the intractable term is isolated within the density function $\rho(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x})$. This density function encodes the constraints imposed on the continuous variables $\boldsymbol{u}$, $\boldsymbol{w}$ and $\boldsymbol{x}$ by the discreteness of the binary variables $\boldsymbol{v}$ and $\boldsymbol{h}$. These constraints appear as an effective potential in the continuous variables phase space. In a similar manner to the replica method, we can express the density function $\rho(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x})$ in Fourier space. After some straightforward algebra, we find that:

$$Z = \int du_{1:M} d\hat{u}_{1:M} dw_{1:M} d\hat{w}_{1:M} dx_{M+1:N} d\hat{x}_{M+1:N} e^{\mathcal{S}} \tag{23}$$

where the variables $\hat{u}_{1:M}$, $\hat{w}_{1:M}$ and $\hat{x}_{M+1:N}$ are the conjugate variables and $\mathcal{S}$ is the effective action:

$$\mathcal{S} = -\beta E(\boldsymbol{u},\boldsymbol{w},\boldsymbol{x}) + i \left(\sum_i (\hat{u}_i u_i + \hat{w}_i w_i) + \sum_j \hat{x}_j x_j \right) - (N+M) \ln 2\pi \tag{24}$$

$$+ \ln \mathrm{Tr}_{\{\boldsymbol{v},\boldsymbol{h}\}} \, e^{-i \sum_i (\hat{u}_i f_i(\boldsymbol{v},\boldsymbol{h}) + \hat{w}_i g_i(\boldsymbol{v},\boldsymbol{h})) - i \sum_j \hat{x}_j \sum_k U_{jk}^t v_k} \tag{25}$$

The saddle point approximation leads to the following equalities:

$$\hat{u_l}^* = i\beta\lambda_l u_l \tag{26a}$$

$$\hat{w_l}^* = -i\beta\lambda_l w_l \tag{26b}$$

$$\hat{x_l}^* = i\beta a_{0l} \tag{26c}$$

$$u_l^* = \langle\langle f_i(\boldsymbol{v},\boldsymbol{h})\rangle\rangle \equiv \left\langle\left\langle \frac{1}{\sqrt{2}}(\sum_k U_{lk}^t v_k + \sum_k V_{lk}^t h_k - x_{0l} - y_{0l})\right\rangle\right\rangle \tag{26d}$$

$$w_l^* = \langle\langle g_i(\boldsymbol{v},\boldsymbol{h})\rangle\rangle \equiv \left\langle\left\langle \frac{1}{\sqrt{2}}(-\sum_k U_{lk}^t v_k + \sum_k V_{lk}^t h_k + x_{0l} - y_{0l})\right\rangle\right\rangle \tag{26e}$$

$$x_l^* = \left\langle\left\langle \sum_k U_{lk}^t v_k \right\rangle\right\rangle \tag{26f}$$

Here we have introduced the auxiliary function $f_i(\boldsymbol{v},\boldsymbol{h})$ and $g_i(\boldsymbol{v},\boldsymbol{h})$ and we have denoted the expectation value

$$\langle\langle \bullet \rangle\rangle = \frac{\mathrm{Tr}_{\{\boldsymbol{v},\boldsymbol{h}\}} \bullet e^{-i\sum_i(\hat{u}_i f_i(\boldsymbol{v},\boldsymbol{h})+\hat{w}_i g_i(\boldsymbol{v},\boldsymbol{h}))-i\sum_j \hat{x}_j \sum_k U_{jk}^t v_k}}{\mathrm{Tr}_{\{\boldsymbol{v},\boldsymbol{h}\}} e^{-i\sum_i(\hat{u}_i f_i(\boldsymbol{v},\boldsymbol{h})+\hat{w}_i g_i(\boldsymbol{v},\boldsymbol{h}))-i\sum_j \hat{x}_j \sum_k U_{jk}^t v_k}} \tag{27}$$

The solution $u_{1:M}^*$, $w_{1:M}^*$, $x_{M+1:N}^*$, $\hat{u}_{1:M}^*$, $\hat{w}_{1:M}^*$ and $\hat{x}_{M+1:N}^*$ corresponds to the saddle point. In principle, one can solve these equations self-consistently similar to the replica method [27]. Note that in the high temperature limit the conjugate variables tend to zero in the saddle point approximation while the variables $u_{1:M}^*$, $w_{1:M}^*$ and $x_{M+1:N}^*$ tend to the arithmetic average over the binary states $\boldsymbol{v}$ and $\boldsymbol{h}$:

$$u_{l\,(T\gg\lambda_l)}^* = \frac{1}{\sqrt{2}}\left(\sum_k U_{lk}^t\langle v_k\rangle + \sum_k V_{lk}^t\langle h_k\rangle - x_{0l} - y_{0l}\right) \tag{28a}$$

$$w_{l\,(T\gg\lambda_l)}^* = \frac{1}{\sqrt{2}}\left(-\sum_k U_{lk}^t\langle v_k\rangle + \sum_k V_{lk}^t\langle h_k\rangle + x_{0l} - y_{0l}\right) \tag{28b}$$

$$x_{l\,(T\gg\lambda_l)}^* = \sum_k U_{lk}^t\langle v_k\rangle \tag{28c}$$

In this context, any binary state has an equal probability 2^{-N-M}, which implies that $\langle v_k\rangle = \langle h_k\rangle = 1/2$ for all $k = 1, ..., N+M$.

We mentioned in the previous section that x_i (y_j) can take on values formed by summing subsets of elements in column i-th (j-th) of matrix $\boldsymbol{U}$ ($\boldsymbol{V}$). Additionally, in a randomly initialized RBMs the eigenvector matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are Haar distributed. Hence the clustering of data around the saddle point can be understood as a consequence of the Haar measure and the central limit theorem. The distribution of $u_{1:M}$, $w_{1:M}$ and $x_{M+1:N}$ is close to Gaussian. In Fig. 3 e) we show the kurtosis of the reciprocal variable y for hidden layer sizes $500, 700, 784, 1200, 2000$ for randomly initialized RBMs, with 784 visible nodes. Each data point was generated from averaging over the kurtosis of the hidden nodes with non-zero singular values. The kurtosis for each node was computed from projecting $7 \cdot 10^6$ random binary vector samples to reciprocal space. The ribbon corresponds to the standard deviation over the hidden layer. The dashed line marks the target for Gaussianity. The blue curve is a linear fit extrapolated to hidden layer size infinity where the reciprocal variables become Gaussian. In the case of a trained RBM, the variables $u_{1:M}$, $w_{1:M}$ and $x_{M+1:N}$ are not close to Gaussian, in general. However, for both trained and untrained RBMs, the mean of u_l is equal to $u_{l\,(T\gg\lambda_l)}^*$ and the same applies to w_l and x_l as we will show later. The results in Eqs. (28) will be useful for when we expand the constraint potential when solving the Schrödinger equation as well as for when we introduce the mean field to show the symmetry breaking in reciprocal space.

In the next section we present the connection with the Fokker-Planck Equation and a set of coupled Bosons. We anticipate further connections between RBMs with diffusion-like models and coupled Bosons.

4.1 Fokker-Planck Equation and coupled Bosons

In the previous section we expressed the partition function in terms of the continuous variables $u_{1:M}, w_{1:M}$ and $x_{M+1:N}$, under the assumption that $M < N$. To make notation homogeneous, we introduce the variable

$\boldsymbol{z} = [u_{1:M}, w_{1:M}, x_{M+1:N}]^T$. The partition function becomes:

$$Z = \int d\boldsymbol{z} e^{-\beta U_{eff}(\boldsymbol{z})} \tag{29}$$

where

$$U_{eff}(\boldsymbol{z}) = E(\boldsymbol{z}) - \frac{S_c}{\beta} + \frac{V_{const}(\boldsymbol{z})}{\beta} \tag{30}$$

with S_c denoting the configurational entropy $(N+M)\ln 2$, and

$$V_{const}(\boldsymbol{z}) = -\ln \rho(\boldsymbol{z}) \tag{31}$$

is the constraint potential that arises due to the constraints imposed via the original discrete variables. We assume the potential $U_{eff}(\boldsymbol{z})$ is a confining potential, such that the probability density function

$$P(\boldsymbol{z}) = \frac{e^{-\beta U_{eff}(\boldsymbol{z})}}{Z} \tag{32}$$

is the stationary solution of the Fokker-Planck Equation:

$$\frac{\partial P(\boldsymbol{z},t)}{\partial t} = D \sum_{i=1}^{N+M} \left[\frac{\partial^2 P(\boldsymbol{z},t)}{\partial z_i^2} + \frac{1}{D} \frac{\partial}{\partial z_i} \left(\frac{\partial U_{eff}(\boldsymbol{z})}{\partial z_i} P(\boldsymbol{z},t) \right) \right] . \tag{33}$$

We assume the over-damped regime with the friction coefficient set to $\gamma = 1$ such that $D = 1/\beta$. It is straightforward to show that Eq. (32) is the stationary solution to Eq. (33). By the variable separation method we propose the ansatz $P(\boldsymbol{z},t) = \phi(\boldsymbol{z}) f(t)$, which substituting in Eq. (33) and dividing by the ansatz leads to:

$$\frac{\dot{f}(t)}{f(t)} = \frac{D}{\phi} \sum_{i=1}^{N+M} \left[\frac{\partial^2 \phi(\boldsymbol{z})}{\partial z_i^2} + \frac{1}{D} \frac{\partial}{\partial z_i} \left(\frac{\partial U_{eff}(\boldsymbol{z})}{\partial z_i} \phi(\boldsymbol{z}) \right) \right] = -\Gamma \tag{34}$$

Solving for $f(t)$ yields $f(t) = Ae^{-\Gamma t}$, where A is a constant and $\Gamma > 0$ in order for the solution to not diverge. The parameter Γ is the inverse of the characteristic relaxation time.

There are many ways to solve the remaining of the Fokker-Planck Equation. Here we reduce the equation to a time-independent Schrödinger-like equation by introducing the substitution:

$$\phi(\boldsymbol{z}) = e^{-U_{eff}(\boldsymbol{z})/2D} \psi(\boldsymbol{z}) . \tag{35}$$

This leads to an eigenvalue problem $H\psi(\boldsymbol{z}) = E\psi(\boldsymbol{z})$, where $H = \sum_i^{N+M} \frac{p_i^2}{2} + V_Q^{(i)}(\boldsymbol{z})$ $(m = \hbar = 1)$. Here p_i is the momentum operator and V_Q is the potential defined as:

$$V_Q^{(i)}(\boldsymbol{z}) = \frac{1}{8D^2} \left(\frac{\partial U_{eff}(\boldsymbol{z})}{\partial z_i} \right)^2 - \frac{1}{4D} \frac{\partial^2 U_{eff}(\boldsymbol{z})}{\partial z_i^2} \tag{36}$$

We can formally express the general solution as:

$$P(\boldsymbol{z},t) = \sum_n c_{\Gamma_n} e^{-\frac{1}{2D} U_{eff}(\boldsymbol{z})} \psi_{\Gamma_n}(\boldsymbol{z}) e^{-\Gamma_n t} \tag{37}$$

with $\Gamma_n = E_n/2D$. The ground state corresponds to $\Gamma_0 = E_0/2D = 0$, such that we can rewrite the previous general solution as:

$$P(\boldsymbol{z},t) = c_0 e^{-\frac{1}{2D} U_{eff}(\boldsymbol{z})} \psi_0(\boldsymbol{z}) + \sum_{\Gamma_n > 0} c_{\Gamma_n} e^{-\frac{1}{2D} U_{eff}(\boldsymbol{z})} \psi_{\Gamma_n}(\boldsymbol{z}) e^{-\Gamma_n t} \tag{38}$$

from which it is easy to notice that $c_0 = 1/Z$ and $\psi_0(\boldsymbol{z}) = e^{-\frac{1}{2D} U_{eff}(\boldsymbol{z})}$.

The explicit general solution will depend on the effective potential $U_{eff}(\boldsymbol{z})$. In general, we can consider expanding the effective potential around a minimum. Notice that the maximum log-likelihood of the point PDF $\rho(\boldsymbol{z})$ corresponds to the minimum in $V_{const}(\boldsymbol{z})$. Hence, we can approximate the constraint potential by Taylor expansion around the minimum,

$$V_{const}(\boldsymbol{z}) \approx V_{const}(\boldsymbol{\mu}_z) + \frac{1}{2} \sum_{ij} (z_i - \mu_{z_i}) k_{ij} (z_j - \mu_{z_j}) , \tag{39}$$

where $k_{ij} = \frac{\partial^2 V_{const}(\boldsymbol{z})}{\partial z_i \partial z_j}|_{\boldsymbol{\mu}_z}$ and μ_z is the expectation value of z, which correspond to the high temperature limit in the saddle point expansion (see Eqs. (28)). Before moving forward, we introduce the parameters

$$\omega_i = \begin{cases} \frac{k_{ii}-\lambda\beta}{2} & \text{for } i = 1, ..., M \\ \frac{k_{ii}+\lambda\beta}{2} & \text{for } i = M+1, ..., 2M \\ \frac{k_{ii}}{2} & \text{for } i = 2M+1, ..., N \end{cases} \tag{40}$$

and

$$z_{i0} = \begin{cases} \frac{\mu_{z_i} k_{ii}}{2\omega_i} & \text{for } i = 1, ..., 2M \\ \frac{\beta a_i}{2\omega_i} + \mu_{z_i} & \text{for } i = 2M+1, ..., N \end{cases} \tag{41}$$

in addition to making the change in variable $\zeta_i = z_i - z_{i0}$. After some algebra, the Schrödinger potential becomes:

$$\begin{aligned} V_Q^{(i)} &= \frac{1}{2}\omega_i^2\zeta_i^2 + \frac{1}{2}\omega_i\zeta_i \sum_{j\neq i}\zeta_j k_{ij} + \frac{1}{2}\omega_i\zeta_i \sum_{j\neq i}(z_{j0} - \mu_{z_j})k_{ij} \\ &+ \frac{1}{2}\left(\frac{1}{2}\sum_{j\neq i}(\zeta_j + z_{j0} - \mu_{z_j})k_{ij}\right)^2 - \frac{1}{2}\omega_i + \mathcal{O}(\zeta^4) \end{aligned} \tag{42}$$

The previous potential correspond to that of $N+M$ coupled harmonic oscillators. We can rewrite the potential $V_Q(\boldsymbol{\zeta})$ as:

$$V_Q(\boldsymbol{\zeta}) \approx \frac{1}{2}\langle\boldsymbol{\zeta}|M|\boldsymbol{\zeta}\rangle + \frac{1}{2}\langle\boldsymbol{\zeta}|O|\boldsymbol{\Delta}\rangle + \frac{1}{2}\langle\boldsymbol{\Delta}|\Theta|\boldsymbol{\Delta}\rangle - \frac{1}{2}\langle\mathbf{1}|\Omega|\mathbf{1}\rangle \ . \tag{43}$$

where

$$\begin{cases} \Delta_k = z_{k0} - \mu_{z_k} \\ \Theta_{ij} = \frac{1}{4}\sum_{k\neq j, k\neq i} k_{ik}k_{kj} \\ M_{ij} = \omega_i^2\delta_{ij} + w_i k_{ij}(1-\delta_{ij}) + \Theta_{ij} \\ O_{ij} = \omega_i k_{ij}(1-\delta_{ij}) + 2\Theta_{ij} \\ \Omega_{ij} = \omega_i\delta_{ij} \end{cases} \tag{44}$$

The matrix M is a square symmetric matrix, which can be diagonalized.

We can obtain k_{ij} by assuming that the reciprocal variables are Gaussian distributed, *viz.*

$$\rho(\boldsymbol{z}) = \frac{\sqrt{det(\boldsymbol{k})}}{(2\pi)^{N/2}} e^{-\frac{1}{2}\sum_{ij}(z_i-\mu_{z_i})k_{ij}(z_j-\mu_{z_j})} \tag{45}$$

which leads to

$$\frac{1}{k_{ij}} = \langle(z_i - \mu_{z_i})(z_j - \mu_{z_j})\rangle = \frac{1}{4}\delta_{ij} \tag{46}$$

The previous implies that $\Theta_{ij} = O_{ij} = 0$ for all i and j. The potential in Eq. (43) reduces to:

$$V_0^{(i)} = \frac{1}{2}\omega_i^2\zeta_i^2 - \frac{1}{2}\omega_i \tag{47}$$

Notice that in this case, the problem reduces to a set of $N+M$ uncoupled harmonic oscillators with a shift in the potential energy. The eigenstates to the Schrödinger Equation reduce to a product of Hermite polynomials, while the eigenvalues are

$$E(\boldsymbol{n}) = \sum_{i=1}^{N+M}\omega_i\left(n_i + \frac{1}{2}\right) - \frac{\omega_i}{2} = \sum_{i=1}^{N+M}\omega_i n_i \tag{48}$$

where $n_i = 0, 1, 2, ...$ for all i. The inverse characteristic relaxation time becomes $\Gamma(\boldsymbol{n}) = \beta E(\boldsymbol{n})/2$. Notice that when $\boldsymbol{n} = \mathbf{0}$ then $E(\mathbf{0}) = 0$ corresponding to the stationary solution, as expected. For $\boldsymbol{n} \neq \mathbf{0}$, when $\lambda_i > k_i/\beta$ the eigenvalues become negative and the solution diverges. In practice, we do not observe this behavior since the oscillators are in general coupled. Therefore, higher-order terms in the constraint potential need to be considered. In a previous section we showed that the reciprocal variables are Gaussian distributed when either of the partition sizes tend to infinity (see Fig. 3 e). In such scenario, we expect the constraint potential to have terms up to second order. Consequently, there will be some modes for which the diffusion process will not converge to the Boltzmann distribution (see Eq. (38)).

Note that the only approximation here was the Taylor expansion of the constraint potential, which ultimately is a standard approach when solving a many-body problem. In this sense, we have shown a clear connection between RBM, diffusion processes and coupled Bosons.

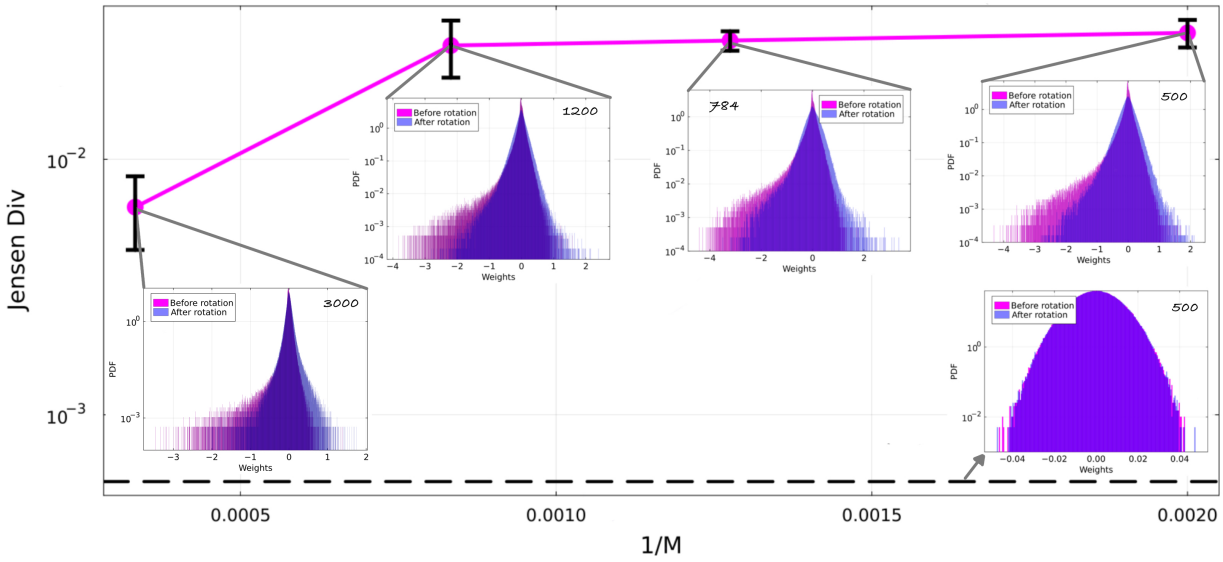

Figure 4: Jensen divergence between rotated and non-rotated weight matrix for RBMs with $M = 500, 784, 1200, 3000$. Each point corresponds to an average over five replicas. The dashed line corresponds to a non-trained RBM with $M = 500$. PDF of RBM weight matrix before and after random rotations for one of the replicas is shown.

5 Symmetry breaking

Deep learning process occurs hierarchically, *i.e.*, different degrees of freedom capture different features of the data. These features usually have a hierarchical sorting [28]. The hierarchical learning process is accompanied by a symmetry breaking and restoration in parameter space [11]. In the case of RBMs, it has been observed that the eigenvector with the highest singular value aligns with the principal components of the dataset [27]. This process can be understood as hierarchical learning. Here we show that training the RBM breaks the rotational symmetry in parameter space. In addition, this symmetry breaking is tied to the hierarchical learning. We show this by randomly rotating the eigenvector matrix and observing the effect it has on the image class. Lastly, we also show a symmetry breaking in the Landau sense in the energy landscape during training.

5.1 Rotational Symmetry

RBMs are typically initialized by drawing each element of the weight matrix independently from a Gaussian distribution with mean 0 and standard deviation 0.01 [29]. This initialization ensures that the RBM begins in a paramagnetic phase [30, 27]. Moreover, this procedure inherently confers rotational invariance to the joint probability density of the weight elements. In fact, the joint probability density of the weight elements is given by

$$p(\{W_{ij}\}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{NM} e^{-\sum_{i,j}\frac{W_{ij}^2}{2\sigma^2}} \propto \exp\left(-\operatorname{Tr}(\boldsymbol{W}\boldsymbol{W}^t)\right) \tag{49}$$

Since $\boldsymbol{W}\boldsymbol{W}^t$ can be decomposed as $\boldsymbol{W}\boldsymbol{W}^t = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^t\boldsymbol{U}^t$, consider a rotation matrix $\mathcal{R}$, and define the rotated orthogonal matrix $\boldsymbol{U}_R = \mathcal{R}\boldsymbol{U}\mathcal{R}^t$. Noting that

$$\operatorname{Tr}\left(\boldsymbol{U}_R\boldsymbol{\Sigma}\boldsymbol{\Sigma}^t\boldsymbol{U}_R^t\right) = \operatorname{Tr}\left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^t\boldsymbol{U}^t\right) = \operatorname{Tr}\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}^t\right) = \sum_i \lambda_i^2 \,, \tag{50}$$

it follows that $\operatorname{Tr}(\boldsymbol{W}\boldsymbol{W}^t)$ is also rotationally invariant and a similar argument holds for $\operatorname{Tr}(\boldsymbol{W}^t\boldsymbol{W})$. Therefore, $p(\{W_{ij}\})$ is rotationally invariant.

We probe the rotational symmetry in trained RBMs by applying random rotations on the eigenvector matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, and studying the effect on the weight matrix distribution. We build a rotation matrix using the method described in Ref. [31]. This method performs rotations in $n-$dimensions around any arbitrary

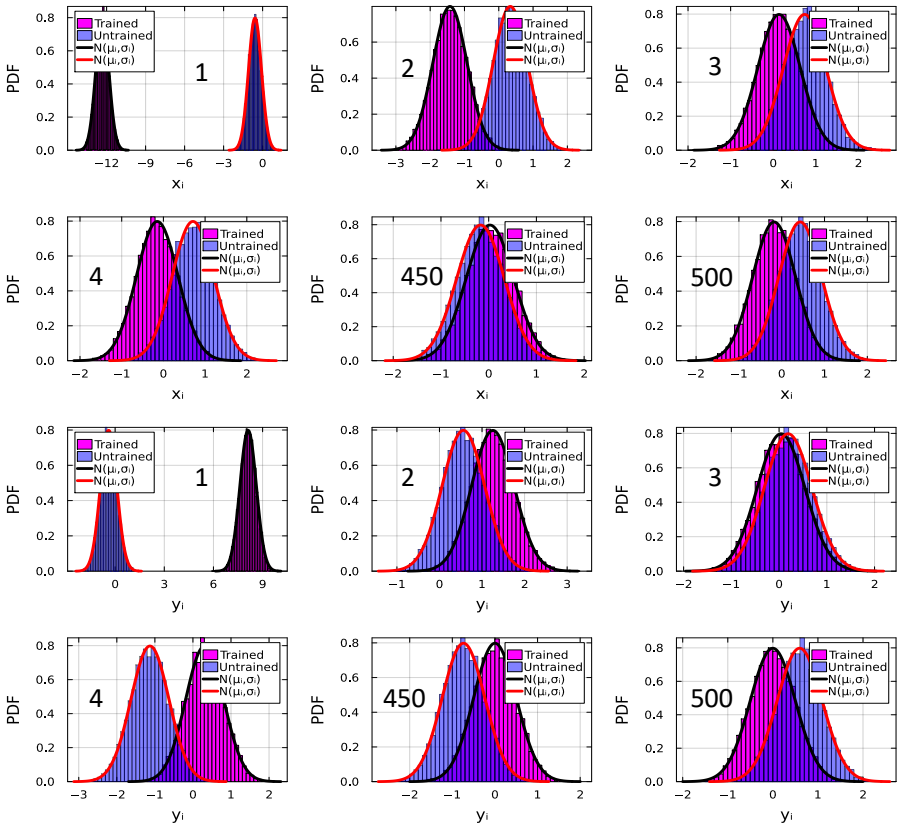

Figure 5: Probability density function of the reciprocal variables for various energy landscapes for trained and non-trained RBM. The samples where generated from a multivariate 1/2-Bernoulli distribution projected onto reciprocal space. The continuous curves correspond to a normal distribution centered at $\mu_x^{(i)}$ and $\mu_y^{(i)}$ and standard deviation $\sigma_x^{(i)}$ and $\sigma_y^{(i)}$ given by (51).

$(n-2)$-dimension subspace. We then apply $N*0.1$ and $M*0.1$ consecutive random rotations to random subspaces of $(N-2)$-dimension and $(M-2)$-dimension of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively. We then compare the weight matrix before and after the random rotations in the case of an initialized RBM with 500 hidden nodes, and trained RBMs with $500, 784, 1200$ and 3000 hidden nodes by plotting the weight matrix PDF in Fig. 4. We also show the Jensen divergence between the rotated and non-rotated weight matrix. Each point corresponds to 5 replicas. The black dashed line corresponds to a non-trained RBM. This shows that the training process breaks the rotational symmetry.

Rotational invariance of the joint distribution ensures that the eigenvector matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are Haar distributed, *i.e.*, they are uniformly distributed over the group $O(N)$ and each eigenvector is uniformly distributed on the unit hypersphere [25]. This property enables the use of the central limit theorem by noticing that projecting binary states to reciprocal space corresponds to summing row elements of $\boldsymbol{U}^t$ and $\boldsymbol{V}^t$ from random positions. But even for trained RBMs where the rotational symmetry is broken, the behavior of projection of binary states onto reciprocal space shows centrality. To illustrate this, we randomly sampled 10^4 states and projected each into the reciprocal space for both, a randomly initialized RBM and a trained RBM. In Fig. 5 we show the resulting probability density functions for various energy landscape indices. The continuous curves correspond to a normal distribution centered at $\mu_x^{(i)}$ and $\mu_y^{(i)}$ and standard deviation $\sigma_x^{(i)}$

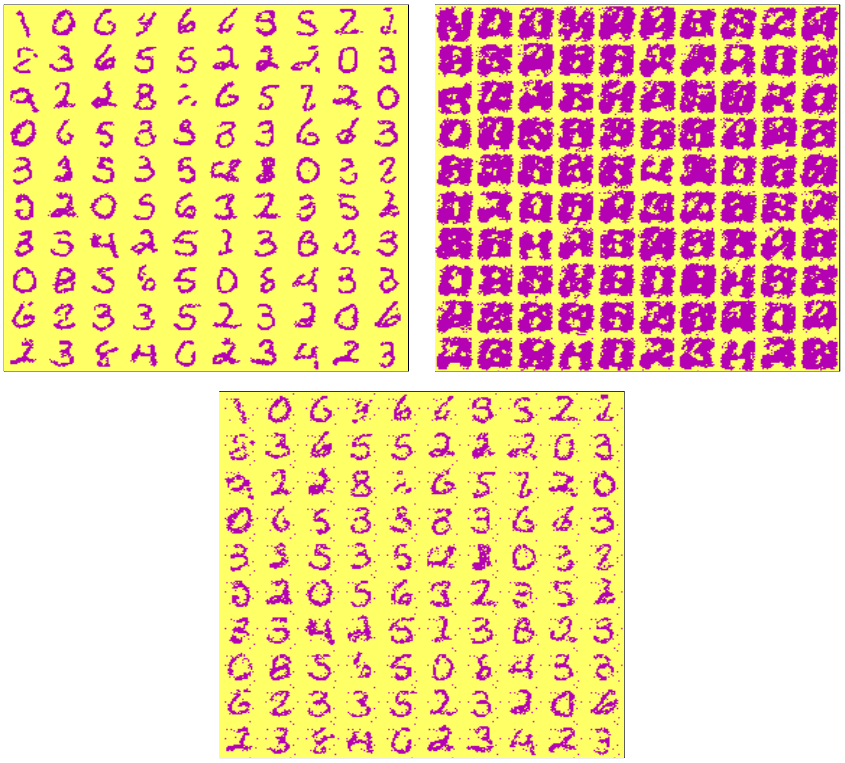

Figure 6: Gibbs sampled data. Same data after rotating by π the eigenvector matrix $\boldsymbol{U}$ around all but the two dimensions associated with the largest singular values. Same data after 10 consecutive rotations around any $(N-2)$-dimension where the dimensions associated with the five largest singular values are always included and the remaining $N-7$ dimensions are randomly selected.

and $\sigma_y^{(i)}$, defined as

$$\mu_x^{(i)} = \frac{1}{2}\sum_j U_{ij}^t \,, \tag{51a}$$

$$\mu_y^{(i)} = \frac{1}{2}\sum_j V_{ij}^t \,, \tag{51b}$$

$$\sigma_x^{(i)} = \sigma_y^{(i)} = \frac{1}{2} \,, \tag{51c}$$

which is straightforward to obtain from computing the first and second moment of the reciprocal variables. Notice that the first moment in the previous Equation is equivalent to the saddle point expansion at high temperature obtained in Eqs. (28). Moreover, these expectation values were used in section 4.1 for the Taylor expansion. In the following, we show that the rotational symmetry breaking in parameter space is tied to hierarchical learning in the RBM.

5.2 Hierarchical Learning

Hierarchical learning is the mechanism by which different pieces of deep learning models learn different levels of abstraction. In the case of deep neural networks, it has been shown that deeper layers in the network learns more complex features [32]. In diffusion models, it has been shown that the feature hierarchies occur at different timescale during the denoising process [28]. In GANs it has been shown that the class typically clusters around a subset of eigenvectors in the latent space [33]. In the case of RBM, it has been shown that during training, the eigenvectors align with the principal components of the dataset [27], which we say is similar to hierarchical learning. We can probe this claim by rotating the eigenvector matrix U. To do so, notice that when sampling from an RBM, the last step consists on sampling from $\sigma(\boldsymbol{W}\boldsymbol{h}+\boldsymbol{b}) = \sigma(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^t\boldsymbol{h}+\boldsymbol{b})$. Consider a rotation matrix $\mathcal{R}$, and define the rotated orthogonal matrix $\boldsymbol{U}_R = \mathcal{R}\boldsymbol{U}\mathcal{R}^t$. We build a rotation

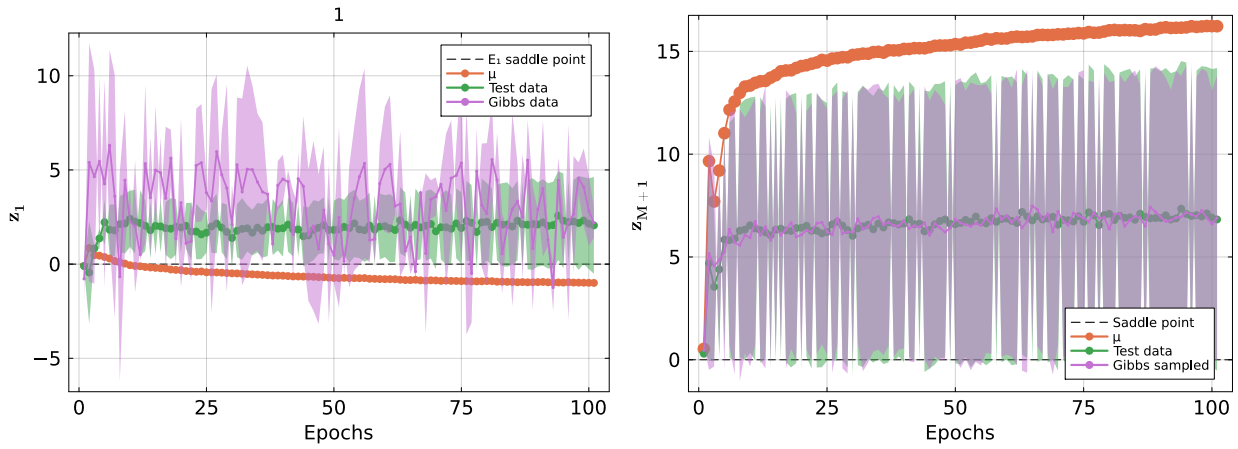

Figure 7: Reciprocal variable z_1 (left) and z_{M+1} (right) *vs* epochs for an RBM with $M = 500$ hidden nodes. The dashed horizontal line marks the position of the energy function saddle point. The orange markers correspond to the constraint potential minima (see Eq. (28)). The green markers correspond to the average in reciprocal space of the test dataset, whereas the purple markers correspond to the average over reciprocal space of Gibbs sampled data. The ribbons correspond to the standard deviation.

matrix using the method described in Ref. [31]. This method performs rotations in $n-$dimensions around any arbitrary $(n-2)$-dimension subspace. In Fig. 6 we show a sample generated by block Gibbs. We also show the effect on the samples of rotating by π the eigenvector matrix $\boldsymbol{U}$ around all but the two dimensions associated with the largest singular values. Similarly, we show the case where we do 10 consecutive rotations around any $(N-2)$-dimension where the dimensions associated with the five largest singular values are always included and the remaining $N-7$ dimensions are randomly selected. Notice how rotating the eigenvectors associated with the largest singular values has a global effect on the image, whereas rotating eigenvectors associated with any but the principal eigenvectors has a local effect on the image. In this sense, the eigenvectors are associated with different feature hierarchies.

In particular, performing a π-rotation on the principal eigenvector has the effect of flipping many bits together, reminiscent of a ferromagnetic phase transition. In what follows, we introduce a mean field, and show that the energy landscapes associated to non-zero singular values can be approximated by free energy similar to the paramagnetic-ferromagnetic free energy in Landau theory.

5.3 Symmetry breaking in energy landscape

In this section we show how the energy landscape associated with the largest singular value presents a symmetry breaking in the Landau sense. We start from the effective potential presented in Eq. (30). We can expand the constraint potential around the minimum $\boldsymbol{\mu}$ up to fourth order, and after arranging terms we reach:

$$U_{eff}(\boldsymbol{z}) = -\frac{S_c}{\beta} + \frac{V_0}{\beta} + \sum_i U^{(i)}_{eff}(\boldsymbol{z}) \tag{52}$$

with

$$\begin{aligned} U^{(i)}_{eff}(\boldsymbol{z}) \quad = \quad & \frac{a_{0i}b_{0i}}{\lambda_i} + \frac{\Lambda_i}{2}z_i^2 + (z_i-\mu_i)\Gamma_i(\{z_p\}^{M+N}_{\substack{p=1\\p\neq i}}) + (z_i-\mu_i)^2\Theta_i(\{z_p\}^{M+N}_{\substack{p=1\\p\neq i}}) + (z_i-\mu_i)^3\Psi_i(\{z_p\}^{M+N}_{\substack{p=1\\p\neq i}}) \\ & +(z_i-\mu_i)^4\Phi_i(\{z_p\}^{M+N}_{\substack{p=1\\p\neq i}}) \end{aligned} \tag{53}$$

where we have introduced the following functions:

$$\begin{cases} \Gamma_i(\{z_p\}_{\substack{p=1\\p\neq i}}^{M+N}) = \frac{1}{2\beta}\sum_{j\neq i} k_{ij}(z_j-\mu_j) + \frac{1}{6\beta}\sum_{\substack{j\neq i\\k\neq i}} k_{ijk}(z_j-\mu_j)(z_k-\mu_k) \\ \qquad\qquad + \frac{1}{24\beta}\sum_{\substack{j\neq i\\k\neq i\\l\neq i}} k_{ijkl}(z_j-\mu_j)(z_k-\mu_k)(z_l-\mu_l) \\ \Theta_i(\{z_p\}_{\substack{p=1\\p\neq i}}^{M+N}) = \frac{k_{ii}}{2\beta} + \frac{1}{6\beta}\sum_{j\neq i}\Pi_{iij}k_{iij}(z_j-\mu_j) + \frac{1}{24\beta}\sum_{\substack{j\neq i\\k\neq i}}\Pi_{iijk}k_{iijk}(z_j-\mu_j)(z_k-\mu_k) \\ \Psi_i(\{z_p\}_{\substack{p=1\\p\neq i}}^{M+N}) = \frac{k_{iii}}{6\beta} + \frac{1}{24\beta}\sum_{j\neq i}\Pi_{iiij} \\ \Phi_i(\{z_p\}_{\substack{p=1\\p\neq i}}^{M+N}) = \frac{k_{iiii}}{24\beta} \end{cases} \tag{54}$$

and $\Lambda_i = \lambda_{i-M}$ for $i > M$ and $\Lambda_i = -\lambda_i$ otherwise. Notice that the previous functions depend on all z-variables but z_i. Determining all the parameters in (54) in general is not tractable. However, we can introduce a mean field, such that $z_k = \mu_k + \epsilon_k$ with $\langle\epsilon_k\rangle = 0$ and $\langle\epsilon_k\epsilon_l\rangle = \delta_{kl}$. We then average the functions in (54) and obtain:

$$f_i(z_i) \equiv \langle U_{eff}^{(i)}\rangle(z_i) = \frac{a_{0i}b_{0i}}{\lambda_i} + \frac{\Lambda_i}{2}z_i^2 + \langle\Gamma_i\rangle(z_i-\mu_i) + \langle\Theta_i\rangle(z_i-\mu_i)^2 + \langle\Psi_i\rangle(z_i-\mu_i)^3 + \langle\Phi_i\rangle(z_i-\mu_i)^4 \tag{55}$$

which resembles the free energy in Landau theory. To validate this approximation, in Fig. 7 we plot the evolution of μ_i *vs* epochs for $i = 1$ and $i = M + 1$. We also include the projection of Gibbs sampled data and test data projected onto the energy landscape, which we can assume correspond to the effective potential minimum. For $i = 1$ we observe a symmetry breaking occurring at early epochs where the free energy evolves from an harmonic well to a double well, reminiscent of a ferromagnetic phase transition. The fact that μ_i is located on opposite side of the local minimum with respect to the origin indicate that the double well is non-symmetric. Hence the interplay between the energy function and the constraint potential during training induces symmetry breaking in reciprocal space. Specifically, an increase in magnitude of the prefactor $-\lambda_i/2$ together with a shift of value of the minimum position of the constraint potential, μ_i, from zero to non-zero leads to the symmetry breaking. In contrast, for $i = M + 1$ we observe that the corresponding mode becomes relatively flat.

This characteristic persists across different hidden layer sizes, as illustrated in Fig. 8. As the hidden layer size increases, the constraint potential minimum, μ_i, shifts closer to the effective potential minimum obtained by the test set and block Gibbs sampling. In general, the symmetry breaking is explicit, since the free energy is asymmetric. In the special case where the number of hidden nodes equals the number of visible nodes, the constraint potential minimum tends to align closely with the energy function saddle point. However, the free energy is still asymmetric, in general, due to the first- and third-order terms in the free energy.

6 Conclusions

In the previous sections we introduced a reciprocal space method. This method allows a better understanding on RBM initialization and training. With this method we showed a direct mapping from a Restricted Boltzmann Machine to a diffusion process. This diffusion is governed by an effective potential that encapsulates the discrete nature of the binary variables through inherent constraints. Although the mapping is quite general, its practical implications have yet to be fully explored. Furthermore, we established a direct connection between an RBM and a many-body problem, specifically, a system of coupled Bosons, in which the eigenvalues are proportional to the relaxation times in the RBM and the logarithm of the ground state is proportional to the effective potential. We also showed that the reciprocal variables are Gaussian distributed to first approximation. As the number of hidden nodes grows, the reciprocal variables tend to Gaussianity. Consequently, if the reciprocal variables are Gaussian, the diffusion process diverges from the stationary solution for certain modes. We intend to investigate this prediction in future work. Another interesting result relates to the distribution of singular values in the case of trained RBMs, shown in Fig. 3, where despite the different sizes considered for the hidden layer, the distribution is qualitatively similar. We systematically observed a gap at $\lambda \in [33, 35]$ for trained RBMs and currently we have no explanation for this.

We further demonstrated that training an RBM induces a break in rotational symmetry, which in turn affects the learning process by enabling the weight matrix eigenvectors to capture different data features in a hierarchical sort. In particular, rotating the eigenvectors associated with the largest singular value has a non-local effect on the binary space. Within the framework of statistical physics, we can interpret the

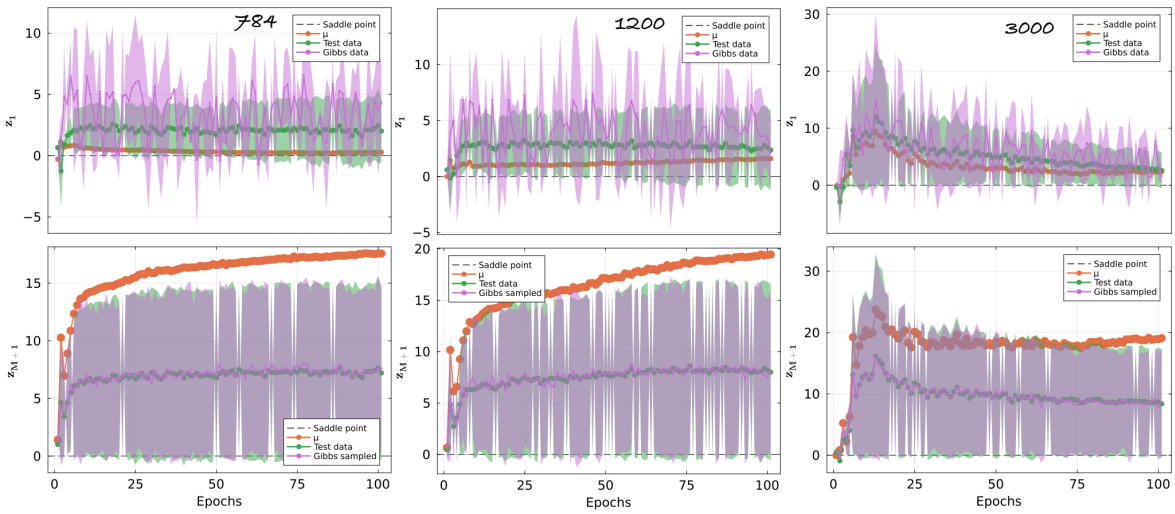

Figure 8: Reciprocal variable z_1 (left) and z_{M+1} (right) *vs* epochs for RBMs with (**left**) 784, (**center**) 1200 and (**right**) 3000 hidden nodes. The dashed horizontal line marks the position of the energy function saddle point. The orange markers correspond to the constraint potential minima (see Eq. (28)). The green markers correspond to the average in reciprocal space of the test dataset, whereas the purple markers correspond to the average over reciprocal space of Gibbs sampled data. The ribbons correspond to the standard deviation.

reciprocal variable associated with the largest singular values as an order parameter. Moreover, we showed that the effective potential associated with non-zero singular values can be expressed in terms of a free energy formulation reminiscent of Landau theory. Finally, our results reveal a symmetry breaking in the free energy landscape during training. It remains to be seen whether increasing the number of hidden nodes hinders the symmetry breaking and how this impacts the learning process.

7 Acknowledgements

JQTM is thankful to Jorge Fernandez de Cossio, Daniel Miravet, Mehdi Drissi and Ejaaz Merali for useful discussions. JQTM acknowledges a Mitacs Elevate Postdoctoral Fellowship (IT39533) with Perimeter Institute for Theoretical Physics. We gratefully acknowledge funding from the National Research Council (Canada) via Agreement AQC-002. This research was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development and by the Province of Ontario through the Ministry of Research, Innovation and Science. The University of Virginia acknowledges support from NSF 2212550 OAC Core: Smart Surrogates for High Performance Scientific Simulations and DE-SC0023452: FAIR Surrogate Benchmarks Supporting AI and Simulation Research. This research was supported in part by grants NSF PHY-1748958 and PHY-2309135 to the Kavli Institute for Theoretical Physics (KITP).

References

[1] Lei Wang. Discovering phase transitions with unsupervised learning. *Physical Review B*, 94(19):195105, 2016.

[2] Giacomo Torlai and Roger G Melko. Learning thermodynamics with boltzmann machines. *Physical Review B*, 94(16):165134, 2016.

[3] Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

[4] Juan Carrasquilla, Giacomo Torlai, Roger G Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1(3):155–161, 2019.

[5] Kyle Sprague and Stefanie Czischek. Variational monte carlo with large patched transformers. *Communications Physics*, 7(1):90, 2024.

[6] Jia-Qi Wang, Rong-Qiang He, and Zhong-Yi Lu. Generalized lanczos method for systematic optimization of neural-network quantum states. *arXiv preprint arXiv:2502.01264*, 2025.

[7] Vinicius Mikuni and Benjamin Nachman. Score-based generative models for calorimeter shower simulation. *Physical Review D*, 106(9):092009, 2022.

[8] Oz Amram and Kevin Pedro. Denoising diffusion models with geometry adaptation for high fidelity calorimeter simulation. *Physical Review D*, 108(7):072014, 2023.

[9] Luigi Favaro, Roman Kogler, Alexander Paasch, Sofia Palacios Schweitzer, Tilman Plehn, and Dennis Schwarz. How to unfold top decays. *arXiv preprint arXiv:2501.12363*, 2025.

[10] Nathan Huetsch, Javier Mariño Villadamigo, Alexander Shmakov, Sascha Diefenbacher, Vinicius Mikuni, Theo Heimel, Michael James Fenton, Kevin Thomas Greif, Benjamin Nachman, Daniel Whiteson, et al. The landscape of unfolding with machine learning. *SciPost Physics*, 18(2):070, 2025.

[11] Liu Ziyin, Yizhou Xu, Tomaso Poggio, and Isaac Chuang. Parameter symmetry breaking and restoration determines the hierarchical learning in ai systems, 2025.

[12] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019.

[13] Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[15] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

[16] Nicolas Le Roux and Yoshua Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural computation*, 20(6):1631–1649, 2008.

[17] Aurélien Decelle, Cyril Furtlehner, and Beatriz Seoane. Equilibrium and non-equilibrium regimes in the learning of restricted boltzmann machines. *Advances in Neural Information Processing Systems*, 34:5345–5359, 2021.

[18] Jorge Fernandez-de Cossio-Diaz, Simona Cocco, and Rémi Monasson. Disentangling representations in restricted boltzmann machines without adversaries. *Physical Review X*, 13(2):021003, 2023.

[19] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[20] KyungHyun Cho, Tapani Raiko, and Alexander T Ihler. Enhanced gradient and adaptive learning rate for training restricted boltzmann machines. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 105–112. Citeseer, 2011.

[21] Jan Melchior, Asja Fischer, and Laurenz Wiskott. How to center deep boltzmann machines. *Journal of Machine Learning Research*, 17(99):1–61, 2016.

[22] J. Quetzalcoatl Toledo-Marin. RBM. `https://github.com/jquetzalcoatl/RBM`, 2025.

[23] Ruslan Salakhutdinov. Learning and evaluating boltzmann machines. *Utml Tr*, 2:21, 2008.

[24] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Accurate and conservative estimates of mrf log-likelihood using reverse annealing. In *Artificial Intelligence and Statistics*, pages 102–110. PMLR, 2015.

[25] Marc Potters and Jean-Philippe Bouchaud. *A first course in random matrix theory: for physicists, engineers and data scientists*. Cambridge University Press, 2020.

[26] Ian Percival and Derek Richards. *Introduction to dynamics*. Cambridge University Press, 1982.

[27] Aurélien Decelle and Cyril Furtlehner. Restricted boltzmann machine: Recent advances and mean-field theory. *Chinese Physics B*, 30(4):040202, 2021.

[28] Antonio Sclocchi, Alessandro Favero, and Matthieu Wyart. A phase transition in diffusion models reveals the hierarchical nature of data. *Proceedings of the National Academy of Sciences*, 122(1):e2408799121, 2025.

[29] Geoffrey E Hinton. A practical guide to training restricted boltzmann machines. In *Neural Networks: Tricks of the Trade: Second Edition*, pages 599–619. Springer, 2012.

[30] Jairo RL de Almeida and David J Thouless. Stability of the sherrington-kirkpatrick solution of a spin glass model. *Journal of Physics A: Mathematical and General*, 11(5):983, 1978.

[31] Antonio Aguilera and Ricardo Pérez-Aguila. General n-dimensional rotations, 2004.

[32] Alsallakh Bilal, Amin Jourabloo, Mao Ye, Xiaoming Liu, and Liu Ren. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics*, 24(1):152–162, 2017.

[33] J Quetzalcóatl Toledo-Marín and James A Glazier. Using deep LSD to build operators in GANs latent space with meaning in real space. *Plos one*, 18(6):e0287736, 2023.